Situatedness and Context Awareness in Mobile Computing

By Yohann Daruwala

A thesis submitted in partial satisfaction
of the degree of Masters of Philosophy

Key Centre of Design Computing and Cognition
Faculty of Architecture, Design Science and Planning
University of Sydney

Supervisor: Dr. Andy Dong
Associate Supervisor: Professor Mary Lou Maher

8th April 2007

© Yohann Daruwala 2007
1. Introduction

When humans interact with other people in the surrounding environment they make use of implicit situational information. People naturally assume and interpret the context of the situation they are in and respond fittingly. Computers are not as good as humans in assuming situational information from their environment and in using it in interactions. Humans cannot easily take advantage of such information in a transparent way, and if they can, they usually require that it is explicitly provided. In a mobile computing environment, it will be nice if a user can obtain services and information according to the current location and activity.

Although still very much in the category of emerging technology, mobile computing devices have become increasingly invaluable to users across a variety of fields. Advances in technology have meant these devices are no longer just electronic replacements for paper diaries; current models can perform a wide range of functions, from document processing to communications. Mobile devices are getting cheaper, more powerful and more ubiquitous. These new technologies can be seen as resources for accomplishing various everyday activities that are carried out on the move. People have tremendous capabilities for utilising mobile devices in various innovative ways for social and cognitive activities. Due to the nature of these trends, the research is motivated by the increasing interest in context aware computing – applications that change their behaviour according to the user’s context.

Since the term context aware was first introduced in by Schilit et al.(1994), it has grown into a technology that is incorporated in a variety of applications as a means to facilitate higher level of interaction between users and applications. The Olivetti Active Badge is commonly agreed as the first research investigation into context aware computing. Dey et al.(1999a) described context as the implicit situational information which humans used when talking with humans. A more detailed definition by them termed context as 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 applications themselves. This includes information which is either implicitly or explicitly indicated by the user. While their definition of context aware is, a system is context aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task (Dey et al., 1999b). According to Schilit et al.(2002), there are three processes in the context aware life cycle:

- Context Discovery – capturing the context information
- Context Selection – interpreting the captured information
- Context Use – utilising the information by the application to respond appropriately.
They also define context aware computing as applications which “adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time. In addition to being able to obtain the context information, it must include some way of processing this information to be meaningful to the user. This is probably the most challenging problem.

In our research we focus on the process of context use in mobile computing and look at the ways context aware information can be used as a basis for constructive memory. Learning will involve making rich connections within these environments to both resources and to other people. In order for the construction of context to be able to be used effectively, it needs to facilitate response to the information which the application might obtain within all possible situations under which the application will be executed. The application designer is required to spend large amounts of time analysing the situations to be able to appropriately determine the information which is relevant and how to respond to it. One major issue here is that handling all possible situations is a non-trivial design time task. A particular type of application which is widely used in context awareness is intelligent personal assistant. To date, there has been large amount of research done in incorporating context awareness into reminder systems, a subtype of intelligent personal assistant, for example Stick-e Notes (Brown et al., 1997), W-Mail (Ueda et al., 2000) and CybreMinder (Dey and Abowd, 2000). The responsibility of a reminder system is to inform users of future activity that the user should engage in by sending a special type of message. By incorporating a constructive memory system into context awareness, a reminder system facilitates the use of rich context for user to specify when to send the reminder and it obtains high level of improvement. However, as with any other context aware application, there is the constant challenge of handling all possible situations to ensure this new form of learning is highly situated and personal.

1.1 Aims and Objectives

The aim of this research is to develop a model of constructive memory for mobile computing for the recollection of context. The main purpose is to assist in the potential improvements to be gained by incorporating a learning component in a context aware reminder system. With learning, we allow user behaviour to be automatically detected by the system. However, to the best of our knowledge, amongst those learning architectures, none supports a specific feature that is perceived as highly essential – the ability to learn in an uncertain learning environment. Additionally the focus is on the ability to determine a pattern of behaviour from the history of interactions/experiences between the user and the system and interpret this into knowledge. We can use the knowledge that is gained as basis to determining a future situation. This development
Chapter 1

involves analysis and deconstruction of existing approaches to context awareness, constructive memory and the issues relevant to mobile computing environments.

