

Context-Awareness in Smart Homes to Support Independent Living

Ronny Haryanto
ronny@haryan.to

August 2005

Master of Science in Internetworking (Co4160)
University of Technology, Sydney
Supervisor: Dr. Peter Leijdekkers

Abstract

The growth of new home technologies, its ubiquity, and the ageing population of the world are some of the main factors that have generated ideas and interests in combining them together to create a new breed of applications that can be used to allow ageing in place, by supporting elderly individuals to live independently at the comfort of their own home while putting the minds of extended family members and care givers at ease. This literature review report will discuss various research and current issues in ubiquitous computing, context-aware computing, smart homes, independent living, how they relate to each other, and how context-awareness and smart homes can provide support for independent living.

Contents

1	Introduction	1
2	Context-Aware Computing	4
2.1	Ubiquitous Computing	4
2.2	Context-Aware Computing	6
2.2.1	Context	6
2.2.2	Context Provision	8
2.2.3	Managing Contextual Information	13
2.2.4	Context Adaptation	16
2.2.5	Architecture of Context-Aware Applications	17
2.3	Requirements and Challenges	19
2.3.1	General Ubiquitous Computing Requirements	20
2.3.2	Context-Aware Computing Specific Challenges	22
2.4	Current Research and Prototypes	24
2.4.1	Context-Aware Frameworks	25
2.4.2	Context-Aware Applications	26
3	Smart Homes	27
3.1	Vision of Smart Homes	27
3.2	Context-Awareness and Smart Homes	27
3.3	Common Infrastructure in Homes to Support Context Awareness	28
3.4	Frameworks and Architecture	30
3.5	Existing Applications in the Smart Home	31
3.5.1	Commercial Applications	31
3.5.2	Current Research Prototypes	33
3.6	Issues Specific to Context-Awareness in Smart Homes	33
4	Support for Independent Living	34
4.1	About Independent Living	34
4.2	Motivation for Context-Aware Homes that Support Independent Living	35
4.3	General Requirements	36
4.4	Applications to Support Independent Living	37
4.4.1	Existing Applications	37
4.5	Issues Specific to Context-Awareness in Supporting Independent Living	47
5	Research Proposal	48

6	Conclusions	50
7	Acknowledgements	51
	References	52

List of Figures

1	Simplified Context-Aware Architecture	18
---	---	----

1 Introduction

People spend most of their time at home. A home has always been associated with a safe place that provides shelter, a place to rest and relax, a place where family members—and occasionally, friends—gather, share meals and interact with one another, a place to come home to, essentially—in a sense—a place where one belongs. Many daily activities are performed while one is at home. Some of the activities like brushing teeth, taking a shower, getting ready to go to work, preparing breakfast, and so on are quite mundane, we do not normally give much attention to them because we are used to them.

Technological advancements, such as the development and widespread use of Wi-Fi¹ (Wireless Fidelity, based on the IEEE 802.11² wireless standards), Bluetooth³, combined with readily available inexpensive Internet access, has spurred many ideas and innovations to apply those technology in the home. Samsung Electronics developed a web-enabled refrigerator⁴ that has a Liquid Crystal Display (LCD) and can connect to the Internet to allow the user to download recipes and check their emails. There are various inexpensive and commercially available X10⁵ devices that can be used for home automation, e.g. for scheduling the garden lights, or turning on the air conditioning via the Internet. Programmable sensors, such as Phidgets⁶, allow programmers and hobbyists to develop their own applications using the sensors (e.g. temperature, pressure, light, etc.).

A *smart home* has its appliances and various parts of the home, that make use of those technological advancements, work together to improve the quality of life of its occupants. The notion of smart homes has been around for quite some time now, as early as 1958 in the movie *Mon Oncle*⁷, and we can see it visualised by many artists in futuristic movies or in fiction novels ever since. However, not many people outside the academia may realise that smart home technologies are an active research topic and there are many interesting and useful applications that are available as a result, and some of them

¹<http://en.wikipedia.org/wiki/WiFi>

²<http://grouper.ieee.org/groups/802/11/>

³<http://en.wikipedia.org/wiki/Bluetooth>

⁴http://product.samsung.com/cgi-bin/nabc/product/b2c_product_detail.jsp?prod_id=RH2777AT%252fXAA

⁵http://en.wikipedia.org/wiki/X10_%28industry_standard%29

⁶<http://www.phidgets.com/>

⁷http://en.wikipedia.org/wiki/Mon_Oncle

are in use today in our daily lives. There are many other technologies that make our homes *smarter*, such as home security systems, resource conservation systems, home automation systems, integrated entertainment systems, and so on. Some of these technologies even go beyond the home, for example wearable computers or mobile devices that are normally part of smart home environments when they are at home but they also go with the people wherever they go away from home.

Computers are no longer bulky desktop computers, big mainframes or servers. They exist practically everywhere we go now, in many different sizes and shapes, and specialised for many different purposes. From mobile devices like smart phones and Personal Digital Assistants (PDAs), wearable computers, special purpose computer chips that controls many functions of a car, sensors in a parking lot connected to a central computer, wireless cameras used for monitoring connected to an archiving server, interactive home media and entertainment centers such as TiVo⁸, ShowShifter⁹ and Microsoft's Media Center¹⁰, integrated home security systems, medical and health monitoring systems intended for home use (e.g. for elderly individuals), networked home appliances, and many more. It is possible that in the near future we will have even more computing devices in even smaller size that help us make our daily lives better by making our activities more enjoyable, communications faster, clearer, more efficient. The idea that computing devices are everywhere and becoming part of our daily lives is more commonly known as ubiquitous computing (or ubicomp for short) or pervasive computing. A smart home consists of interconnected specialised systems that work together, and those systems are growing in terms of variety, purpose and number but they are getting smaller in sizes. Therefore, the advancements in ubicomp provides an important foundation for smart home environments.

Context-awareness is a specific topic in ubicomp that deals with how devices can infer and understand the current context based on the input given or received by the sensors. By understanding the current context (situation, environment) the user or the device is in, the system can intelligently make a decision to do a certain action, or at the very least notify and ask the user to confirm the action to be taken. Smart devices in the home can potentially do

⁸<http://www.tivo.com/>

⁹<http://www.showshifter.com/>

¹⁰<http://www.microsoft.com/windowsxp/mediacenter/default.mspx>

many useful and interesting things if they "understand" the contexts in which they or their users are. For example, if the system senses that the user is in the bedroom, the user is breathing normally and the eyelids are closed, the room is dark and quiet, and it is past midnight, then the system could infer that the user is sleeping and would like to not be disturbed. The system then could, for example, make the phones in the home silent and redirect incoming calls to the voice mailbox without letting the ringer make any sound. An even smarter system would intelligently analyse the caller's identity and the voice message content—if the caller leaves a message—using a speech analysis system to determine if the call was urgent and if so it would notify the user.

Context-aware smart homes can help improve the quality of life of elderly individuals and people with disabilities that would like to maintain independent lifestyles *in their own home* without having to rely on other people to do routine things for them, or having to move to a senior home. The relevant context-aware smart home applications range from health and medication monitoring systems that can remind the elderly to take their pills and do their exercises, monitor their vital signs, detect emergency situations and ask for outside help, and so on, to memory augmentation applications such as reminders, step-by-step instructions for doing certain tasks like cooking, and other applications like enabling easy communication with their loved ones. A combination of these applications can be used to help elderly individuals and people with disabilities to live independently in their own home, depending on the individuals' specific needs.

This paper gives introductory background information on context-awareness, its usage and importance in smart homes, and more specifically how context-awareness and smart homes can help support independent living for the elderly and for people with disabilities. This paper attempts to identify:

- some of the relevant existing applications and research in the area,
- frameworks and standards available, and which ones are most supported and why,
- challenges and open issues, and
- research and development trends in the area.

The main objective of this paper is to serve as an inspiration for research and development of *new* context-aware applications in the smart homes that support independent living for the elderly and people with disabilities.

The structure of this paper is as follows. In section 2 of this paper I describe ubiquitous computing and context-aware computing in general, enumerate some examples and recent developments in the area, and also the requirements and challenges of ubiquitous computing and context-aware computing. Section 3 discusses smart homes, some examples of smart home technologies, and the role of context-awareness in smart homes. In section 4, I discuss how context-awareness in smart homes can help support independent living. I describe my proposal for a project in developing a context-aware smart home application for the elderly in section 5. And finally, section 6 concludes this paper.

2 Context-Aware Computing

This section describes ubiquitous computing and context-aware computing in general, enumerates some examples and recent developments in the area, and also the requirements and challenges of ubiquitous computing and context-aware computing.

2.1 Ubiquitous Computing

Ubiquitous computing is commonly known by its short form, *ubiquitous computing*. It is also known as *pervasive computing*. (Wikipedia 2005e)

The Merriam-Webster Online Dictionary¹¹ defines the term *ubiquitous* as *existing or being everywhere at the same time; constantly encountered; widespread*. One important aspect of ubiquitous computing is that computing devices are everywhere around us. The computing paradigm is quite the opposite of what we are used to today, where we have one (or more) desktop computer(s) that we focus our attention to. If we have other computing peripherals, then we connect those to the computer to perform various additional computing tasks. In a sense, the desktop computer acts as a central hub or brain. In a ubiquitous

¹¹<http://www.m-w.com/>

computing world, the computing devices are scattered around us and interconnected to each other. They tend to do more specific tasks instead of being general purpose computers that can be programmed to do anything. We do not need to focus our attention to these devices, instead these devices will focus their attention to best understand our needs and provide for it accordingly. (Weiser 1991)

The father of ubiquitous computing, Mark Weiser, said in the beginning of his famous article, *The Computer for the 21st Century*, that *the most profound technologies are those that disappear, they weave themselves into the fabric of everyday life until they are indistinguishable from it* (Weiser 1991). The computers are pushed into the background, out of focus, so that we instead become more aware of the information and other people using the system. I believe this idea is the vision of ubiquitous computing, as well as another important aspect of ubiquitous computing.

To summarise, ubiquitous computing has these following important aspects.

- The computing devices are scattered everywhere around us,
- They are interconnected to and interact with each other,
- They do not require continuous active attention from the user in order to work,
- Each device tends to be more special purpose rather than general purpose,
- They become invisible by being seamlessly integrated into the background.

When ubiquitous computing is applied to the home context, it would turn a regular home into a smart home. However, since the devices are supposed to be invisible, then the appearance of the home itself should not differ much from a regular house. All the computing devices and interconnection infrastructure are hidden from the residents of the home. The only difference that the residents will notice is how the home and its smart applications are adjusting itself and its environment to constantly adapt to the residents' dynamic needs, while requiring none to only minimal interaction between the residents and the smart home applications.

2.2 Context-Aware Computing

Wikipedia (2005a) defines context-awareness as a computer science term which is used to denote *devices that have information about the circumstances under which they operate, and can react accordingly*. Context-aware computing can also be described by what the applications can do. *Context-aware applications can discover and take advantage of contextual information* (Chen & Kotz 2000) in order to provide a service.

Section 1 on page 1 gives a simple example of a context-aware computing device: a smart phone systems than can detect when the home residents may not want to be disturbed (e.g. sleeping, or having dinner), then the system does not let the phones ring, and takes voice messages instead.

These definitions can be broken down into their components. I will elaborate what *contextual information* or *context* means, what is *sensing*, and how to *take advantage* of the context.

Context-aware computing and ubiquitous computing go hand in hand and mutually benefit each other, in my opinion. Having ubiquitous computing environment provide the infrastructure that supports advancements in context-aware computing. And advancements in context-aware computing will in turn help build better ubiquitous computing environments.

Many traditional applications, which are not context-aware, simply make assumptions (Lieberman & Selker 2000) about the user's context while context-aware applications senses the user's actual context and adjust its behaviours accordingly. Lieberman & Selker (2000) argue that context-aware application development approach should move away from the traditional black-box approach, where only explicit input and output are considered, and rather adopt an approach where the current user and application context is taken into account as implicit input and output in addition to—or better yet, *in place of*—the explicit input and output, because the explicit input and output almost always require the user's attention and tend to distract the user.

2.2.1 Context

In a general sense, context can be defined as relevant, interrelated conditions, such as environments or situations, in which something exists or occurs

(Merriam-Webster Inc. 2005). However, it is helpful to narrow down the broad definition. Therefore, we will focus only on the aspects used by context-aware computing. Chen & Kotz (2000) mentioned that many researchers have tried to enumerate examples of context, which in turn classified into three categories by Schilit, Adams & Want (1994). Chen & Kotz (2000) then summarised and expanded the classification as follows:

- *Computing context*, such as available processors, network connectivity, communication costs, communication bandwidth, nearby resources such as printers, displays, workstations, and wireless devices.
- *User context*, such as the user's identity, user's profile, location, movements, people nearby, social situation, gesture, facial expression, emotional state, focus of attention.
- *Physical context*, such as lighting, sound, temperature, humidity, velocity, direction/bearing, altitude, smell, taste, weight, pressure, volume, distances.
- *Time context*, such as time of the day, week, month, season of the year.

