Towards a dynamic process model of context

Peter Lonsdale
p.lonsdale@bham.ac.uk
School of Engineering
University of Birmingham
Birmingham B15 2TT

Russell Beale
rxb@cs.bham.ac.uk
School of Computer Science
University of Birmingham
Birmingham B15 2TT

ABSTRACT
The complex usage of mobile devices coupled with their limited resources in terms of display and processing suggests that being able to understand the context of the user would be beneficial. In this paper we present a model that describes context as a dynamic process with historic dependencies. This model allows us to i) build a useful, understandable context-aware system in collaboration with content creators and stakeholders; ii) describe this set-up with other system developers; iii) represent the current context state to users and allow them make changes where necessary.

Keywords
context awareness, mobile learning

INTRODUCTION
PDAs, mobile phones, and laptop PCs are used by a variety of users in a variety of different environments. Getting hold of information about the user and their environment and then putting it to good use lets us provide timely support for user activities and allow the user to maintain their attention on the world around them. Context is important because it allows us to make use of the environment in a way that supports the user. Current Nokia mobile phones have a set of modes that determine ring volume, text message notification and suchlike, so that different profiles can be chosen for outdoor use, or when in a meeting, allowing the phone to respond most appropriately. Such choices are made manually by the user, but the principle that the different contexts require different actions is the same. For more advanced systems, we can envisage the scenario of a mobile phone that is aware of its user's location, for example, and will not disturb an important meeting. But the same phone, being aware of its user's call list and calendar will permit a call from a pregnant wife. In this way the user themselves forms part of the environment they occupy, and we can use information about the user themselves to further enhance our contextual model.

What is becoming clear is that there are difficulties in implementing context-awareness. Firstly, how do we get hold of contextual information; and secondly, what do we do with it once we have it?

In order to address these issues, we believe that there is a need for a model of context, to facilitate dialogue about what does and does not constitute context for the purposes of enabling context-aware computing, and to enable flexible re-use of context awareness architectures in a variety of scenarios. The problem with this is that 'context' in itself is all encompassing and recursive – it is difficult in light of this to offer a prescriptive model. It is possible to look at the kinds of things that can be used as contextual data, and to build a model from these examples that can help us explore future implementations of context awareness.

The technological approach versus the user-centred approach
A review of the current literature on context awareness research indicates that there is a polarisation of approaches (for recent reviews, see [4, 5]). Much research can be seen to be driven from a technological perspective, focusing on what current devices, sensors, and software platforms can provide in the way of context aware computing. This approach is understandable given the need to consider the technical aspects of how to acquire and use contextual data. However, the focus on this approach is at the expense of another significant perspective: that of the user. In MOBIlearn [1] we are aiming to work from a user-centred standpoint, identifying the kinds of context awareness that might be required by users in specific scenarios of use, and then implementing a context awareness system around them. Our aim is to provide context aware learning experiences in at least three different scenarios, and our experiences so far have taught us that we need a generalised architecture and model for context awareness to enable useful dialogue between project partners. We therefore suggest that there is a need for an increased attention to the user-centred approach and the need for reusable models. In MOBIlearn, we are aiming for a hybrid approach, working from the user-centred perspective, building re-usable models of context, but at the same time maintaining an awareness of technical constraints.

We consider context not as a static phenomenon but as a dynamic process, where context is constructed through the learner's interactions with the learning materials and the surrounding world over time. For mobile learning, there is an essential interaction between the environment, the user, their tasks, and other users. All of these domains provide information in themselves, and can interact with the others in a variety of ways, building a rich model of the current world and hence allowing the system to be more specific in what it offers the user. The environment contains much ambient information, as do the other users in that space. The learning tasks and the user themselves provide a clearer view of what is important to them, whilst all define the knowledge that is useful and available. A simple example clarifies these concepts: environmental information such as geographical position allows us to provide location-specific information, e.g. for a museum. Other user information such as the identification and presence of another person allows us to create a peer-to-peer network for informal chat. But the combination of the two may allow us to determine that the other user is a curator, and we can provide the mechanisms for one to give a guided tour to the other. The combination of models is potentially richer than each on their own.

OUR IMPLEMENTATION: CONTEXT AWARENESS FOR MOBILE LEARNING

M-learning, the mobile equivalent of e-learning, is an emerging field of research being embraced by manufacturers, content providers, and academics alike. More and more people are carrying mobile computing devices everywhere they go in the form of PDAs, smart phones, and portable computers. There is something compelling about the possibility of being able to take advantage of these devices to offer new ways of interacting with information. Learners on the move can use mobile devices to take their learning materials into a rich variety of environments – the challenge is how to make the best use of this environmental richness provide both intelligent content delivery and engaging learning experiences.