In order to achieve the stated aim, the following objectives need to be met:

- Identify the gaps in the context aware lifecycle
- Identify the issues relevant to mobile computing
- Design a constructive memory model for context to observe past user experiences
- Analyse capabilities of the model comparison to existing ones
- Design, develop and implement an application prototype
- Evaluate results by the utilisation of the implemented prototype
- Determine the criteria for measuring the success

1.2 Motivation

Mobile technologies offer experiences which can effectively engage and educate contemporary users and are often markedly different from those afforded by conventional desktop computers. These devices are used dynamically, in many different settings, giving access to a broad range of uses and situational experiences. The personal nature of these technologies means they are well suited in engaging users in individual learning experiences and to giving them increased ownership and responsibility over their work. The nature of learning in an educational context is closely linked to the concept of mobility and there are three ways in which learning can be considered mobile. Firstly learning is mobile in terms of space as is it happens at the classroom, at home and at places of leisure. Secondly it is mobile between different areas of life and may relate to work demands, self improvement or leisure. Lastly learning is mobile with respect to time as it happens at different times during the day, on a working day or on a weekend. The close relation of context and the situation in which knowledge arises is discussed in the literature. People continuously learn through their experiences and utilise knowledge whenever they want to carry out some tasks. Simply, human interaction in a particular situation depends on the knowledge gained through their past experiences. Knowledge is information in context and since mobile devices enable the delivery of context specific information they are well placed to enable the construct memories for the recollection of past experiences.

Although a lot of research has been conducted within context aware systems, the core term context is not yet a well defined concept. Currently most attempts to use context aware systems within ubiquitous computing environments have centred on the physical elements of the environment, the user, or the mobile device. The research will focus on a more precise definition for context in deducing new and relevant information to the use of applications and users from the
various sources of context data. The focus on context in mobile computing concerns the ability of devices to detect, interpret, reason and respond to aspects of the user’s physical environment. This capability promises to add value to existing uses as well as create new types of applications. Mobile devices that intelligently act autonomously in these environments will become increasingly useful in the future.

1.3 Significance

As the human computer interface becomes more pervasive and ubiquitous, it will need to explicitly draw upon cognitive science as a basis for understanding what people are doing. The significance of the research lies in fusing the benefits of context awareness with a constructive memory model for mobile computing to provide a higher level of efficiency. From our current understanding of both types of models, there is an interesting issue in the ways knowledge is characterised and used. From the view of a context aware model, some type of contextual knowledge is required, such as information about the user or the user’s environment. The knowledge use in context aware systems is almost invisible. Nevertheless this is not adequate; some mechanism to reason and learn about this information is also needed. Within a constructive memory system, the very act of constructing a memory of an associated experience effects the memory through its grounding in the interaction with the environment. Any knowledge about the current environment, the internal state of the environment is used as cues in the construction process. The central point is that knowledge is fluid and knowledge evolves continually as it is being used. The problem of filtering the vast amount of contextual information that is available, in such a way that the identification of important groups of the contextual information is practical, has not been thoroughly addressed. In combining context awareness and memory, we can concern ourselves with constructing memories for the recall of context-as-data and adapting to new contexts by remembering similar previous experiences.
2. Literature Review

2.1 Context Awareness

In a general sense context can be defined as relevant, interrelated conditions such as environments or situations, in which something exists or occurs. While this is a general definition of context, it does not help in understanding the idea in a computing environment. Context aware computing (CAC) is a field of study researchers have just begun to explore in respect to mobile computing. In respect to mobile computing environments, context awareness will include the characteristics of the surrounding environment determining the behaviour of the mobile application and the relevancy to the application. Mobile devices are especially well-suited to context awareness simply because they are available in different contexts and so can draw on those contexts to enhance the event or activity.