Dey & Abowd (2000) gave their own definition, which they claim will make it easier for application developers to enumerate the context for a given application scenario. Their definition of context is *any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*. Dey and Abowd further categorises contexts based on what kind of information application developers will consider the most when designing their applications. Context-aware applications look at the *whos, wheres, whens, and whats* of entities and use this information to determine *why* the situation is occurring.

As previously enumerated examples in the context classification, there are definitely other types of information than location that can be used to provide context (Schmidt, Beigl & Gellersen 1999). However, location is probably the most common context (Abowd, Atkeson, Hong, Long, Kooper & Pinkerton 1997, Cheverst, Mitchell & Davies 1998, Want, Hopper, Falcao & Gibbons 1992, Ward, Jones & Hopper 1997). A simple example would be music

that is played only in the room where a user is in, or lights that turn off by themselves automatically when the last person leaves the room (this combines location with presence as context). There is a similar example of the latter which is actually an old technology, and we probably see it in action everyday when we open the refrigerator. But when talking about context-awareness in ubicomp, and especially for this paper, context-awareness refers to the computational aspects of context-awareness, unlike the mechanical-type context-awareness used in refrigerator lighting (Kim, Park, Lee, Jin, Park, Chung, Choi & Choi 2004).

Although there are many context-aware computing research that are geared towards mobile computing applications (Chen & Kotz 2000, Abowd et al. 1997, Maaß 1997, Want et al. 1992, Weiser 1993, Asthana, Cravatts & Krzyzanowski 1994, Pascoe 1998), it is important to note that context-aware computing is not limited only to mobile computing applications, because there are context-aware computing applications designed for stationary devices as well, for example in smart homes.

2.2.2 Context Provision

The process of gathering contextual information from the surroundings by means of sensors to provide context is commonly known as sensing. Generally, sensing means acquiring one particular low-level or raw contextual information such as temperature, speed, location, noise level and so on. Depending on the type of application and the application itself, such low-level information may be used directly and sufficiently or it may have to be combined with other data to be useful. The combination of many contextual information acquired by sensing is commonly known as *sensor fusion* (Schmidt, Beigl & Gellersen 1999). The result of sensor fusion is a higher-level contextual information. For example, in order to judge what the traffic condition is like when driving a car, an application could combine noise level, heat, motion, and velocity of the passing vehicles. But sensor fusion does not have to always use different kind of contexts, it could use the same kind of sensors placed at different locations, for example.

I will enumerate some examples of sensors and systems that uses specific kind of sensors, that can generate digital data output which in turn can be used in

context-aware applications that perform the computations upon these data. This list is not meant to be exhaustive.

- Location and Presence

- ★ Global Positioning System (GPS) (Wikipedia 2005c), normally used to determine location, but can also give additional information like elevation, and a precise time. GPS can work anywhere on earth as long as the receiver has a clear sight to the sky, meaning it does not work indoors. To determine location, GPS devices receives signal transmitted by at least four of the GPS satellites (there are at least 24 of them orbiting the earth), then performs calculations to *compute the distance to each of the four satellites from the difference between local time and the time the satellite signals were sent* to determine the latitude, longitude and elevation information.
- ★ Wireless LAN (WLAN) positioning systems that perform triangulation calculations based on the received signal levels from fixed access points whose locations are known in advance, such as the *Horus* system (Youssef & Agrawala 2005). Other techniques involve the use of clustering and probabilistic methods (Roos, Myllymäki, Tirri, Misikangas & Sievänen 2002, Youssef, Agrawala & Shankar 2003).
- ★ Wide-area cell-based, such as Global System for Mobile Communication (GSM), positioning methods (Swedberg 1999, Zhu & Durgin 2005). The techniques used are somewhat similar to WLAN positioning system. The system performs triangulation by measuring the received signal strength, angle of arrival, and/or time difference of arrival between the mobile device and the base stations.
- ★ The *Active Badge* (Want et al. 1992) system uses unique IR signals transmitted by the individual badges every few seconds. The signals are then picked up by a network of sensors placed in the building. The sensors are, in turn, connected to a server which then records the location of the badges based on which sensors picked up the signal.
- ★ *RADAR* (Bahl & Padmanabhan 2000) is meant to be an indoor user location tracking system. The system is Radio Frequency (RF) based,

and the experimental testbed described in (Bahl & Padmanabhan 2000) uses 2.4 GHz WaveLAN based network cards in the mobile and the base stations. The mobile stations broadcast User Datagram Protocol (UDP) packets, and then the base stations measure the received signal strengths together with the packet trip times. This information is then used to determine the position of the mobile devices relative to the base stations.

- ★ *CampusSpace* (Fershca, Beer & Narzt 2001) from the University of Linz, Austria, uses a hybrid Wi-Fi (802.11b) and Radio Frequency Identification (RFID) positioning system. The first source is the measured signal strengths between the mobile stations and the access points scattered around the campus. That information is then augmented with the use of strategically placed RFID tags to give more accurate proximity information. The mobile stations are equipped with RFID readers so that they can pick up RFID tags as they move. The locations of the RFID tags are recorded in a database in a server, which is connected to the wireless LAN. The combined information can be stored locally in the mobile station or in the server, which the mobile station connects via the wireless LAN.
- ★ *Place Lab*¹² (LaMarca, Chawathe, Consolvo, Hightower, Smith, Scott, Sohn, Howard, Hughes, Potter, Tabert, Powledge, Borriello & Schilit 2005) is a project and a software architecture that allows provision of location information using existing wireless infrastructure (or *radio beacons*) commonly found around us, such as 802.11, GSM, Bluetooth, and GPS. It is meant to be used in a large environment (a city instead of a building), and is meant to work out-of-the-box without first requiring users to setup a beacon database or collect similar data themselves. The prototype system makes use of beacon databases collected by the wardriving¹³ community, which can be found online on websites such as <http://wagle.net/> and <http://www.wifimaps.com/>.

- Time

¹²<http://www.placelab.org/>

¹³<http://en.wikipedia.org/wiki/Wardriving>

- ★ Built-in clocks in devices, for example the hardware clock that exists in all computers.
 - ★ Radio clocks such as GPS clocks (Wikipedia 2005d), which have much higher accuracy compared to typical consumer devices and ordinary desktop computers.
- Biological and physiological conditions
 - ★ Attention level. The *StartleCam* (Healey & Picard 1998) is a wearable digital image memory computer system that can detect changes in a user's attention level by monitoring changes in the user's skin conductivity level. The user needs to wear on her fingers skin conductivity sensors, which are connected to a wearable computer system that also has a wearable digital camera. When the system senses that the user attention level is high, then it will record some images using the digital camera.
 - ★ The Affective Computing Group at the MIT Media Lab have various projects¹⁴ that can detect a person's emotional state. Some of the projects combine the use of a wide variety of sensors, such as image and video to recognize facial expressions, postural movements sensors, pressure sensitive chairs, mouse pressure, speech recognition, and so on.
 - ★ Alive Wireless Heart Monitor¹⁵ from Alive Technologies can monitor and digitise heart beat rate, electrocardiogram (ECG), and accelerometer.
 - Environmental conditions
 - ★ Phidgets¹⁶ are Universal Serial Bus (USB) based building block sensors that can be connected to a computer. There are various sensors available, most of them detect environmental conditions such as temperature, humidity, electrical voltage and current, force, distance, light, magnetic, rotation, motion, pressure, touch, and vibration.

¹⁴<http://affect.media.mit.edu/projects.php>

¹⁵<http://alivetec.com/products.htm>

¹⁶<http://www.phidgets.com/>

- Multi-sensor systems
 - ★ A prototype employing sensor fusion to detect light level, tilt and vibration, sound and noise level, temperature, pressure, and carbon monoxide (CO) gas (Schmidt, Beigl & Gellersen 1999).

Many of the computing contexts such as the presence and quality of network connectivity typically can be sensed programmatically via software thus they normally do not require additional hardware.

Sensors must have these desired characteristics and requirements:

- reasonably small size,
- lightweight,
- low power consumption,
- unobtrusive,
- relatively inexpensive, and
- flexibility to be attached and detached as necessary depending on the application.

These requirements are necessary so that the sensors can be easily integrated into devices or various infrastructures without causing unnecessary attention from the user. This goal is in line with the vision of ubiquitous computing to make the technology invisible.

Most context-aware devices in smart home environments will most likely be specific purpose devices rather than general purpose devices, because generally one single device is used for one specific application only. Having many general purpose devices doing fixed specific tasks does not make much sense because it would cost more and it would be a waste of resources. For example, a wireless access point can be used as a router when connected to a desktop computer and then the desktop computer does the routing on software. However, there are specialized wireless routers that can do hardware routing, consume less resources (power), and can generate less heat and noise. In most

environments, having a dedicated wireless router is more preferable than using a general purpose desktop computers attached to wireless access points, because the flexibility of a general purpose computer, other than basic routing and firewalling functions, is rarely needed.

Since most of the devices are specific purpose then it is more sensible that these devices have only the required sensors for the specific applications. Some general purpose devices such as a PDA may have the ability to be expanded with many sensors, this is one of the reasons why it is important for some sensors to have the flexibility to be attached and detached as necessary depending on the application.

2.2.3 Managing Contextual Information

The raw output from various sensors may be in low-level formats, and they may be in different formats, thus making it harder to deal with in applications.

In order to make it easier for application developers to build applications, contextual modelling is introduced. Contextual modelling refers to the way we can represent contextual information in a structured and meaningful manner so that it can be accessed and used by applications that perform further computations. A context model can be accessed potentially by many applications at once. A standard specification explaining these models serve as an agreement to ensure consistency and correct assumptions between different parts of an application accessing the same context as well as different applications who can potentially be developed by many people in the same group or even in different groups.

In a ubiquitous computing environment there will be potentially many contextual information acquired by many heterogeneous devices. The contextual information will in turn be managed by many different special purpose applications that co-exist within the ubiquitous computing environment like a context-aware smart home. Interconnectivity and interaction between these applications is part of the basic requirements of ubiquitous computing. Some of these different devices and/or systems need to talk to each other for sharing information or to notify other applications regarding a certain condition that other applications may be interested in. Therefore it is important for these applications to agree on a common standard for modelling contextual infor-

mation and probably other relevant non-contextual information that needs to be shared. This means there are at least two parts to the standard: one is for modelling the contextual information, and the other one is for communication protocol, which could be between the applications using the same or different standards for modelling contextual information.

Different contextual information may have different requirements for modelling (Chen & Kotz 2000). How a location is represented and what information makes up location is different than how time is represented and what information makes up time. Location may consist of variables such as latitude and longitude, or a high-level symbolic information such as "Building 10, Level 4, Room 101". Time is usually represented by a combination of date, hour, minute, second, and timezone.

There are some existing efforts on modelling contextual information. Becker & Dürr (2005) describe the properties and requirements for modelling *location* as contextual information. The paper also describes geometric, symbolic, and hybrid location models. Held, Buchholz & Schill (2002) elaborates the requirements to a comprehensive context representation: structured, interchangeable, composable/decomposable, uniform, extensible, standardized. McFadden, Henricksen & Indulska (2004) developed the Context Modelling Language (CML) which is a fact-based context modelling approach. Harter, Hopper, Steggle, Ward & Webster (2001) argue that it is appropriate to model the environment using an object-oriented modelling technique because the real world environment is comprised of a collection of real objects. Others have used simpler approaches to model contextual information, such as using key-value pairs as the data structure (Salber, Dey & Abowd 1999, Schilit, Adams & Want 1994, Voelker & Bershad 1994, Maaß 1997). Other techniques include markup scheme models, graphical models (McFadden, Henricksen & Indulska 2004), Petri nets (Murthy & Krishnamurthy 2005), logic based models, ontology based models (Strang & Linnhoff-Popien 2004), relational representation, and situation abstraction (Henricksen & Indulska 2004) to represent generic contextual information. The main reason behind standardising the model is so that applications can share the same contextual knowledge thus enabling interoperation between the applications.

Ontology-based approaches have the potential to provide better support for interoperability and reasoning, which have significant importance in developing

context-aware applications. However, ontology-based approaches have some shortcomings: they are normally based on OWL Web Ontology Language¹⁷ which does not provide support for axiomatic rules thus limiting the types of reasoning that are possible, they do not address the issue of reasoning over imperfect context information, and the process of creating and extending context ontology is often complex and error prone. Henricksen, Livingstone & Indulska (2004) explain these shortcomings as a motivation to a hybrid approach, which they also describe in the paper. They create a mapping of concepts from CML (McFadden, Henricksen & Indulska 2004) to OWL DL¹⁸, then they transfer a sample context model in CML into an OWL representation based on the mapping. Their conclusion is that the standards used are still immature thus their approach cannot be validated fully.

I believe that each of the mentioned approaches has its own strengths and weaknesses. Some are suited to be used in one type of applications but not others. It would be overkill to use an overly complex but powerful modelling system, such as CML + OWL DL, for simple context-aware applications that do not require complex reasoning capabilities where a simple key-value data structure or a simple Extensible Markup Language (XML) application would suffice.