The MOBIlearn project aims to produce an integrated architecture for learners with mobile devices. The system includes support for collaborative learning, an adaptive human interface, and context-aware presentation of content, options, and services. We have been exploring the use of context-awareness as part of a larger m-learning architecture to provide an engaging and supportive learning experience in different environments.

The MOBIlearn context awareness subsystem [7], currently being developed at the University of Birmingham, allows learners to maintain their attention on the world around them while their device is presenting appropriate content, options, and resources that support their learning activities.

For example, learners following a particular course in an art museum will see different content and options being presented to them as they move around the galleries and exhibits. The context awareness subsystem will use contextual information such as location, time, and learner profiling to make recommendations to the content delivery engine about what items should be displayed. Services can also be recommended directly to the user interface: a student who has been struggling with a particular question for some time will be presented with the option to start a chat session with another learner, who may be someone from their own study group, another visitor to the gallery, or perhaps an online student who is visiting the gallery remotely.

Our activities in the MOBIlearn project are centred on specific learning scenarios, of which the art gallery scenario is one example. We have found it useful to describe an underlying model of context that has informed our architecture and enabled relevant discussions between project partners about the use of contextual information in the system as a whole.

Model of context

For MOBIlearn, the purpose of context awareness is to enable learning on mobile devices, and so our approach to describing context and applying this description to producing a usable software architecture is based on this focus. Figure 1 shows the basic hierarchy for our description of context.

Instead of a rigid definition, our intention is to provide a hierarchical description of context as a *dynamic process with historical dependencies*. By this we mean that context is a set of changing relationships that may be shaped by the history of those relationships. For example, a learner visiting a museum for the second time could have his or her content recommendations influenced by their activities on a previous visit.

A snapshot of a particular point in the ongoing context process can be captured in a *context state*. A context state contains all the elements currently present within the ongoing context process that are relevant to a particular learning focus, such as the learner's current *project*, *episode*, or *activity* (see [9]). A learner may at any one time be engaged in a number of simultaneous activities and episodes that relate to one project, and they may have several ongoing projects each of which has its own set of relevant activities and episodes. It is therefore important, from a design perspective, to clearly identify the focus for our current implementation of context awareness.

A *context substate* is the set of those elements from the context state that are directly relevant to the current learning and application focus, that is to say those things that are useful and usable for the current learning system.

Context features are the individual, atomic elements found within a context substate and each refers to one specific item of information about the learner or their setting (for example current learning task or location). In our description of context, context features an indivisible and refer to only one item of relevant information about the learner or their setting.

Note that so far we have not specified what elements of the learner's current context we are interested in – this is done on a scenario by scenario basis to allow for maximum flexibility and to better match the context awareness to the learner's needs.

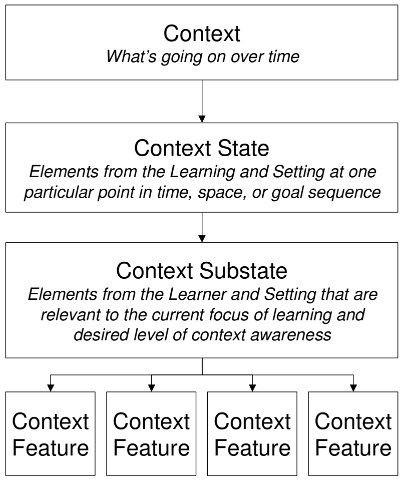

Figure 1: Context hierarchy

Contextual information is also made available to other parts of the MOBIlearn system by means of XML (eXtensible Mark-up Language) documents in an agreed format. At any given time, the current context state is represented as a nested set of context features, all described in XML form. An XML schema for this XML object is an agreed format that allows all components of the MOBIlearn architecture to access this information as and when it is required. Storage of a set of timestamped XML context objects provides the historical context trace that can be inspected and used by subsequent sessions.

CONTEXT-AWARENESS ARCHITECTURE

Figure 2 provides a basic illustration of how the MOBIlearn context awareness subsystem relates to other architecture components and how it provides recommendations to the user. A learner with a mobile device is connected to a content delivery subsystem, which in turn is linked to the context engine. The context awareness subsystem (CAS) collates contextual metadata from sensors, user input, and a user profile. A set of software objects then use this metadata to perform evaluations of the metadata available on a set of learning objects, options, and services. These evaluations lead to recommendations that are then used by the delivery subsystem in determining which content to deliver to the learner. Note that user input to the system is acknowledged as an input source of contextual data: meaningful context is difficult to establish and we aim to include the learner themselves in the context gathering process.