The particular work that propelled context aware computing into popularity was conducted by Schilit et al. (1994). The article defines context aware computing as software that adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time. According to Abowd et al. (1999), this is due to the richness of the language that humans share, the common understanding of how the world works, and an implicit understanding of everyday situations, which is enabled by usage of implicit situational information, that is context. Unfortunately, when humans interact with computers, the ability to convey ideas does not transfer well. To improve level of communication in human-computer interaction, there is a need to enhance computer's access to context. Dey and Abowd (1999a) give their own definition, which 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 characterise 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.

A categorisation is done to assist application developers in determining context types, such as which member of context is appropriate for their application, arranging the context used, and finding other relevant contexts. In practice, there are several members of context which are significant – location, identity, activity, and time. These are the primary context types used to characterise a state of a specific entity, whereas some other members of context are classified as secondary context types. Context aware applications are systems which utilise context to provide appropriate information and/or services to users depending on their tasks. The utilisation of context awareness in application is explained as steps within context aware life cycle. The broad
structure of context aware life cycle is described as consisting of these three particular processes; Context Discovery, Context Interpretation, Context Use (Schilit et al., 2002). Dey and Abowd (1999a) further categorises context based on what kind of information application developers will consider the most when designing their applications. Context aware applications look at who’s, where’s, when’s, and what’s of entities and use this information to determine why the situation is occurring. As previously enumerated examples in the context classification, there are many types of information other than location that can be used to provide context. However, location is probably the most common context. A simple example would be music that is played only in the room where a user is, or lights that turn off themselves automatically when the last person leaves the room (this combines location with presence as context). In fact this is an old technology which is seen in action everyday when the light turns on when the refrigerator is opened and turns off when closed. However when talking about context awareness in ubiquitous computing, it refers to the computational aspects and not the mechanical type context awareness used in refrigerator lighting.

2.1.1 Context Discovery

The idea behind context aware computing is the user’s environment – including where they are, who they are with and what they are doing- can inform the mobile computing device. In the step of context discovery, the process of capturing the context information which is available is done. As context awareness is quite a recently developed technique, much ongoing research is concerned with discovering the most effective approach to capture the context information. The results of the survey in the area of context discovery shown that context capturing can be either done by having badges which transmit a unique identifier to a network of ceiling based sensors using infrared (Want et al., 1992), or having a device which is carried around and gathered information about its location and orientation from the ceiling based sensors (Abowd et al., 1997b), or even by capturing context information about user’s state (Dey et al., 1998).

The Active Badge system from the Olivetti Research Lab is generally considered to be one of the first context aware applications (Want et al., 1992). The system was designed so that people could be located in an office and calls forwarded to the closest phone. The office personnel wore badges that emitted a unique infra-red signal every 10 seconds, which is detected by sensors placed around the office building. A central location server polled these sensors and the receptionist was able to locate the person and direct the call to the closest appropriate telephone. The system also included commands to obtain the current location of a badge, to find out which other badges were in immediate proximity to a named badge, to find out which badges were currently near a specified location. A parallel to draw from this system will be to only disseminate
the information that is required to a specific location for the user. Additionally by reversing the method developed by the research lab, the user can be informed by the system that they in close proximity to others with similar contextual information and preferences.

Despite its high similarity with the Active Badges system, Cyberguide (Abowd et al., 1997b) distinguish its approach by focusing in the domain of tourism. Started as an indoor mobile tour guide service for Graphics, Visualisation and Usability (GVU) Center’s visitors at Georgia Tech, it extends its ability to provide outdoor mobile tour guide service as well. However, after a revision of the original implementation, the use of static configuration causes a detrimental effect on its evolution and extensibility.

CyberDesk (Dey et al., 1998) differs itself by having user’s state as its focus instead of location. It was built as an architecture to automatically offers web-based services based on context information gathered from user activity. Its focused primary context are the data and applications that user is utilising and user’s location as well as time of the day. CyberDesk supports limited types of context and it is unable to support the case where multiple applications are running simultaneously. Another significant shortfall is that it neither store nor able to query the context used. Nevertheless, it has made quite contribution in the area of context awareness.