Keeping a History of Contextual Information

Chen and Kotz noted the importance of keeping a history of the acquired contextual information (Chen & Kotz 2000). Depending on the application and the context, it is probably ideal that only a subset of the collected data are stored due to storage space limitations. For example, when an application is only interested in a context *change*, then the application need only to store the last data collected to be compared with the current condition (certain applications need to take the average over a certain number of samples) in order to determine that there is a change in the context. However, in other applications, for example the ones that require learning/training, it may be necessary to store many samples of various contextual information. Naturally these applications may have higher requirements.

¹⁷<http://www.w3.org/TR/owl-ref/>

¹⁸OWL DL is a sub language of OWL which is named so due to its correspondence with Description Logics (DL). More information about OWL DL is available from the W3C guide to OWL website: <http://www.w3.org/TR/owl-guide/>.

2.2.4 Context Adaptation

After all the needed contextual information is acquired and stored in an accessible data structure, a context-aware application must somehow make use of that contextual information to determine what actions to take according to the application specifications. The process of making use of contextual information in order to change the behaviour or characteristics of a context-aware application is commonly known as *context adaptation*. In general there are two ways a context-aware application can adapt to the new or changing context.

- *Self action or self reconfiguration.* When an application becomes aware of a new context it can adjust its *own* characteristics relative to its user's pre-defined or learned preferences for that context. In some cases, the application developer can make some assumptions about the user's preferences, or the application itself can employ some machine learning techniques to learn about its user's preferences. The user may need to set in advance her preferences in certain conditions, especially for situations where it is difficult or impossible for the application to figure out the user's intention, or when the user wants to set some preferences explicitly. The preferences could be modelled as profiles (Leijdekkers & Gay 2005) to make it possible for the user to group together related preferences in a situation, and for the user and the application to distinguish between sets of preferences in different contexts.
- *Other action.* Instead of selecting profiles or adjusting the device's own settings based on the user's preferences, a context-aware application could also trigger other actions to be performed which do not involve changing anything on the device itself. Examples of such actions include, but not limited to, sending notifications (e.g. via SMS, email), grabbing user's attention by beeping, playing a tone, or displaying a warning message, sending messages to other applications. The latter example is actually a way for a context-aware application to interact with other context-aware or ubiquitous applications, when it the information might be of some interest to other applications it could be configured (or via automatic discovery and negotiation) to pass along that information.

Context adaptation almost always involves a varying degree of Artificial Intelligence (AI). Some of the relevant AI techniques that have been used include

rule-based inferencing (Zhang 2004), expert systems (Kwon, Yoo & Suh 2005), neural networks (Mozer 1998) and machine learning (Dey, Hamid, Beckmann, Li & Hsu 2004). Rule-based systems and expert systems are less capable to adapt to new situations because new rules need to be put into the system to deal with new situations. While neural networks and machine learning systems have more tolerance to new situations and can adapt better to changes in the context which the application has not previously seen before. It seems that neural network and machine learning techniques are more suitable for context-aware applications, especially complex applications that need to adapt to constantly changing environments.

2.2.5 Architecture of Context-Aware Applications

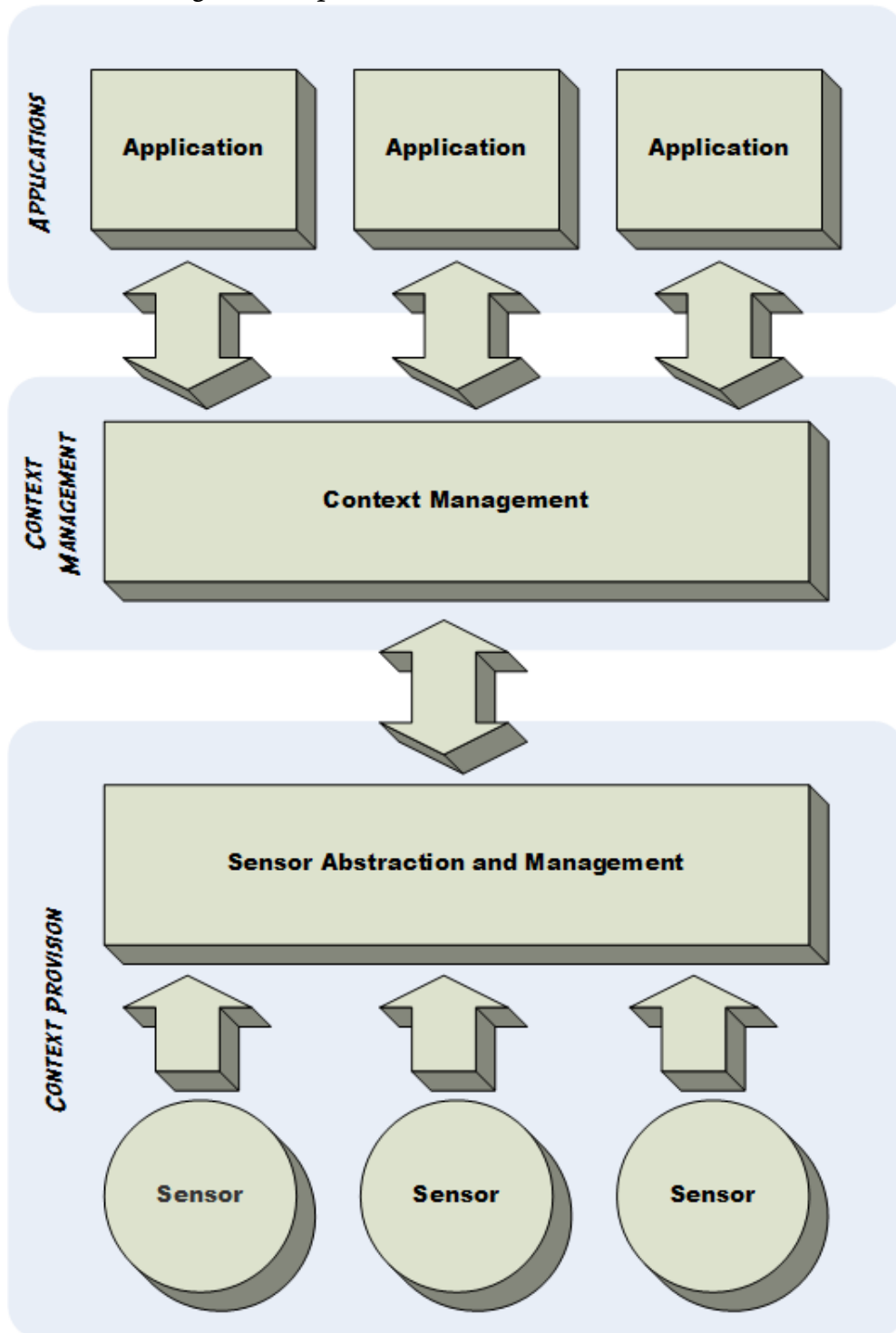
The core of typical context-aware applications follow roughly the same simplified architecture (or abstraction) as depicted in figure 1 on the next page. There are three distinct layers that facilitate separation of concerns.

The lower layer deals with context provision. It consists of all the sensors and other lower level services which are then managed by a framework of sensor abstraction and management layer. This type of framework normally consists of the software (i.e. drivers) necessary to communicate with different sensors. Each sensor may have its own driver, making it more complex to work with many different sensors from different vendors. It provides an abstraction to the upper layer to make it easier to acquire data from different sensors programmatically by hiding all the complexity in a framework.

The middle layer manages context. Basically it gathers raw information from various sensors through the sensor abstraction and management framework, then it processes that raw information and turn it into more meaningful high-level contextual information. Optionally, this layer may store a history of the acquired information into a database when an application requires such historical data. The context management layer may store the contextual information in some context modelling scheme, for example an XML-based format. Alternatively, it could be stored directly in a database and then the framework can provide different representations or models based on the same data. The context model is in turn accessed by various applications.

The upper layer consists of context-aware applications that build on top of

Figure 1: Simplified Context-Aware Architecture



the two lower layers. This layer is where context adaptation is performed. Context-aware applications retrieve contextual information from the context management framework, then use that information to adapt to the context (reconfigure itself) and perform actions (e.g. send notification, or adjust the room temperature). The applications themselves may be further divided into layers, for example, the User Interface (UI) layer that is concerned with directing output to the users via a display¹⁹, for example.

Chen & Kotz (2000) classify context-aware applications architecture into two categories: centralized approach and distributed approach. In a *centralized approach*, context management is done in a centralized server, then the applications or devices connect to the server to obtain context information. This approach allows knowledge sharing between applications, however it usually has scalability problem. In a *distributed approach*, the context information are decentralized, it is held in the applications or individual devices. This approach is ideal to maintain privacy between applications or between different users of the same application but using separate devices, and that no central server is required, however, the computational and communication requirements of the devices become significantly higher. For context-aware applications in *smart homes*, I argue that the centralized approach is more suitable because the applications should run in an integrated environment, i.e. the smart house itself, therefore, knowledge sharing is important, and privacy between applications is not an issue in most –if not all– cases.

2.3 Requirements and Challenges

Most of the requirements and challenges are general to context-aware computing, they are not specific to smart homes or to supporting independent living – which are the main topics of this paper, and will be discussed later in this paper.

Some of the requirements may actually become challenges themselves.

It is often impossible to address all of the challenges without sacrificing some of the benefits; there is almost always a trade-off. The trade-offs should be flexible depending on the particular applications or conditions. For example,

¹⁹Implicit input and output is preferred in context-aware applications to be as non-intrusive as possible. Refer to section 2.2 on page 6.

for Personal Area Networks (PANs) it is considered acceptable when the range of wireless connections are limited in order to save power, but for military purposes where the range and power of wireless connections are important then the extra weight for the additional power supply may be justified.

2.3.1 General Ubiquitous Computing Requirements

- *Mobility.* Aside from being everywhere, there are some devices that follow the users wherever they go, commonly known as mobile devices or wearable computers. Although not all mobile devices are wearable, the point is that they move together with the user. Mobility is considered crucial because motion is such an integral part of everyday life. Therefore, in order to make these devices "disappear" into the background the device must support mobility so that the user is not aware of the technology by its absence (Satyanarayanan 2001).
- *Scalability.* Normally scalability is often seen as support for bigger and more complex requirements, or "scale up". However, that is only half of the definition of scalability, the other half is making applications work in a small devices with limited resources and less complexity, or "scale down". Many of the devices that make up for the ubiquitous computing infrastructure are specific purpose devices that are normally small devices with limited resources.
- *Invisibility.* Invisibility of ubiquitous computing technology can be achieved when the user does not perceive it as a bother, the user feels like the technology is part of the everyday activities. This require careful considerations right from the design phase, combined with smart, intuitive, non-intrusive, and effective user interfaces. The design must balance between convenience and non-intrusiveness in the user interface as well as the physical and logical design of the technology itself.
- *Minimum interaction.* Ideally ubiquitous computing applications work by considering their and their users' context (i.e. they are context-aware), and function in a non-intrusive manner, without requiring the users to direct their attention to, or continuously input information into, the applications or devices. In other words, the applications keep the human interaction to as minimum as possible; ideally none. This will let

the applications to do their job in the background (i.e. become *invisible*) without distracting the user thus minimising the perceived burden associated with using traditional non-context-aware applications.

- *Heterogeneous environment* (Abascal 2005). Many devices that comprise the infrastructure of ubiquitous computing environment are made by different vendors with different characteristics. Some of them may interoperate easily, some require extra effort to interoperate, some may not even interoperate at all. When devices cannot interoperate with each other, it is often necessary to build an abstraction layer that can integrate the devices so that applications can access the devices indirectly via this abstraction layer. However it is not always possible because some devices may have a closed specification making it impossible to write one's own driver, for example. Some open standards need to be developed and adhered to by vendors and other relevant bodies if the best interoperability is to be achieved between the heterogeneous devices and applications.
- *Power consumption* (Want & Pering 2005). Devices have power requirements. When devices become ubiquitous then the overall power consumption requirements will also increase. When ubiquitous computing technology becomes ubiquitous and everywhere, the demand for generating power will be so great that it may not be possible to fulfill that demand if the power consumption caused by the technology is not managed properly from the design phase. Therefore it is important to minimise the devices' power consumption so the impact to the overall power consumption will be as minimal as possible.
- *Discovery and management*. The vast amount of devices in ubiquitous computing environments will pose a challenge of identifying, keeping inventory, and managing them. Want & Pering (2005) argue that pervasive or ubiquitous devices will require some wireless discovery techniques. Without proper discovery and management techniques it becomes tedious and practically impossible to be able to identify and track the physical and logical location of devices, propagate updates to select devices, perform maintenance on broken devices, and more importantly, develop powerful applications that rely on the ability to work on individual devices as well as integrate the different devices. There are some

protocols that can be used for automatic discovery of services, for example, JINI²⁰ and UPnP²¹.

- *Security and Privacy.* There are some basic principles to keep in mind when discussing computer security in general as well as in ubiquitous computing domain, namely confidentiality, integrity, accounting, availability, authentication, and security policies (Stajano 2002). If not designed carefully, ubiquitous computing applications may intrude on the users' (and maybe others') privacy. For example, improperly managed data gathered from identity recognition camera could be abused. Clear and well-defined security policy is essential for ensuring data confidentiality. Applications need to adhere to these security policies to manage who and what applications can access which data at which time, and so on. To ensure data integrity, some cryptographic techniques, such as checksum or hash algorithms, are often needed to ensure that the data transmitted during communication between the ubiquitous computing devices and applications is intact, not altered, and that its authenticity can be validated. Accounting in security context basically means keeping a record of who did what and when, this record could then be used for auditing or debugging. Ensuring availability means making sure that no one (or nothing) can cause a denial of service (DoS). When a DoS occurs, the system, or parts of it, becomes unusable. A simple example of DoS is radio frequency jamming. These security risks need to be addressed as appropriate depending on the type of applications. In applications such as wireless health monitoring and emergency detection, ensuring availability may become a critical issue.

It is also worth noting that some devices such as PDA and sensors may use wireless transmission, which uses a shared medium, for communication. Additional precautionary steps and measures may be required to ensure reasonable and acceptable security in a shared medium.

2.3.2 Context-Aware Computing Specific Challenges

Salber, Dey & Abowd (1999) describes the following four main difficulties in using context information in computing applications.

²⁰<http://en.wikipedia.org/wiki/JINI>

²¹http://en.wikipedia.org/wiki/Universal_plug-and-play

- The context information is often acquired from unconventional sensors. Although the term unconventional is debatable, but generally it refers to sensors which are not commonly found in day to day computing environments. Examples: GPS, biosensors, RFID, pressure sensitive floors. Unconventional sensors are not easily accessible because they may not be easy to find, and may require a relatively high cost to obtain.
- Low level data acquired from the sensors needs to be abstracted into a high-level representation in order to be useful. For example, temperature reading in degree Celsius may need to be abstracted into cold, cool, medium, warm, hot. An application may need to abstract a higher-level information using two or more sensor data, for example, in addition to temperature the may need humidity information to determine a room's comfort level.
- The context information may be acquired from multiple distributed and heterogeneous sources. It is uncommon that a context-aware application requires more than one sensors in order to understand a certain context. The temperature and humidity example above can be used again here.
- Context information is dynamic in nature. A context-aware application often needs to detect changes in real time and adapt to constant changes. For example, a mobile location-based application being used while in motion needs to handle the constantly changing location of the user and adapt accordingly in real time by giving the user the needed information most relevant to the current location.

In addition to these difficulties, there are other challenges associated with developing context-aware applications.

- One of the main challenges in context-awareness is *understanding user intent*. This process is often crucial during context adaptation phase. Once a context-aware application understands the current context, then it must decide how to adapt to that context best. An application can adapt itself (self reconfiguring), or adapt its own or the user's environment (send messages to other application). The application has to make

its best guess on what kind of behaviour is desired from the application by the user, given the current context. Some of the AI techniques used in context-aware applications relies on learning from past situations to result in the *same behaviour*, however, the application may not do well when a *new behaviour* is expected in new or previously learned situations.

- *Multidisciplinary*. Building context-aware systems, such as smart homes to support independent living, usually involves and requires knowledge in computer science, psychology, Human-Computer Interaction (HCI), architecture, mechanical and electrical engineering, social sciences, and maybe others. Depending on the domain of the context-aware application, an expert knowledge in the domain may also be required. The development of such systems is normally done by a team of many people with diverse backgrounds, including cultural backgrounds.
- *Cost*. Some of the devices, such as biosensors, used to build context-aware computing infrastructure are not common, so they generally cost more than other common computing devices. Some computing or communication infrastructure simply does not exist yet, for example, WiMax is not publicly available in Sydney at the time this paper is written. This creates a barrier of entry to people who wants to develop context-aware applications that require specific hardware or infrastructure support. Hence the design of applications are often dictated by what devices are easily accessible to the developers. However, I believe that when the devices become ubiquitous enough, they will be mass produced thus driving the cost of production, and eventually end-user price, lower. I argue that the devices will become ubiquitous when there are many demands for them. These demands are in turn created by context-aware applications that are actually practical and useful.

Some of the challenges described above become requirements for some context-aware applications.

2.4 Current Research and Prototypes

The following is a short list of frameworks and applications that are relevant to context-aware computing as a starting point for others interested in devel-

oping context-aware applications.

2.4.1 Context-Aware Frameworks

The following frameworks are general purpose framework that can be used as a foundation to develop context-aware applications.

*Context Toolkit*²² (Salber, Dey & Abowd 1999) is one of the earlier efforts on developing a reusable context-aware toolkit framework. The toolkit is built around what is called *context widgets* whose design is based on the concepts of Graphical User Interface (GUI) widgets that hides the complexity of dealing with various low-level sensor devices and information by providing an abstraction layer. The design addresses some of the issues described in section 2.3.2 on page 22. From a programmer's point of view, a context widget is somewhat similar to an object in object-oriented programming methodology. A context widget has *attributes*, e.g. location, identity, timestamp, and *methods* or *callbacks*, e.g. `PersonArrives(location, identity, timestamp)`. To better adapt to dynamic context changes, the toolkit framework allows context information to be acquired in real-time whenever there is a request, as well as allowing applications to subscribe to only some information in which it is interested. The initial version of the toolkit has limited context widgets (Presence, Identity, and Activity) that make up the core of the toolkit.

a CAPpella (Dey et al. 2004) is a programming by demonstration context-aware framework geared towards end users instead of programmers. This framework uses machine learning techniques to recognize behaviour or situation. End users demonstrate a behaviour or situation of interest to the system that may have video cameras, microphones, RFID readers and other kinds of sensors. After the demonstration is finished, the user prunes the captured data in the system so that only relevant information is used by the system for learning. Then the user uses that captured information to train the system. The user needs to repeat the training process a number of times so that the system can get better recognizing the behaviour or situation. The user can test how well the system performs after a demonstration is finished, then the user can decide whether to train the system with the recently captured data.

²²<http://www.cs.berkeley.edu/~dey/context.html>

Java Context Awareness Framework (JCAF) (Bardram 2005) is a J2EE-inspired runtime environment and programming framework designed for developing context-aware applications. The event-based runtime environment aims to support distributed and coordinated services, access control, and modular infrastructure to allow for extensions. Some unique features of the JCAF Application Programmer Interface (API) are that it aims to be semantic-free (generic purpose), and that it takes quality of context information into account. JCAF provides event listeners that applications can subscribe to. The event listeners will call the appropriate callback functions whenever there is an interesting event, e.g. a context change. What to do with the context information is left up to the applications. In other words, JCAF does not have AI built into it.

There are other frameworks and toolkits that are more specific. They focus on certain areas of context-awareness, so they may need to be integrated into, or combined with, a bigger framework to complement each other functionalities. These frameworks include Mayrhofer's context prediction framework (Mayrhofer 2004) that focuses on recognising and predicting high level context information from low level sensor data. The framework is intended to run locally in a limited resource devices without requiring an infrastructure before hand. Some unique features of this framework are that it uses heterogeneous feature vectors²³ for context recognition, it uses Growing Neural Gas algorithms (Fritzke 1994, Fritzke & Leen 1994) for machine learning and Active LeZi algorithm (Gopalratnam & Cook 2003) for context prediction. Nishigaki, Yasumoto, Shibata, Ito & Higashino (2005) proposes a framework based on a natural-language-like rule language called *CADEL* (short for context-aware rule description language) which are targeted for end users.

Other frameworks which are more relevant to building context-aware applications for smart homes are described in section 3.4 on page 30.

2.4.2 Context-Aware Applications

Context-aware applications are developed for a broad spectrum of domains. Applications like *Cyberguide* (Abowd et al. 1997), *GUIDE* (Cheverst, Mitchell & Davies 1998), and *COMPASS* (van Setten, Pokraev & Koolwaaij 2004) are

²³Feature vector is a pattern recognition term for an n-dimensional vector of features extracted from raw data for further processing.(Wikipedia 2005b)

geared for tourism. Health related context-aware applications include *context-aware pill bottle and medication monitor* (Agarawala, Greenberg & Ho 2004), and other applications to help the elderly especially in smart homes, such as the ones described in section 4.4.1 on page 37. Chen & Kotz (2000) survey various context-aware *mobile computing* applications such as Active Badge (Want et al. 1992), Active Map (Weiser 1993), Shopping Assistant (Asthana, Cravatts & Krzyzanowski 1994), and others. In typical desktop computer applications we often encounter context-sensitive help. The Hypertext Transport Protocol, specifically HTTP/1.1²⁴, allows automatic content representation depending on the client's preference using content negotiation mechanisms.

A list of other context-aware applications may be found in <http://users.tkk.fi/~mkorkeaa/doc/context-aware.html> and http://www.icta.ufl.edu/projects_B.html.

3 Smart Homes

3.1 Vision of Smart Homes

A simplified view of a smart home could be merely a localised version of the ubiquitous computing vision at the home context, where most people spend most of their time, where technology becomes pervasive and invisible to the home occupants because *the technology interacts closely in the most natural ways with the occupants* (Meyer & Rakotonirainy 2003). The goal of smart homes is simple, to make the quality of living at home better by making everyday activities at home more intuitive, more enjoyable, more convenient, safer, easier, faster, and better in many other ways.

3.2 Context-Awareness and Smart Homes

Intille (2002) argues that *the home of most value in the future will not use technology primarily to automatically control the environment but instead will help its occupants learn how to control the environment on their own*. This statement is in line with Erickson's (2002) opinion to keep humans inside the control loop instead of keeping them out of the control loop. However, I believe that the

²⁴Section 12 of RFC 2616, <http://www.faqs.org/rfcs/rfc2616.html>.

most useful smart home of the future will be somewhere in the middle of the two extremes, a home that fully controls its environments automatically, and a home that lets the occupants control the environment on their own. A smart home needs to be as flexible as possible in order for the technology to become invisible to its occupants. There are situations where it is desirable for the smart home to act autonomously without bothering the occupants but there are also situations where it is imperative for the smart home to let the occupants decide what to do, or at least let the occupants know in a non-intrusive way so that the occupants do not perceive the notification as a bother. This is where context-awareness comes into play. The system should be smart enough to know when to non-intrusively inform the occupants and when to make its own decision. It is somewhat analogous to having a (human) personal assistant.

Smart homes are often analogous to home automation. By adding context-aware computing to smart homes, I believe that smart homes can offer *intelligent* home automation. In other words, context-awareness is the key to the intelligence; it is the "smart" in "smart home".

3.3 Common Infrastructure in Homes to Support Context Awareness

Smart homes usually have less sophisticated infrastructure compared to offices. But there are some infrastructure that are common to both. The components that make up the usual infrastructure are most likely heterogeneous, they have different purposes, possibly made by different vendors, and they have different characteristics. Some of the components may support output interfaces or connection to other devices, which is necessary to make the infrastructure integrated and interconnected so that the components in it can interact with each other.

Interoperability between the infrastructure components is one of the key factors to the smart home platform. Without interoperability it will be a tedious, time consuming, potentially complex, error prone, and challenging task to develop applications that make use of the non-interoperable or only partially interoperable infrastructure components. There are efforts to address this interoperability, especially for networked home appliances, for example AMI-

DEN (Minoh & Kamae 2001), which uses a peer-to-peer architecture.

Context-awareness can be applied to many common infrastructure components in the homes. Some examples are enumerated below to give a general idea.

- Environment control: HVAC (heating, ventilation, air conditioning), humidity and temperature control, lighting.
- Safety and security: alarms, smoke detectors, access control devices, surveillance systems.
- Appliances: refrigerators, microwaves, ovens, stoves, dishwashers, laundry machines and dryers, vacuum cleaners.
- Communications:
 - ★ Wired Local Area Networks, e.g. 10Base-T Ethernet using.
 - ★ Wireless Local Area Networks (LANs), e.g. Wi-Fi (IEEE 802.11), Infrared, and Bluetooth.
 - ★ Wireless Wide Area Networks (WANs), e.g. WiMax (IEEE 802.16), EDGE (Enhanced Data Rates for GSM Evolution), and EV-DO (Evolution Data Only).
 - ★ Wireless voice networks, e.g. GSM, CDMA, and UMTS.
 - ★ Wired voice lines, or POTS (plain old telephone system).
 - ★ Internet access, e.g. ADSL and Cable TV.
 - ★ Cabling for audio and video entertainment systems, and surveillance systems.
- Fixtures: doors, windows, walls, furnace, ceilings, fences.
- Outdoor: garden, outhouse
- Rooms: bedroom, bathroom, living room, kitchen, garage, storage room, kids room, study.
- Resources: electricity, gas, and water.
- Furniture: bed, sofa, tables, chairs.

- Entertainment systems: audio, video, and game consoles.

There are other infrastructure which are less common. For example, lifts or escalators for the elderly or people with disabilities, X10 devices for home automation and home security.