The ParcTab is another system developed at the beginning of the 1990's at the Xerox Palo Alto Research Centre (Chen and Kotz, 2000). The ParcTab acts as a mobile personal digital office assistant. The system is based on palm-sized wireless ParcTab computers and an infrared communication system (Figure 1) that links them to each other and to desktop computers through a local area network. The system developed did experiments with ubiquitous computing and context awareness in an office environment. The ParcTab system can continuously know where each ParcTab is – which were in the same room and notify applications of location changes. The new model will build upon the features of the ParcTab including information about the room the user is in but additionally information on other rooms in the building. Improvements can be made on ParcTab’s use of context by identifying the presence of other mobile devices to communicate with it users and other non-mobile objects in the user’s environment.
Brown et al. (1997) argues that context aware applications can be divided into continuous and discrete categories. In a continuous application, the information on the mobile device constantly updates and be likely more “challenging to implement” (Brown et al., 1997, p.60). In contrast discrete applications break down the information into separate contexts, disseminating the information whenever the context is entered or changes. Discrete context is more beneficial and efficient as the user can be truly context aware as and when their context does change.

Salber et al. (1999) have even used these identified shortfalls as the basis for their work on the Context Toolkit. Both Active Badges (Want et al., 1992) and Cyberguide (Abowd et al., 1997b) are quite similar in a way that they both focused in a specific type of context, namely location. Another similarity which is not so obvious is the fact that the system’s architecture is decomposed into smaller components. Whereas Active Badge system consist of four layers – network control, representation, data processing and display interface – the Cyberguide decomposed itself into four different components based on the role each of the component is required to fulfill. In the article by Dey and Abowd (1997), it is stated that the greater the knowledge of the user’s context, the better CyberDesk is able to support the user in their task. This statement clearly indicates how important the context capturing in context aware lifecycle. The main issue that all of these context capture system need to consider is the amount of the information it should obtain. While it seems like the more information is captured, the more helpful the system is, in some situations this is not the case. Further, there is a consideration of the fact that the activity of gathering information might raise social fears regarding personal freedom and individual privacy rights. Hence the conclusion that context information is useful only if it can be interpreted or transformed usefully as well as treated with sensitivity.

2.1.2 Context Interpretation

Context awareness and mobility are core concepts in the emergence of truly ubiquitous computing, enabled by the availability of mobile devices that support context information. The main components of context aware computing and applications is sensors for capturing the context data, a set of rules governing behaviour according to context and a set of actuators for generating responses. Context interpretation involves integration of various types and sources of gathered information into one useful entity, and appears to be the most complicated process. As with the context capture, there have been quite a few notified attempts to do the context interpretation process effectively. However, even up until now, researchers are still working on this area as this is a complicated process, yet of high significance in the context aware life cycle.
Context interpretation process is responsible to select and transform the contextual information gathered in context capture process into a piece of information which is of high usability.

The AROMA (Pedersen and Sokoler, 1997) project strives to assist people who is located geographically dispersed in keeping in touch with other people by providing peripheral awareness. The peripheral awareness is the kind of awareness that effortlessly maintained by people when they are physically located nearby, but grows into a problem when dealing with people who live far away, for example, to allow a person to view caller’s site condition before making a phone call to the callee. Audio Aura (Mynatt et al., 1997) aims to provide serendipitous information which is related to people’s physical actions in the work place through background auditory cues as well as to explore audio utilisation to relate a person’s activities in the physical world with information collected from the virtual world. Audio Aura’s server also maintains a history by keeping the environment captured context. The Context Toolkit (Salber et al., 1999) aims to assist in the development and deployment of context aware applications. It does the interpretation of context while hiding the complex sensing and accessing context information process. Hence by utilising the Context Toolkit, it is expected that developers can concentrate on utilising context information to build applications. One commonality between all these examples are their focus on the interpretation of context. All of them share the architecture with a layer of sensors and a layer of interpreters and a layer of applications.

Biegel and Cahill (2004) have designed the sentient object model (Figure 2) for context aware applications in mobile environments. The appeal of this model is its organised approach to developing context aware applications supporting the important aspects of sensor fusion, context extraction and context reasoning. In terms of developing the computational model, it is more concerned with the efficient approach to intelligent reasoning based on a hierarchy of contexts. The sentient object model is an encapsulated entity with its interfaces being sensors and actuators. The sensors produce software events in reaction to real world stimulus whilst actuators consume software events and react by attempting to change the state of the real world via a mobile device (Biegel and Cahill, 2004).
Figure 2. The sentient object model where actuation is controlled based on sensor input (Biegel and Cahill, 2004, p.362).