3.4 Frameworks and Architecture

The following are some of the frameworks that can be used for developing context-aware applications in smart home environments.

- *HomeOS* (Rosen, Sattar, Lindeman, Simha & Narahari 2004) is a Java-based infrastructure framework for developing and deploying applications in smart homes. It focuses on integrating the devices and acting as an intermediate abstraction layer as well as a runtime environment for the applications. Just like a regular operating system (OS), HomeOS acts as a bridge between devices and applications. HomeOS is not specifically designed for context-aware applications only, regular applications may be deployed on HomeOS as well. HomeOS relies on the use of RFID infrastructure for identification and location tracking of objects in the system. For its UI layer, HomeOS uses a specifically designed XML-based UI description language.
- *Open Services Gateway Initiative (OSGi)*²⁵ (Marples & Kriens 2001) framework. This framework is not a context-aware framework *per se*, but it is relevant in developing smart homes. The Java-based open-specification OSGi framework is a standardized, component oriented, computing environment for heterogeneous networked services. The OSGi framework maintains a registry of available services throughout their life cycles. In a smart home and other pervasive space environment there will most likely be many devices from different vendors that need to interoperate with each other or need to be integrated into applications. The OSGi framework addresses the integration and interoperability issues, and potentially other management and software engineering issues in a

²⁵<http://www.osgi.org>

context-aware pervasive space such as a smart home. The OSGi framework is used by the Gator Tech Smart House project (refer to 4.4.1 on page 37) and also in other research (Choi, Shin & Shin 2005).

There are other research papers that describe architecture design (Huebscher & McCann 2004, McFadden & Indulska 2004, Choi, Shin & Shin 2005, Haya, Montoro & Alamán 2004, Lee & Chung 2004). However, they do not discuss about any specific framework, or they use one of the existing frameworks mentioned above. For example, the architecture described by Choi, Shin & Shin (2005) is built on top of the OSGi framework.

Refer to section 2.4.1 on page 25 for more general context-aware frameworks that are not specifically geared towards smart home applications.

Most of the architecture design (Huebscher & McCann 2004, McFadden & Indulska 2004, Choi, Shin & Shin 2005, Haya, Montoro & Alamán 2004, Lee & Chung 2004) are similar in general. They use a multi-layered approach, similar to figure 1 on page 18, to allow separation of concerns and abstraction. Huebscher & McCann (2004) put more emphasis on how to handle context adaptation²⁶. It also discusses Quality of Context (QoC), service discovery, and the problem of trust between context-aware applications.

3.5 Existing Applications in the Smart Home

3.5.1 Commercial Applications

Some of the applications in smart homes are already commercially available in the market albeit not very common.

*LG HomNet*²⁷ from LG Electronics is a range of home appliances with advanced communications capabilities. Some of the appliances include the following.

- Internet enabled digital TV, combined with a digital set-top box, that can also act as display terminal for monitoring and controlling other networked home appliances.

²⁶Refer to section 2.2.4 on page 16 for more information on context adaptation.

²⁷<http://global.dreamlg.com/>

- Internet enabled refrigerator with an LCD display that can act as a home server or gateway, home monitoring and controller, and entertainment outlet (music, TV, video). It can also be accessed from the Internet via mobile phones, PDA, or other computer systems. The refrigerator has an automatic self-checkup and reporting functionality built into it that can report problems to the service center.
- Internet enabled washing machine that can be controlled and monitored via the Internet. It can store and display messages sent via the Internet, download new washing programs, and perform self-checkup and reporting to the service center.
- The air conditioner and the dish washer has similar capabilities to the washing machine.
- Internet enabled robot vacuum cleaner that can be controlled and monitored from the Internet.
- Gas oven range that can download recipes from the Internet, perform self checkup and reporting to the service center, and store and display messages sent via the Internet.
- Microwave oven that has similar capabilities to the gas oven range, and can be monitored from the Internet.

Vendors such as x10.com²⁸ and SMARTHOME²⁹ sell various X10 devices ranging from cameras, home automation controllers, switches, motion sensors, lamp modules, and so on. The X10³⁰ standard is one of the more popular standards used for home automation. It uses signals embedded in home electricity to allow communication between devices and controllers. There are other vendors that have their own proprietary standard for home automation. Siemens Smart Home Technology³¹, Home Automation, Inc.³², and Echelon (with its LonWorks standard³³) are some examples of such vendors. There are also other open standards besides X10, for example, Konnex³⁴.

²⁸<http://www.x10.com/>

²⁹<http://www.smarthome.com/>

³⁰http://en.wikipedia.org/wiki/X10_%28industry_standard%29

³¹<http://www.automation.siemens.co.uk/main/business%20groups/et/smart%20homes/>

³²<http://www.homeauto.com/>

³³<http://en.wikipedia.org/wiki/LonWorks>

³⁴<http://www.konnex.org/>

Most of the commercial devices and applications are probably more connected to the home, to each other, and to other system, hence they are more accessible and more convenient to use. However, they lack intelligence or the actual "smart" component, even though they use the term "smart" in their names.

3.5.2 Current Research Prototypes

There are various research groups that have created prototypes of their smart home projects. Some of the projects, such as the *Aware Home Research Initiative* from Georgia Institute of Technology, the *Gator Tech Smart House* from the University of Florida, the *Smart Medical Home* from Center for Future Health, University of Rochester, and the *MARC Smarthouse* from the University of Virginia, are discussed in more details in section 4.4.1. These projects have applications that are more relevant to the topic of this paper, supporting independent living for the elderly or for people with disabilities.

3.6 Issues Specific to Context-Awareness in Smart Homes

Earlier research on context-awareness focus on applying context-awareness in an office or work environment (Want et al. 1992, Ward, Jones & Hopper 1997, Weiser 1991). Context-awareness in a home environment has different characteristics than in an office or work environment, thus the requirements are also different. Offices normally have better technology infrastructure than homes. Social interactions in office environments tend to be more formal, structured, task-oriented and geared to optimize productivity, where in homes people are more free to do to decide what to do, when, where, and who to involve (Meyer & Rakotonirainy 2003). Therefore the developing for context-aware applications in smart homes could bring more challenges than in offices.

In keeping the vision in line with the general ubiquitous computing vision, then it is necessary to make context-aware applications as non-invasive as possible, regardless of the environment. The residents of the smart homes should not be required to put extra efforts or burdened with inconveniences when doing their daily activities as if they were living in non-smart homes. Careful UI and interaction design is required to produce intuitive context-aware applications for use in homes.

Integrating context-aware applications into existing home infrastructure could pose some problems. Some new infrastructure required by the context-aware applications may need to be installed. Depending on the type of the infrastructure and other factors, like the age of the house, local housing regulations, and so on, it is not always possible. Some incompatible context-aware applications from different vendors may not be possible to installed, for example, context-aware applications that rely on RF-based wireless infrastructure cannot be used together with context-aware applications that uses various medical and health monitoring systems due to interference.

4 Support for Independent Living

4.1 About Independent Living

In general, living independently means being able to live and take care of oneself in doing activities of daily life without depending on other people. Maintaining an independent lifestyle is important to improve the quality of life of individuals, especially the elderly (McFadden & Indulska 2004). It is also said that decreasing independence can have a profound impact on the quality of life of family caregivers (McFadden & Indulska 2004). However, in some cultures such as some Asian cultures, the impact of maintaining independence on the quality of life may not be as big as in other cultures because for the elderly people being dependent to their family or extended family is an expected phase of life and viewed as a sign of caring while encouraging the elderly to live independently may be seen as abandonment.

It is common that elderly individuals have some disabilities ranging from reduced cognitive and memory abilities to various physical and psychological disorders, impairments or other degenerative conditions. Some of these conditions may be age-induced and some may not be caused by ageing. These conditions or disabilities tend to limit the individuals from doing their daily activities, even when doing simple tasks such as walking, going to the toilet, cooking, and so on. These limitations often require elderly individuals to be attended by a caregiver, family member or otherwise. In a sense, the elderly individuals become dependent to their caregivers.

Some people with certain disabilities may be in a similar situation as the el-

derly individuals described above, where they also depend on their caregivers to perform their daily activities.

Supporting independent living means empowering the elderly individuals and people with disabilities to become more independent in performing their daily activities. From a medical perspective, supporting independent living may improve the quality of life of the affected individuals, and from a socio-economic perspective it could mean reduced healthcare costs by not requiring placement in full-time care facilities (McFadden & Indulska 2004). It could also put the minds of loved ones at ease knowing that their elderly relatives are able to take care of themselves. There are some technologies (Mynatt, Rowan, Craighill & Jacobs 2001, Agarawala, Greenberg & Ho 2004) that allow the smart home to provide limited feedback information regarding the occupants health condition and activity levels remotely to their caregivers, e.g. a younger relative or a physician.

4.2 Motivation for Context-Aware Homes that Support Independent Living

Smart home technologies are quite new, thus they have not been widely adopted, mostly because people do not appreciate the value or they do not see that the value is justifiable. Based on my personal observation, people seem to be quite happy with the way they live now even without these smart home technologies, therefore the cost would not be justified for the value they would be getting if they were to buy smart home technologies.

Elderly individuals, who often have dependencies on other people to take care of them (e.g. a personal caregiver), benefit the most from context-aware applications (Sanders 2000, Mihailidis & Fernie 2002) that can change their living style to become independent. This is especially true in western countries where the culture suggests independent lifestyles in order to have a higher quality of life.

The percentage of Australian residents aged between 45 and 64 years has grown from 21.01% to 24.16% since June 1996 to June 2004, and from 12.03% to 12.95% for residents aged 65 and above³⁵. These figures indicates that there is a grow-

³⁵<http://www.health.gov.au/internet/wcms/publishing.nsf/Content/health-statistics-june2005-table12>

ing potential market for context-aware smart homes.

4.3 General Requirements

To support independent living, an application must support an individual's health-care needs and activities of daily living (ADL), while extending social interaction, environment control, and information flow to care giver and family communities. (McFadden & Indulska 2004)

The needs of elderly individuals and individuals with disabilities are diverse. Mihailidis & Fernie (2002) argue that *there is no one-size-fits-all approach to address the needs of every individual; instead, design must take into account the unique needs of each user, and fluctuations in those needs*. I believe that context-aware applications should avoid making assumptions about the user, the user's conditions, and the context of the user and the application at the time of usage, unless absolutely necessary, and that context-aware applications should instead maximise its capability to acquire contextual information implicitly and somehow utilise that information to dynamically assess the user's current needs based on the particular contexts.

Tracking location of people and objects becomes an integral aspect of context-aware applications since a majority of the applications rely on this knowledge. In a smart home environment, it is more relevant to track position indoor as opposed to outdoor, so a technology like GPS will not work because GPS signals cannot be received indoors and the level of accuracy offered by GPS may not be sufficient (Wikipedia 2005c). There has been many efforts to address indoor location tracking (Want et al. 1992, Ward, Jones & Hopper 1997, Bahl & Padmanabhan 2000, Yamazaki 2005), ranging from RFID tags to force-sensitive floor sensors. The accuracy or precision of location requirements vary from application to application. In some cases, simply locating a person or an object to room-level, e.g. "in which room?", is sufficient. In other cases, it might be necessary to get more precise location.

The hardware and software infrastructure also has some requirements common to most applications such as interconnectivity and interoperability between devices and applications (refer to 3.3 on page 28), and some middleware to glue the devices and the applications together by acting as an intermediary framework that provide a well-understood and well-documented pro-

gramming abstraction of the heterogeneous devices and services available to the application developers.

4.4 Applications to Support Independent Living

4.4.1 Existing Applications

At the time of this writing there are no known commercially available context-aware application that directly support independent living targeted for the elderly. There are, however, several projects and prototypes that are actively researched, and are listed below. This list may not be a complete list.

- The *Aware Home Research Initiative* (Kidd, Orr, Abowd, Atkeson, Essa, MacIntyre, Mynatt, Starner & Newstetter 1999) is probably one of the most well-known context-aware smart home research prototypes. It supports many smaller sub-projects, such as the *Aging in Place*³⁶ project that focuses specifically support the elderly to live independently without having to move to a fulltime care institution or home. The research home is a living laboratory where people can actually live and researchers can test and monitor their research. It is equipped with various different sensors, e.g. force-sensitive floor for location tracking and identification, ultrasonic and RF-based sensors, and vision sensors, as well as broadband connections for internal interconnection and connection to the Internet. The project has three areas for interface design and sensing technology research (Kidd et al. 1999) which were originally:
 - ★ supporting social connections between elder parents and their adult children promoting peace of mind for family members,
 - ★ supporting everyday cognition by augmenting aspects of memory that decline with age and planning capabilities of elder residents,
 - ★ sense and identify potential crisis situations so that appropriate outside services can be contacted as needed,

but later on it was revised (Mynatt et al. 2001) to:

- ★ crisis recognition

³⁶<http://www.gatech.edu/innovations/futurehome/index.php>

- ★ everyday cognitive support
- ★ awareness of daily life and long-term trends

The *Digital Family Portraits (DFP)* (Mynatt et al. 2001) is a result of one of the areas of research, supporting social connections with extended family members. The DFP is a portrait of the elderly family member which is a qualitative visualisation of daily life of the person in the portrait. The photograph is left as is, but the photo frame shows some icons that represent various current conditions, as well as their history so that possible developing trends can be easily noticed. The information shown include health, environment, relationships, activity and events. The DFP is meant to be placed in a fireplace mantle in the remote extended family member (e.g. the adult children of the elderly), but for privacy reasons a local copy is also provided in the elderly home as well so they can see for themselves what information is being communicated to the other side.

Déjà Vu Displays (Tran & Mynatt 2003) are passive displays which only serve as output media and require no explicit input. Their main purpose is to help memory recall of recent activity on a continuous basis. The initial study (Tran & Mynatt 2003) uses déjà vu displays to augment the kitchen environment by providing a collage of cooking action sequence of images recorded by video cameras, such that the collage can be referenced at anytime by the cook.

The *Gesture Pendant* (Starner, Auxier, Ashbrook & Gandy 2000) is a wearable infrared vision sensor in the form of a pendant worn like a necklace that can detect the wearer's gestures to perform tasks such as home automation (for example, to change the television channel, turn up the volume, turn on the lights, etc.), and medical monitoring.

- *Gator Tech Smart House*³⁷ (Helal, Mann, El-Zabadani, King, Kaddoura & Jansen 2005), from the University of Florida's Mobile and Pervasive Computing Laboratory³⁸, is a laboratory-house which is designed to assist older persons in maximizing independence and maintaining a high quality of life using various pervasive computing technologies to create a supportive and assistive environment for the elderly and the disabled.

³⁷<http://www.icta.ufl.edu/gt.htm>

³⁸<http://www.icta.ufl.edu/index.htm>

There is an online interactive visual tour³⁹ available showing features of the smart house such as:

- ★ smart cameras for security (e.g. surveillance, motion detection) and image analysis (e.g. face recognition),
- ★ smart blinds for controlling sunlight and privacy,
- ★ acoustic-based ultrasonic location tracking,
- ★ smart floor for location tracking and fall detection,
- ★ smart mailbox to sense mail arrival and notify the resident,
- ★ smart front door utilising RFID sensors for keyless entry, microphone, text Liquid Crystal Display (LCD), camera, automatic door opener, electric latch, and speakers,
- ★ smart display that is used as output and communication, or TV monitor, supports follow mode for communication so that the message is displayed on the display nearest to the recipient, the display can also be used for video/audio conferencing so that such things as remote dining with friends in another smart house is possible, there is also a smart projection system that can use walls as a projection screen to display information with a moving projector where an LCD/plasma displays are not available or out of sight,
- ★ smart phone that acts as a remote control to all appliances and media players, as well as acting as a Smart Home Agent while away from home (to convey reminders, notification, etc.),
- ★ smart closet that can give suggestions on what to wear depending on the outdoor weather condition, and sense clothes worn by the resident from the embedded RFID tags and notify the person if the clothing is unsuitable for the weather,
- ★ smart bed that can monitor sleep pattern, so that any sleeping disorders can be detected,
- ★ smart bathtub and shower that can regulate the water temperature and remembers each residents' preferences and set the temperature according to who is in the bathtub/shower,
- ★ smart mirror, a regular mirror that can also act as a smart display,

³⁹<http://www.icta.ufl.edu/gatortech/index2.html>

- ★ smart soap dispenser that can monitor the residents cleanliness and also give notification when a refill is needed,
- ★ smart toilet that monitors the occupant's biometrics (body weight, temperature, etc.), toilet paper sensor, flush detector,
- ★ smart microwave that can read RFID tags on the items that the resident wants to cook, and provide video instructions on a separate monitor located near the microwave, the resident will be notified on the speakers as well as the monitor when the cooking is finished,
- ★ smart refrigerator and pantry that monitors food availability and consumption, detects expired food items, creates shopping lists automatically, with integrated meal preparation advisor based on items in refrigerator and pantry,
- ★ smart laundry, in combination with the smart closet, notifies the home owner when to do laundry and helps sort the laundry, might give warnings if white and colored clothes are put together,
- ★ the garage has a driving simulator that focuses on older driver screening and evaluation, and approaches to remediation of conditions or environmental factors that contribute to unsafe driving, as well as counseling individuals contemplating driving cessation, the goal is to assist elders in driving safely, longer,
- ★ smart plugs (El-Zabadani, Helal, Abdulrazak & Jansen 2005, Helal et al. 2005) equipped with RFID low-cost readers so that the system can sense electrical devices plugged into the smart home,
- ★ smart leak detector to detect water leaks from washing machine, dishwasher, water heater in the garage and in the kitchen,
- ★ smart stove that monitors stove usage and notifies the occupants accordingly if the stove has been left on by accident,
- ★ home security monitor to monitor doors and windows to check which ones are open, closed and locked,
- ★ emergency notification system to track potential emergency situations that can query the residents and call for outside help if necessary,
- ★ cognitive assistant to augment memory of the residents by providing audio and visual cues regarding appointments, medications, etc.

This smart house is a second-generation research after Matilda's Smart House (Helal, Giraldo, Kaddoura, Lee, El-Zabadani & Mann 2003) which was intended to prove the feasibility and usefulness of programmable pervasive spaces as assistive environments.

The developers emphasises the Gator Tech Smart House as a *programmable pervasive space* as a runtime environment and a software library that tries to address these challenges:

- ★ easy integration and interoperability between heterogeneous devices in the pervasive space,
- ★ open specification architecture thus allowing easy extension and development by third party,
- ★ ability to evolve as new technologies emerge.

The middleware architecture is built around the Open Services Gateway Initiative (OSGi) framework (Marples & Kriens 2001) which allows registering and discovery of, as well as maintaining, available services. It will, in turn, allow application developers to create composite services, which become the applications in the pervasive space, from many basic services. The sensors and actuators register themselves to the system (i.e. the OSGi framework) by automatically sending an OSGi service bundle definition file normally located in the device's Electrically Erasable Programmable Read-Only Memory (EEPROM), that may contain the driver for the device and a definition of available methods, when the device is turned on. The OSGi service bundle definition files are prepared by the developers of the sensors or the actuators.

- *MARC Smarthouse*⁴⁰ from the University of Virginia's Medical Automation Research Center (MARC). *The MARC Smarthouse technologies are passive and unobtrusive technologies for monitoring elders' activities, designed with privacy and security in mind.* It puts an emphasis on the medical and health aspects of the smart home residents, which are targeted for elderly individuals. The website claims that the sensors used are non-invasive, there are no cameras or microphones. The system monitors the residents activity, store the collected data and analyse the data for

⁴⁰<https://smarthouse.med.virginia.edu/>

understanding the residents health and activity levels so that any unusual levels can be reported and be acted upon if necessary, e.g. in an emergency situations like an accident where the activity levels are low over a prolonged period of time. The system can also generate reports that can be accessed remotely via the Internet by caregivers, health care providers and the residents themselves. An example of the reports and the kinds of information available can be viewed at <https://smarthouse.med.virginia.edu/index.php?content=202>.

- *New Songdo City*⁴¹ (O'Connel 2005), South Korea. New Songdo City is a new city currently in early development stages, built from scratch on top of a landfill located 40 miles from Seoul. This new city is planned to be a ubiquitous city (*U-city*) that supports sharing of data between major information systems from the government, medical, businesses, residential, and so on. Computer systems are built into the buildings including houses, offices, streets, parks, and other infrastructures, hence ubiquitous. Among some of the planned ubiquitous computing technology that will be used are smart cards that can be used as a house key, transportation access card (subway, bus), electronic wallet (e.g. pay a parking meter, see a movie), borrow a free public bicycle, receive credit when tossing a bottle in a recycling bin, and so on. The smart card will be anonymous, it will not be linked to the holder's identity, and it can be replaced easily if lost. Naturally, the networking infrastructure will also be provided to support Internet access, interconnection between information systems, video and audio conferencing, wireless access to digital content and property from anywhere in New Songdo City. There will be plenty of opportunities to put various smart home technologies into the residential homes, for example, force-sensitive floors in the homes of elderly people that can detect falls and contact outside help. South Korea's privacy laws and regulations are not as strict as in other countries, and the Korean culture itself seem to expect less privacy with new technologies, therefore it would be easier for the South Korean market to be successful early adopters of new technologies such as the ubiquitous New Songdo City. The project was announced and started in 2005 and is expected to complete before the end of 2015.

⁴¹<http://www.new-songdocity.co.kr/>

- *Context-Aware Pill Bottle and Medication Monitor* (Agarawala, Greenberg & Ho 2004) is a medication reminder system especially for elderly individuals. It consists of three distinct parts. The first part is a stand containing RFID tag reader, light indicators, a button and a network connection to a computer. This stand will read the medicine identification from special medication bottles equipped with RFID tags. The stand will then send that identification to the computer containing an XML database of recording information regarding the frequency and duration the medication should be taken, warnings, and so on, so that the system can notify the user accordingly. The second part of the system is a remote display or monitor meant to be located in the home of the elderly individual's caregiver (e.g. relative). This component performs one specific function of a similar device, the *Digital Family Portraits* (Mynatt et al. 2001). The third part is the computer that keep a record of the medication history, and it can generate reports accessible over a web-based interface (e.g. via the Internet). This early prototype of context-aware pill bottle and medication monitor system has various drawbacks, especially: heuristic inaccuracy in detecting whether the user has actually taken their medication (it is merely based on an assumption), reliance on inaccuracy, mobility (the device location is fixed), and some user interface and usability issues (not easily understandable, may be annoying, etc.) The authors have clearly stated that this prototype is not an ideal solution, it is only a research prototype used to better understand context-awareness in the domain of the elderly and medication compliance. A similar case study using the *a CAPpella* framework is described in (Dey et al. 2004).
- *Gate Reminder* (Kim, Kim, Park, Jin & Choi 2004) is a research prototype of a reminder in the form of a home appliance display. It is meant to be located near the front door of the house so that the house residents can be warned if they forget to bring their house keys, car keys, mobile phones, wallets, and so on. The Gate Reminder can also provide messages to the residents about things that they like to be reminded of, for example, doing the laundry, bringing an umbrella if the weather forecast says it would rain, returning a rental DVD, and so on. The system relies on detecting the identity of the user (with a camera and/or RFID tags) to provide the context, along with other sensed information like RFID

scanners for checking missing items (keys, wallets, mobile phones) before leaving the house, and easily obtainable information, such as time. In the prototype system, the residents need to input some information into the system manually to tell the system about the events or things they would like to be reminded of, or sending messages to other residents before they leave the house.

The Gate Reminder has a clear specific purpose: reminding. However, I believe that it could be integrated as part of a smart context-aware front door system that does more than just reminding. Smart front doors functions are usually more concerned with activities from *outside* the house while the Gate Reminder is mostly relevant *inside* the house. One possible example scenario where this integration would be useful is when the Gate Reminder detects a resident is about to leave the house without bringing her house keys it would prevent the front door to be automatically locked and it would also attract the resident's attention to remind her that she forgets to bring the house keys.

Reminder applications are especially useful as a memory augmentation tool for elderly individuals who suffer from reduced cognitive and memory capabilities.

- The Massachusetts Institute of Technology (MIT) Media Laboratory researchers created a prototype of a set of context-aware smart sinks (Bonnani, Arroyo, Lee & Selker 2005, Arroyo, Bonnani & Selker 2005).

The *Heat Sink* detects the temperature of the running water and illuminates the water stream with red colour when it is hot, and blue colour when it is cold, or a mix between red and blue when it is warm. The light will blink when the water temperature is dangerously hot.

The *See Sink* can adjust its behaviour based on the kind of object placed in the sink. The sink has a camera mounted to detect the type of object, for example, if it detects human hands it will dispense warm water as long as the hands are in the sink, if it detects vegetable it will dispense cold water, and it will fill a pot with hot water. This proof-of-concept is very rudimentary and this simple rule-based task model might not work well in real life because it is not always the case that a person washing her hands prefers warm water to cold water, for example. And some people may prefer to cook with cold water in the pot instead of hot wa-

ter. There is plenty of room for improvements in the rule-based model to better understand and accommodate many different scenarios and preferences.

The *WaterBot* is a faucet attachment that monitors water usage patterns by sensing the water flow volume and duration, it will provide visual and auditory feedback to the user accordingly to persuade good behaviour in conserving the use of water.

The *Clean Sink* is especially useful in environments where enforcement of hygienic behaviour is especially important, for example, in hospitals and food manufacturing or processing factories. The sink has a RFID reader to identify the person and a camera to monitor that hands are under a stream of warm water for a predetermined amount of time before letting the person enter clean areas.

The *Up+Down Sink* can adjust its height based on the user's height. The sink has a camera that detects a person's head and adjust the height accordingly. After use the sink readjusts itself to a default universal height. This sink is especially useful for tall persons, children and individuals in wheelchairs.

All of these smart sinks, except the *Clean Sink*, are appropriate for use in smart homes that support independent living.