The sentient object model defines context as any information sensed from the environment which maybe used to describe the situation. The sensory capture and context fusion component manages the uncertainty of sense data being consumed. Biegel and Cahill (2004), based upon Bayesian networks, employ a probabilistic sensor fusion mechanism for measuring the effectiveness of derivations of context from noisy sensor data. The context hierarchy summarises the information about the activities to be taken and possible future situations in contexts. The advantage of having a hierarchy in the above model (Figure 2) and by associating the activities to commence in each context, "a sentient object's behaviours is influence by context" (Biegel and Cahill, 2004, p.363). The inference engine component is responsible for changing the applications behaviour according to the present context. Additionally the inference engine has a built-in language that when given a set of facts, and predefined rules, is able to decide which rule to fire giving the model intelligence and the ability to react to changing contexts.

Treating context as hierarchy helps to manage the complexity of development by reducing the problem to developing a set of contexts. The context hierarchy increases the efficiency of the inference engine and by introducing the notion of an active discrete context (Brown et al., 1997) it can potentially have a hierarchy where only one context is active at any point in time. Moreover within this context, only a subset of the overall rules governing the behaviour is active. If it can achieve this desired result from the hypothesis that there are only a limited number of event inputs which can take place in any context, the efficiency of the model increases.

The specifications required for handling context in an environment must be able to detect its current state and determine what actions to take based on the context. Dey et al. (1999a) present
an infrastructure solution to help in modelling context and its relation to building applications. The context infrastructure model consists of sensors, widgets, interpreters, a server and Figure 3 shows the relationship between the context components and the application. The context widgets capture a single piece of context. A context server is very similar to a widget in that it supports the same set of features as a widget. The server is responsible for the aggregation of context and for subscribing to every widget of interest, acting as a proxy to the application. The aggregation facilitates the access of context by applications that are interested in multiple pieces of context about a single entity. Context must often be interpreted before it can be used by an application. The interpreters role in model is to abstract the pieces of context that are not sensed immediately from the widgets. The advantage of having this layer of abstraction is context must often be interpreted before it can be used by an application. In previous cases the interpretation of context has usually been performed by applications but by separating the interpretation abstraction from the application, we allow for the reuse of interpreters by other applications. Additionally the model supports interpreters that can be called by widgets, servers, applications and even by other interpreters. When viewing the model as whole and then by breaking down the context infrastructure model, the conclusions drawn are the following: the division of sense data – how to sense and access context information; the aggregation of context in terms of high level information rather than low level details; and finally the abstraction from which we use the context data required for the application.

Figure 3. The relationship between the application and the context infrastructure model (Dey et al., 1999b, p.25)
2.1.3 Context Use

The idea behind context aware computing is the user’s environment - including where they are, who they are with and what they are doing - can inform the mobile computing device. Compare to the other processes in context aware life cycle, this is the one that most related to the users. The knowledge of context aims to be utilised to automate tedious tasks with a certain level of intelligent. Hence, users only need to put a minimum amount of effort into the tedious tasks and concentrate in their other important tasks. In addition, devices that are aware of a user context have higher levels of usability simply because these devices provide interface that are able to adapt to the user instead of forcing the user to adapt to itself. This additional knowledge results in a changed interaction between the user and the mobile computing device. By improving the mobile device’s access to context, it increases the richness of communication in human computer interaction and makes it possible to increase the number of computational services to the device. Dey et al. (1999b) acknowledges context rapidly changes in situations where the user is mobile and has to adapt to provide “relevant services and information to the user” (Dey et al., 1999b, p.21).