- *SmartWave* (Russo, Sukojo, Helal, Davenport & Mann 2004) is a microwave cooking system that lets the user cook food without having to remember detailed instructions or interact with the microwave user interface, which can be confusing for people with reduced memory and cognitive abilities. Hence the *SmartWave* is especially useful for elderly and other individuals who may suffer from reduced memory and cognitive abilities.

The core of the *SmartWave* system consists of a slightly modified microwave to allow it to be controlled from a computer, and an RFID reader to read RFID tags from the food packages or containers so that the system can lookup from a database about the cooking instructions and other important information about that food based on the RFID tag. The *SmartWave* system is part of the *SmartKitchen* system, which is itself part of a smart house. As such, the *SmartWave* system relies on some

of the smart house infrastructure to provide its services, for example, the SmartWave system makes no assumption about the location of the user, it can simply use the smart house's notification system to tell the individual when the food being cooked in the SmartWave is ready.

SmartWave has some additional features, such as, keeping a history of what has been cooked, voice recognition to capture feedback from the user so that the next time the cooking could be adjusted, monitoring a user's diet and restrictions,

The SmartWave is quite comparable to the *Déjà Vu Displays* (Tran & Mynatt 2003) system, but there is a fundamental difference: the SmartWave system is an *active* memory aid tool where it notifies its user what to do next, while the *Déjà Vu* display is a *passive* memory aid tool where the user needs to view the collage recording the steps of previous cooking.

- *Smart Medical Home*⁴² from Center for Future Health, University of Rochester. The project is focused on the medical and health applications within the smart home. The prototype uses various sensors including infrared sensors, biosensors and video cameras in different rooms. It also employs techniques to measure various vital signs.
- *House_n*⁴³ from MIT. PlaceLab (not to be confused with *Place Lab*, the location based framework from Intel) is House_n's major project of a shared apartment research facility for researchers to develop and test new technologies in the home that improve the quality of life of the residents. PlaceLab is equipped with various sensors, audio and video capture, location beacons, biometric devices, home automation and environmental controls. (House_n Research Group, Department of Architecture & Massachusetts Institute of Technology 2003)
- Ubiquitous Home from NICT Japan (Yamazaki 2005) is a testbed for home context-aware service experiments which the author, Yamazaki, claims *superior* to other similar testbeds. It is equipped with many microphones and cameras in practically every space of the home, however, the author acknowledges the need for privacy consideration of installing

⁴²http://www.futurehealth.rochester.edu/smart_home/

⁴³http://architecture.mit.edu/house_n/

so many microphones and cameras but does not elaborate further on its implications. The home has some Japanese specific services in place, for example, it is connected to a remote Japanese style room to test remote family connectivity. The home also has robots that performs various services in the home. Other sensors that are installed in the home include pressure sensitive floor to track residents or furniture positions, infrared sensors in the ceilings to detect human movements, RFID systems to detect presence of people or items with RFID tags, accelerometer or vibration sensors which are more sensitive than the pressure floor.

- There are various other applications listed at http://www.cc.gatech.edu/fce/seminar/fa98-info/smart_homes.html

4.5 Issues Specific to Context-Awareness in Supporting Independent Living

One of the major issues with context-aware applications that support independent living is *User Interface* design. It is more significant to this class of context-aware applications because the users may be less adept to using complicated devices. Minimum interactions is generally desirable, as well as intuitiveness and overall simplicity of the design. When some interactions are unavoidable and necessary for gathering input from the user, a physical user interface, such as the iStuff toolkit (Ballagas, Ringel, Stone & Borchers 2003), may be more appropriate in certain cases, for example for people with disabilities.

There are other non-technical issues such as *cultural differences*. For example, in some culture the people may be reluctant to having robots in the home, but others such as in Japan (Yamazaki 2005) may be more accepting. The *Gate Reminder* (Kim, Kim, Park, Jin & Choi 2004) relies on the Korean home tradition that the residents take off their shoes near the entrance of the home, and they will stop to wear their shoes near the entrance before going out.

As explained in 2.3.1 on page 20, ubiquitous computing applications generally have a relatively high cost associated with them. This means that the final price that the consumers (e.g. the elderly) have to pay will be relatively high, too. In turn, this may or may not affect the healthcare system. If the average total costs for the smart home applications are higher than the average costs

of the expenses replaced by the smart home applications, then there may be an increase in the overall healthcare costs. However, I also speculate that the positive effects of supporting independent living on the individuals' health might mean that there will be less healthcare expenses. Coory (2004) showed that increased ageing population does not necessarily mean increased healthcare costs. When smart homes become more pervasive, then it also becomes more relevant to bring smart homes and the effects of supporting independent living into the discussions about healthcare costs projection.

Different levels in the Quality of Context (QoC) could lead to inaccuracies of the deductions made by context-aware applications about the current context. This could, in turn, lead to wrong behaviour being performed by the applications relative to the actual context. Wrong system behaviour combined with trust and reliance on the system could have potentially fatal consequences in medical and healthcare applications. For example, if a medication reminder application incorrectly deducted that a patient has not taken her medication when actually she has just did, then it would notify the patient to take her medication. If the patient trusts the system 100%, or if she has memory problems, then she could be taking her medication again. (Agarawala, Greenberg & Ho 2004)

Context-aware applications can acquire context from various sources to improve the overall quality of context. This technique also has the benefit of improving the overall availability of the system in case one of the sensors stops working. On the topic of availability, some critical context-aware applications, for example a home emergency medical attention application that calls for outside help when the resident is in emergency situation, should have some kind of a backup system just in case the main system experience a failure.

5 Research Proposal

A wearable wireless heart monitor device from Alive Technologies called Alive Monitor⁴⁴ has recently become available to the Faculty of Information Technology at the University of Technology, Sydney. The device can output ECG and 2-axis accelerometer data via the Bluetooth Serial Port Profile (SPP).

⁴⁴<http://alivetec.com/products.htm>

The objective of my research is to discover the capabilities of the Alive Monitor device and exploit them in the best possible way to provide a useful context-aware application. More specifically, my proposal is to design and develop a context-aware application that can monitor elderly residents with heart or cardiovascular problems, and actively responds if it detects any abnormalities. The system monitors various heart related variables that are available from the Alive Monitor device. It analyses the data in real time for reporting purposes as well as for detecting any anomalies in the patient's condition that should receive immediate attention from qualified physicians or family members. The system should enable elderly individuals with cardiovascular conditions to live independently in the comfort of their own homes, and allow them to do their daily activities normally, while still allowing their care givers and extended family members to monitor or occasionally check up on the elderly's health condition.

The suggested approach for the research is an approach similar to an incremental top-down design approach. The researcher should start by obtaining some background knowledge in cardiovascular diseases, electrocardiography, accelerometer, data acquisition techniques, pattern recognition, similar previous work done by other people, open issues, and so on. This step is important so that the researcher would get a better overall picture for the research and could discover some ideas about possible ways to tackle the open issues. The researcher could then start by focusing on each of the functionality offered by the Alive Monitor device. It seems that the accelerometer function has the potential to be used to detect falls. Therefore, the researcher could start by looking at the accelerometer capabilities and how to identify patterns from the output that represents falls. The researcher then would do the same with the ECG capabilities of the Alive Monitor device to detect abnormalities in the ECG output that would indicate various heart-related problems. The next step would be to integrate all the functionalities in a bigger application. The researcher would design the application, decide which framework to use (if applicable), and then implement the design by developing a prototype of the application.

6 Conclusions

In order to provide introductory background information on the topic of context-awareness in smart homes to support independent living, this paper have discussed the notions of ubiquitous computing, context-aware computing, smart homes, and supporting independent living, the relation and relevance to each other, as well as presented some example applications, current research and issues for each of the subtopic.

To summarise, some of the primary issues to be considered are ease of use, intuitiveness, and non-intrusiveness of the technology. Security, privacy and safety issues are equally important and should be studied in more detail, especially for critical applications that may have fatal implications if improperly used or managed. Many earlier research and studies have not been paying much attention to the use of advanced AI techniques for *understanding context*, and *understanding user desires*, which are the essence of *intelligence* in context-aware applications. The cost and availability of some more esoteric sensors is also another major issue in the lack of wide adoption of context-awareness.

This paper have discovered that some research have started to use common infrastructure to develop their context-aware applications, for example the Gator Tech Smart House (Helal et al. 2005) and CyberGuide (Choi, Shin & Shin 2005) both use the OSGi framework. Investigating whether some of these frameworks are mature enough and could be used to quickly and efficiently develop context-aware applications in smart homes is another potential research topic.

As a general conclusion based on the literatures surveyed in this paper, I would say that the various testbeds and prototypes of smart homes from the research groups have resulted in many interesting and promising context-aware applications in the home. Some of the context-aware smart home applications that have been presented in this paper can be highly beneficial, not only to the general population, but also to supporting independent living of elderly individuals and people with disabilities. However, there does not seem to be any *killer context-aware application* yet. Understanding contexts and user intent is the key to intelligence in successful context-aware applications, and there does not seem to be any context-aware applications that can successfully understand both contexts and user intent.

7 Acknowledgements

I would like to thank Dr. Peter Leijdekkers, Dr. Valérie Gay, and Dr. Jenny Edwards for their guidance, patience, support and invaluable inputs in writing this paper.

References

- Abascal, Julio. 2005. "Ambient Intelligence for people with disabilities and elderly people." <http://www.andrew.cmu.edu/course/60-427/aisd/elderly.pdf>. 28 September 2005
- Abowd, Gregory D., C. G. Atkeson, J. Hong, S. Long, R. Kooper & M. Pinkerton. 1997. "Cyber-guide: A Mobile Context-Aware Tour Guide." *ACM Wireless Networks* 3 . 1997
- Agarawala, Anand, Saul Greenberg & Geoffrey Ho. 2004. "The Context-Aware Pill Bottle and Medication Monitor." *UBICOMP 2004, Nottingham, England* . 2004
- Arroyo, Ernesto, Leonardo Bonnani & Ted Selker. 2005. "Waterbot: Exploring Feedback and Persuasive Techniques at the Sink." *CHI 2005, Portland, Oregon, USA* . April 2005
- Asthana, Abhaya, Mark Cravatts & Paul Krzyzanowski. 1994. "An indoor wireless system for personalized shopping assistance." *IEEE Workshop on Mobile Computing Systems and Applications*, pp. 69-74 . December 1994
- Bahl, Paramvir & Venkata N. Padmanabhan. 2000. "RADAR: An In-Building RF-based User Location and Tracking System." *INFOCOM* . 2000
- Ballagas, Rafael, Meredith Ringel, Maureen Stone & Jan Borchers. 2003. "iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments." *Human Factors in Computing Systems, CHI Letters 2003* . April 2003
- Bardram, Jakob E. 2005. "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications." *Third International Conference on Pervasive Computing (Pervasive 2005)*, vol. 3468 of *Lecture Notes in Computer Science*, pp. 98-115, Munich, Germany . May 2005
- Becker, Christian & Frank Dür. 2005. "On location models for ubiquitous computing." *Personal and Ubiquitous Computing*, 9: 20-31 . 2005
- Bonnani, Leonardo, Ernesto Arroyo, Chia-Hsun Lee & Ted Selker. 2005. "Smart Sinks: Real-World Opportunities for Context-Aware Interaction." *CHI 2005, Portland, Oregon, USA* . April 2005
- Chen, Guanling & David Kotz. 2000. "A Survey of Context-Aware Mobile Computing Research." *Dartmouth Computer Science Technical Report TR2000-381* . 2000
- Cheverst, K., K. Mitchell & N. Davies. 1998. "Design of an Object Model for a Context Sensitive Tourist GUIDE." *International Workshop on Interactive Applications of Mobile Computing (IMC98)* . November 1998
- Choi, Jonghwa, Dongkyoo Shin & Dongil Shin. 2005. "Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances." *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 301-306 . February 2005
- Coory, Michael D. 2004. "Ageing and healthcare costs in Australia: a case of policy-based evidence?" *The Medical Journal of Australia*, Vol. 180, pp. 581-583 . 7 June 2004
- Dey, Anind K. & Gregory D. Abowd. 2000. "Towards a Better Understanding of Context and Context-Awareness." *CHI 2000 Workshop on the 'what, who, where, when and how' of Context-Awareness, Netherlands* . April 2000