The following are some examples applications which utilise context awareness to provide itself with a certain level of intelligence. The Responsive Office (Elrod et al., 1993) is one of the early pieces of research which utilises context information gained using active badge attempting to develop intelligent building management system. During the development, as with any other ubiquitous computing systems, a number of design issues were encountered, mainly reliability and invisibility, privacy and access control. Despite all that, this system has certainly contributed its fair share of knowledge to the world of context utilisation. An attempt to augment both teaching and learning process in a lecture-room environment defines the Classroom 2000 project (Abowd et al., 1997a). It aims to be to automatically capture different streams of classroom activities on an electronic surface and integrate these materials together, which is then made accessible through the Web by means of providing imitation of actual classroom experience. The most obvious context used in this particular project is various events’ occurring over time. To enhance the effectiveness of context use in this case, the materials should be related by topic, not time. Development of the GUIDE project (Cheverst et al., 2000) is a result of investigation in provisioning city visitors with context aware mobile multimedia computing support as their intelligent visitor guides. In GUIDE system, context awareness is built in its end systems while its cell base-stations broadcast all of the required information. This is as opposed to Cyberguide where its end-systems carry pre-installed required information. The Responsive Office utilise users actions as their contextual information in providing intelligent automated service. Where as Classroom 2000 utilises time as a synchronizing quantity for its integrated materials. The GUIDE
system, however, based its context intelligence on changes in its environment, mainly uses movement around the city.

The Conference Assistant, a prototype application developed by Dey et al. (1999b) at Georgia Institute of Technology, is used to aid people attending conferences and help them with concurrent activities that occur whilst in present at a conference. The prototype operates on RFID technology to sense identity and location. The Conference Assistant uses location information, time, activities in a certain location, and user preferences as context information (Chen and Kotz, 2000). The Conference Assistant is an excellent prototype to base parts of the integrated model as it provides awareness of activities to the users of the system. It promotes interaction between users and their surrounding environment. The prototype demonstrates a wide variety of context services such as contextual sensing and adaptation to provide information to its users in real time. These applications can help users navigate unfamiliar territory and have access to information in the most useful and unobtrusive way. The use of context information in applications decreases the human-centred attention an application requests to service the user’s needs. CAC offers an opportunity to substantially increase human productivity. The Owl application developed and experimented with at the IBM T.J. Watson Research Centre (Ebling et al., 2001) accumulates and retains context information from numerous sources. Users can query for the current context information or submit a query to be notified of when their query conditions are satisfied and the available information. There is a belief that knowledge of the past contextual information may allow understanding into present or future behaviour in the absence of current information. Therefore reducing the time needed to produce this information to the user based upon their request. Additionally it may allow us to pre-fetch information in anticipation of future contexts providing better performance and availability.

Brown proposed a framework called Stick-e Documents (1995) for creating context aware applications and it highlighted the importance of contextual awareness in mobile computing. The stick-e document framework is composed of a set of stick-e notes, each of them resembling a page on HTML. Each stick-e note consists of the contents and relevant contextual information to be triggered. When a user carries mobile device equipped with location sensing software, they can place a stick-e note at a physical point of interest. Once the user returns to that location in the future, the stick-e note will be triggered and can inform the mobile device that the associated note exists. A stick-e note can be considered as an electronic form of post-it notes which people place in their working places not to forget or inform different things to be done. Some other examples of context that can trigger a stick-e note is the adjacency of a person to physical objects and the temperature above/below a certain level. The stick-e note approach offers a useful general mechanism for the creation of context aware applications (Brown, 1995).
From these examples we can see how much contribution context awareness has provided in advancing users’ adaptation to ubiquitous computing. It has also provides us with basic capability to break the paradigm of desktop computing. However, current common main issue that arises across them were that context awareness is yet to be part of ubiquitous computing. It still needs to deal with minimize the obstruction for the user to behave normally in his/her activity and still gets the benefits of context awareness. A higher level of ubiquity in context awareness would be to do its life cycle processes without the user realising it.

2.1.4 Strengths and Limitations of Context Awareness

A pervasive computing system like context aware applications must track user’s intention and determine which actions will help and not hinder user. Certainly, the emergence of context awareness effective utilisation has proven to provide applications with the ability to interact richly with their surrounding environment. However, while quite a handful of context aware applications have been developed as a result of it being a well-known research topic, its deployment is yet to be ubiquitous. This is due to the lack of currently available programming support to develop applications with context awareness. Another issue with current state of context awareness is that it is still mainly focusing on location as its primary context type. The development of context aware systems is a complicated process. To develop systems with context awareness requires high programming skill. To develop systems with effective use of context awareness, it requires high application design in addition to programming skills. To develop an effective context aware application which is also user friendly, is even more difficult, because on top of the previously required skills, it also requires someone with an advance graphical user interface skills. Even then, the result is still only a user friendly and effective context aware application. What is missing from this application is the personalisation required to be able to track user’s intentions. Human beings are very complex and diverse. Different people have different intentions. This makes it hard for developers to develop the ideal context aware applications that would suit everyone. Another development issue comes from the fact that most of the context aware applications that exist today work on the basis of conditional rule statements. It is unfeasible to create enough rules statements at design time to satisfy the ever-changing user’s intentions and to comply with real life’s complexity.

2.2 Constructive Memory

There is the obvious observation that we can recall things. We can recall names, numbers, sentences from a poem, laws of physics and so. Although empirical evidence clearly shows recall
is much bettering contexts similar to the context where the information was acquired, recall of the
de-contextualised information is possible and even desirable in problem solving. According to
Clancey (Clancey, 1995), we can only know what we perceive. Constructing a representation
means seeing something in a new light. For example, each act of speaking is a complete act of
perceiving in itself. By speaking we create new meanings which are perceivable by ourselves and
other and thus open to reinterpretation. Memory is what is retained from a previous activity and is
capable for replaying previously enacted sequences of behaviour (“phrases”). New phrases can
be constructed through substitution of actions and recombination of phrases (Clancey, 1995).

Recall of memory is based on matching between specific features and indices contained within
the context/environment. An index (or set of indices) used to retrieve a context event is not
modified as it used during the retrieval process. A constructed memory may or may not contain
the same information used initially to cue the system. What gets constructed is not a function to
cue the system but depends what the system has learnt to be associated with proposing
alternative experiences as a basis for interpreting environment. Situatedness contains two
fundamental ideas of interaction and construction (Gero, 1999). Interaction implies that what the
systems knows is not encoded priori and indexed for use later, but rather knowledge is developed
through interactions with the environment. This interaction interprets the experiences processed
by the system according to the current situation in such a way that the interpretations of past
experiences are filtered defined by the present situation. Compared to non-situated systems they
lack the interaction. The construction of memory is about previous experiences providing the
basis for interpreting the past according to the current situation. A past experience is not copied
into the present but rather interpreted based on the current interactions with the environment.
Once interpreted it eventually becomes part of the memory system – as a new experience – that
influences subsequent interpretations.

A memory is not piece of information that resides in some stored form, indexed and reproduced in
the same way each time it is called upon. It is constructed anew each time the need for it arises.
Memories are initially constructed from an experience in response to demands for memory of that
experience but the construction of memory includes the situations pertaining at the time of
demand for memory. Thus the construction of memory in response to a condition in the external
or internal environment is a situated act (Liew and Gero, 2003).

Constructive memory as a theoretical concept is more concretely known than situatedness. In
one line it refers to the property or characteristic of memory in which the memory of the past
experiences effects, guides and cues the construction of the current experience, and the current
experience in turn changes the memory. In computational terms the current models of memory
are indexing based where exactly the same thing of the past is pulled up as memory – that is memory remains static and changes only as additions or subtractions occur. However in human memory, memory is constructive, that is past experiences affect the current ones and the current ones go and change the memory of past experiences in a circular way which leads to more learning and richer behaviour. This is the basics core concept