- Dey, Anind K., Raffay Hamid, Chris Beckmann, Ian Li & Daniel Hsu. 2004. "a CAPpella: Programming by Demonstration of Context-Aware Applications." *CHI Letters 2004* . 2004
- El-Zabadani, Hicham, Sumi Helal, Bessam Abdulrazak & Erwin Jansen. 2005. "Self-Sensing Spaces: Smart Plugs for Smart Environments." *3rd International Conference On Smart homes and health Telematic (ICOST), Sherbrooke, Québec, Canada* . July 2005
- Erickson, Thomas. 2002. "Some Problems with the Notion of Context-Aware Computing." *Communications of the ACM, Vol. 45, No. 2, pp. 102-104* . February 2002
- Fershca, Alois, Wolfgang Beer & Wolfgang Narzt. 2001. "Location Awareness in Community Wireless LANs." *Informatik 2001, Vienna, Austria* . September 2001
- Fritzke, Bernd & T. Leen. 1994. "A Growing Neural Gas Network Learns Topologies." *NIPS 1994* . 1994
- Fritzke, Bernd. 1994. "Fast learning with incremental RBF networks." *Neural Processing Letters* . 1994
- Gopalratnam, K. & D. J. Cook. 2003. "Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction." *Florida Artificial Intelligence Research Symposium 2003* . 2003
- Harter, Andy, Andy Hopper, Pete Steggles, Andy Ward & Paul Webster. 2001. "The Anatomy of a Context-Aware Application." *Wireless Networks 1 (2001) 1-16* . 2001
- Haya, Pablo A., Germán Montoro & Xavier Alamán. 2004. "A Prototype of a Context-Based Architecture for Intelligent Home Environments." *CoopIS/DOA/ODBASE 2004, LNCS 3290, pp. 477-491* . 2004
- Healey, Jennifer & Rosalind W. Picard. 1998. "StartleCam: A Cybernetic Wearable Camera." *International Symposium on Wearable Computing, Pittsburg, PA, USA* . October 1998
- Helal, Sumi, Carlos Giraldo, Youssef Kaddoura, Choonhwa Lee, Hicham El-Zabadani & William Mann. 2003. "Smart phone based cognitive assistant." *UbiHealth 2003* . 2003
- Helal, Sumi, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura & Erwin Jansen. 2005. "The Gator Tech Smart House: A Programmable Pervasive Space." *IEEE Computer magazine* . March 2005
- Held, Albert, Sven Buchholz & Alexander Schill. 2002. "Modeling of Context Information for Pervasive Computing Applications." *Sixth World Multiconference on Systemics, Cybernetics and Informatics, SCI2002, Orlando* . July 2002
- Henricksen, Karen & Jadwiga Indulska. 2004. "A Software Engineering Framework for Context-Aware Pervasive Computing." *2nd IEEE Conference on Pervasive Computing and Communications (PerCom), Orlando* . 2004
- Henricksen, Karen, Steven Livingstone & Jadwiga Indulska. 2004. "Towards a hybrid approach to context modelling, reasoning and interoperation." *First International Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004* . 7 September 2004
- House_n Research Group, Department of Architecture & Massachusetts Institute of Technology. 2003. "PlaceLab." http://architecture.mit.edu/house_n/documents/PlaceLab.pdf. 11 November 2003

- Huebscher, Markus C. & Julie A. McCann. 2004. "Adaptive middleware for context-aware applications in smart-homes." *2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing*, pp. 111-116. . 2004
- Intille, Stephen S. 2002. "Designing a Home of the Future." *IEEE Pervasive Computing, Integrated Environments*, pp. 80-86 . 2002
- Kidd, Cory D., Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Brian MacIntyre, Elizabeth Mynatt, Thad E. Starner & Wendy Newstetter. 1999. "The Aware Home: A Living Laboratory for Ubiquitous Computing Research." *Second International Workshop on Cooperative Buildings (CoBuild'99)* . 1999
- Kim, SungWoo, MinChul Kim, SangHyun Park, YoungKyu Jin & WooSik Choi. 2004. "Gate Reminder: A Design Case of a Smart Reminder." *DIS2004, Cambridge, Massachusetts, U.S.A.* . August 2004
- Kim, SungWoo, SangHyun Park, JungBong Lee, YoungKyu Jin, Hyun-mi Park, Amy Chung, SeungEok Choi & WooSik Choi. 2004. "Sensible appliances: applying context-awareness to appliance design." . 2004
- Kwon, Ohbyung, Keedong Yoo & Euiho Suh. 2005. "ubiES: An Intelligent Expert System for Proactive Services Deploying Ubiquitous Computing Technologies." *38th Hawaii International Conference on System Sciences* . 2005
- LaMarca, Anthony, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello & Bill Schilit. 2005. "Place Lab: Device Positioning Using Radio Beacons in the Wild." *Pervasive 2005, Munich, Germany* . 2005
- Lee, Sang-Hak & Tae-Choong Chung. 2004. "System Architecture for Context-Aware Home Application." *Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04)* . 2004
- Leijdekkers, Peter & Valerie Gay. 2005. "Personalized Service and Network Adaptation for Smart Devices." *IEEE Asia Pacific Conference on Communications (APCC 2005), Perth, Australia* . October 2005
- Lieberman, H. & T. Selker. 2000. "Out of context: Computer Systems that adapt to, and learn from, context." *IBM Systems Journal, Vol. 39 No.3/4*, pp. 1-16 . 2000
- Maaß, Henning. 1997. "Location-aware mobile applications based on directory services." *Third Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 23-33 . November 1997
- Marples, Dave & Peter Kriens. 2001. "The Open Services Gateway Initiative: An Introductory Overview." *IEEE Communications Magazine, Vol. 39, No. 12*, pp. 110-114 . December 2001
- Mayrhofer, Rene. 2004. An Architecture for Context Prediction PhD thesis Johannes Kepler University Linz, Austria. 2004
- McFadden, Ted & Jadwiga Indulska. 2004. "Context-Aware Environments for Independent Living." *3rd National Conference of Emerging Researchers in Ageing (ERA 2004), Brisbane, Australia* . December 2004

- McFadden, Ted, Karen Henriksen & Jadwiga Indulska. 2004. "Automating Context-Aware Application Development." *First International Workshop on Advanced Context Modeling, Reasoning and Management, Ubicomp 2004, Nottingham, England* . September 2004
- Merriam-Webster Inc. 2005. "Merriam-Webster Online Dictionary." <http://www.m-w.com>. September 2005
- Meyer, Sven & Andry Rakotonirainy. 2003. "A Survey of Research on Context-Aware Homes." *Conferences in Research and Practice in Information Technology Series, Proceedings of the Australasian information security workshop conference on ACSW frontiers Vol. 21, Adelaide, Australia, pp.159-168* . 2003
- Mihailidis, Alex & Geoffrey R. Fernie. 2002. "Context-aware assistive devices for older adults with dementia." *Gerontechnology, Vol. 2, No. 2* . December 2002
- Minoh, M. & T. Kamae. 2001. "Networked Appliances and Their Peer-to-Peer Architecture AMI-DEN." *IEEE Communications Magazine, pp. 80-84* . October 2001
- Mozer, Michael C. 1998. "The Neural Network House: An Environment That Adapts to Its Inhabitants." *AAAI 1998 Spring Symposium on Intelligent Environments, pp. 110-114* . 1998
- Murthy, V. K. & E. V. Krishnamurthy. 2005. "Contextual Information Management Using Contract-Based Workflow." *2nd Conference on Computing Frontiers, CF'05, Ischia, Italy* . May 2005
- Mynatt, Elizabeth D., James Rowan, Sarah Craighill & Anne Jacobs. 2001. "Digital family portraits: Providing peace of mind for extended family members." *ACM Conference on Human Factors in Computing Systems* . 2001
- Nishigaki, Kouji, Keiichi Yasumoto, Naoki Shibata, Minoru Ito & Teruo Higashino. 2005. "Framework and Rule-based Language for Facilitating Context-aware Computing using Information Appliances." *25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)* . 2005
- O'Connell, Pamela L. 2005. "Korea's High-Tech Utopia, Where Everything Is Observed." <http://www.nytimes.com/2005/10/05/technology/techspecial/05oconnell.html>. 5 October 2005
- Pascoe, Jason. 1998. "Adding generic contextual capabilities to wearable computers." *Second International Symposium on Wearable Computers, Pittsburg, Pennsylvania* . October 1998
- Roos, Teemu, Petri Myllymäki, Henry Tirri, Pauli Misikangas & Juha Sievänen. 2002. "A Probabilistic Approach to WLAN User Location Estimation." *International Journal of Wireless Information Networks, Volume 9, Issue 3, pp. 155-164* . July 2002
- Rosen, Neal, Rizwan Sattar, Robert W. Lindeman, Rahul Simha & Bhagirath Narahari. 2004. "HomeOS: Context-Aware Home Connectivity." *International Conference on Pervasive Computing and Communications (PCC'04), Las Vegas, Nevada, USA, pp. 739-744* . June 2004
- Russo, James, Andiputranto Sukojo, Abdelsalam (Sumi) Helal, Rick Davenport & William C. Mann. 2004. "SmartWave - Intelligent Meal Preparation System to Help Older People Live Independently." *Second International Conference on Smart Homes and Health Telematics, ICOST'2004, vol. 14 of Assistive Technology Research Series, pp.122-135* . 2004

- Salber, Daniel, Anind K. Dey & Gregory D. Abowd. 1999. "The Context Toolkit: Aiding the Development of Context-Enabled Applications." *CHI'99, Pittsburgh, PA, USA* . May 1999
- Sanders, Jane M. 2000. "Sensing the Subtleties of Everyday Life." GT Research Horizons <http://gtresearchnews.gatech.edu/reshor/rh-winoo/main.html>. 2000
- Satyanarayanan, M. 2001. "Pervasive Computing: Vision and Challenges." *IEEE Personal Communications* . 2001
- Schilit, Bill N., Norman Adams & Roy Want. 1994. "Context-Aware Computing Applications." *IEEE Workshop on Mobile Computing Systems and Applications* . December 1994
- Schmidt, A., M. Beigl & HW Gellersen. 1999. "There is more to context than location." *Computers & Graphics* . 1999
- Stajano, Frank. 2002. *Security for Ubiquitous Computing*. Wiley. March 2002
- Starner, Thad, Jake Auxier, Daniel Ashbrook & Maribeth Gandy. 2000. "The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring." *IEEE International Symposium on Wearable Computing (ISWC 2000), Atlanta, GA* . October 2000
- Strang, Thomas & Claudia Linnhoff-Popien. 2004. "A Context Modeling Survey." *First International Workshop on Advanced Context Modelling, Reasoning and Management, Ubicomp 2004* . 7 September 2004
- Swedberg, Göran. 1999. "Ericsson's mobile location solution." *Ericsson Review No. 4*, pp. 214-221 . 1999
- Tran, Quan T. & Elizabeth D. Mynatt. 2003. What Was I Cooking? Towards Déjà Vu Displays of Everyday Memory. Technical Report GIT-GVU-TR-03-33 Georgia Institute of Technology. 2003
- van Setten, M., S. Pokraev & J. Koolwaaij. 2004. "Context-Aware Recommendations in the Mobile Tourist Application COMPASS." *Adaptive Hypermedia 2004* . 2004
- Voelker, Geoffrey M. & Brian N. Bershad. 1994. "Mobisaic: An information system for a mobile wireless computing environment." *IEEE Workshop on Mobile Computing Systems and Applications*, pp. 185-190 . December 1994
- Want, Roy, A. Hopper, V. Falcao & J. Gibbons. 1992. "The Active Badge Location System." *ACM Transactions on Information Systems* 10(1) . 1992
- Want, Roy & Trevor Pering. 2005. "System Challenges for Ubiquitous & Pervasive Computing." *ICSE'05* . May 2005
- Ward, A., A. Jones & A. Hopper. 1997. "A New Location Technique for the Active Office." *IEEE Personal Communications* 4(5) . 1997
- Weiser, Mark. 1991. "The Computer for the 21st Century." *Scientific American* . 1991
- Weiser, Mark. 1993. "Some computer science issues in ubiquitous computing." *Communications of the ACM*, Vol. 36, Issue 7 . July 1993
- Wikipedia. 2005a. "Context awareness." http://en.wikipedia.org/wiki/Context_awareness. 22 October 2005

- Wikipedia. 2005b. "Feature vector." http://en.wikipedia.org/wiki/Feature_vector. 28 October 2005
- Wikipedia. 2005c. "Global Positioning System." <http://en.wikipedia.org/wiki/GPS>. 23 September 2005
- Wikipedia. 2005d. "Radio clock." http://en.wikipedia.org/wiki/Radio_clock. 23 September 2005
- Wikipedia. 2005e. "Ubiquitous computing." http://en.wikipedia.org/wiki/Ubiquitous_computing. 22 October 2005
- Yamazaki, Tatsuya. 2005. "Ubiquitous Home: Real-life Testbed for Home Context-Aware Service." *First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM'05)*. 2005
- Youssef, Moustafa A., Ashok Agrawala & A. Udaya Shankar. 2003. "WLAN Location Determination via Clustering and Probability Distributions." *IEEE PerCom*. 2003
- Youssef, Moustafa & Ashok Agrawala. 2005. "The Horus WLAN Location Determination System." *ACM Mobisys*. 2005
- Zhang, Tao. 2004. "An Architecture for Building Customizable Context-Aware Applications by End-Users." *Pervasive 2004 Doctoral Colloquium, Austria*. April 2004
- Zhu, J. & G. D. Durgin. 2005. "Indoor/outdoor location of cellular handsets based on received signal strength." *ELECTRONICS LETTERS*, Vol. 41, No.1. 6 January 2005