Memory is an imaginative reconstruction or construction built out of the relation of our attitude towards a whole active mass of organised past reactions or experience, and to a little outstanding detail which commonly appears in an image or language form. It is thus hardly ever really exact, even in the most basic form of memorisation by repetition, and is not all important that it should be so. Hence remembering is a constructive act. Memory is a label for a diverse set of cognitive capacities by which humans retain information and reconstruct past experiences, usually for present purposes. Humans remember experiences and events which are not happening now, so memory seems to differ from perception. Additionally they remember events which really happened, so memory is unlike pure imagination. Memory seems to be a source of knowledge, or perhaps just is retained knowledge. Remembering is often suffused with emotion and it is an essential part of much reasoning. Some memories are shaped by language, others by imagery. Liew and Maher (2004) use knowledge and experience to provide the basis for memory construction according to the environment and the current situation. One of the advantages of situated case-based reasoning is “the way its knowledge and experience are understood and used” (Liew and Maher, 2004, p.200). Knowledge and experience is built upon the understanding of the environment and relies on the selected pieces of information based on the situation. Knowledge can be defined as the general pieces of information that are known and recognised through experience. Experience therefore refers to previous events or conditions encountered. These experiences can be in the form of information collected over time under controlled conditions and requires the systems direct involvement with that particular situation.
Chapter 2 – Literature Review

Figure 4. A generic case-based reasoning model (left) and a situated case-based reasoning model (right) (Liew and Maher, 2004, p.200).

A constructed memory defines both the interpretation of the relevant content and the interpretation of the environment despite the interactions taking place between the system and the environment. The model of constructive memory (Figure 4) exists within the structure of situated computing (Gero, 2003). The important characteristic of constructive memory is the ability to incorporate the previous experiences with the construction of new and different memories. New memories result from the recursive interpretations of the context and environment which the system is framed – leading to added memory and experience. After each new memory is constructed, it is available for subsequent memory constructions.
Memory construction occurs whenever the system uses a past experience in the current environment in a situated manner (Gero, 1999). Any information available in the current environment is used as cues in the construction process. Associated memories are constructed and grounded according to the current interactions between the system and the environment. This behaviour of a constructive memory system within a situated environment has the following characteristics:

- Memory operates as a dynamic process, it is not a static imprint to be stored in a specific location and retrieved for use later.
- The operations of the memory system are not predefined, they are influenced by the situatedness.
- Construction of memories do not rely on the exact matching between what the system has in the memory and the current designing environment.
- Memories that are constructed may not match the original experience exactly as it was first experienced, but change according to when, where and what the memory system is cued with.

This process is a contradistinction to memory as a retrieval process where there is a memory stored that can be retrieved directly and the retrieval has no effect on the memory.
Consider MIT's remembrance agent (Rhodes and Starner, 1996) which operates in a note-taking application from where it monitors the themes the user is currently entering and offers links back to previous notes with similar themes. It is possible to imagine a future with a situated remembrance agent embedded in a mobile computing device that is able to cross reference different media files captured in related situations or with related contexts. The situatedness of the application gives the opportunity to offer unexpected value while the user is primarily engaged in an activity. As we enhance situations in the physical world with relevant connections, our experiences will be greatly enriched.

2.3 An Integration of Context Awareness and Constructive Memory

In our research we are interested in investigating the impact of the incorporation of a constructive memory component in context aware applications. To the best of our knowledge, amongst those learning architectures, none supports a specific feature that is perceived as highly essential, the ability to construct contexts from memories in uncertain environments. While for a context aware application to be able to be effective in the real world, it needs to be able to function well in uncertain environment. Context awareness is widely used in intelligent personal assistant application domain. Context awareness provides more sophisticated human-computer interaction level, which is crucial in intelligent personal assistant. One particular field which commonly utilised context awareness is reminder systems. Reminder systems which are integrated with context awareness proven to be the one with higher level of effectiveness, due to its ability to allow the use of rich context to specify when a reminder should be delivered. Previous work in this area mainly consists of context aware reminders systems, such as Stick-e Notes (Brown, 1995), W-MAIL (Ueda et al., 2000), CybreMinder (Dey and Abowd, 2000) and Conference Assistant (Dey et al., 1999b). Based on this research within the intelligent personal assistant application domain, we will like to focus on a context aware reminder system to illustrate some possible benefits of combining context awareness and constructive memory. In the particular field of reminder systems, users have chosen to utilise reminder systems to ease them in doing their tasks. Hence, an ideal reminder system will require minimal amount of effort from the user in doing its tasks. In this project we propose the incorporation of a suitable constructive memory algorithm into context aware computing.
3. References