The basic cycle of operation of our context-awareness system is as follows:

1. *gathering* and *input* – of context metadata
2. *construction* – of context substate
3. *exclusion* – of unsuitable content
4. *ranking* – of remaining content
5. *output* – of ranked list of content.

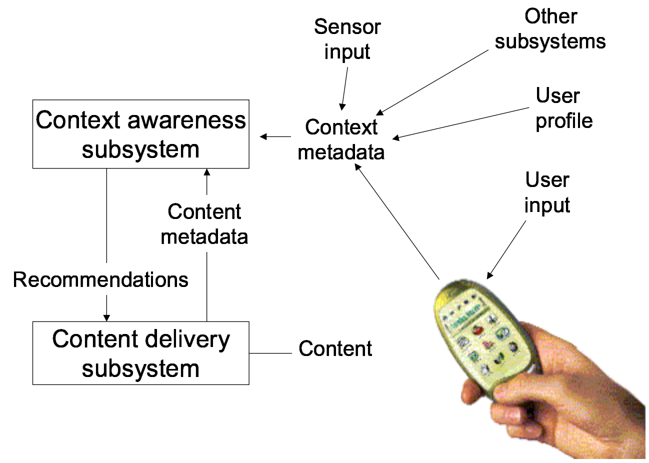

Figure 2: Context awareness in action

The CAS comprises a set of software objects called context feature objects (CFOs) that correspond to real-world context features relating to the learner's setting, activity, device capabilities and so on to derive a context substate, as described above. Data can be acquired through either automated means (for example sensors or other software subsystems) or can be input directly by the user. This context substate is used to perform first exclusion of any unsuitable content (for example high-resolution web pages that cannot be displayed on a PDA) and then ranking of the remaining content to determine the best *n* options. This ranked set of options is then output to the content delivery subsystem. CFOs are created at run-time from a set of definitions provided by the designer(s) of the context-aware experience for which the system is being employed. These definitions specify values for the parameters described below, including the relative salience values for different CFOs and the links between them.

Types of context features

Context feature objects are either excluders or rankers. Items of content that are deemed entirely inappropriate for the current context are excluded. That is to say they are removed from the list of recommended content and not subject to any further consideration. Content remaining in the list after the exclusion process is then ranked according to how well it matches the current context. The ranking process simply increments the score of each item of content that has metadata matching the stimulus values of any particular context feature. The size of the increment depends on the *salience value* of the context feature doing the ranking. Individual CFOs can have their salience values changed so that they exert more influence on the ranking process. Any individual CFO can be de-activated at any time so that it has no effect on the exclusion or ranking processes.

A CFO has a set of possible values, and an indicator of which value is currently selected. It is also possible for CFOs to have multiple sets of possible values, with the current active set being determined by the current value of another linked context feature. Whilst this has no bearing on the recommendation process, it is important in terms of providing an inspectable model of the context state to the user, who can observe the influence of one context feature on another. For example, options relating to current activity can change depending on the user's current location.

Linked context features

Each context feature object responds to only one metadata tag and performs either an exclusion or ranking function. To achieve more complex filtering of content, CFOs can be linked together so that their function can depend on the state of other context feature objects. Link objects are used to send either the *values* of context features or the *time* they have held that value to other context features. Criteria on that link determine whether action should be taken.

For example, we might have a context feature that responds directly to input from a sensor network specifying the location of the user. Another context feature infers the level of interest of the user by taking input from a link that acts on the *time* the location feature has had its current value. A user dwelling in one place for a longer period implies a higher level of interest in that location. A third context feature may respond to user input that can over-ride the inferred level of interest – this uses a link object that acts on the *value* input by the user. Conflicts between links and context features are resolved using salience values which specify the relative importance of each. These salience values are at present specified by the designer(s) of the context-aware experience, but more automated methods of conflict resolution could be employed in future iterations.

Output

The ordered list of ranked items of content is passed to delivery subsystems for use in determining exactly what content should be made available to the user. In this way, the context-awareness sub-system has no way of specifying exactly what is made available – the system is intended only to make recommendations to the system and to the user. This method of recommendation is preferred so that should the system make a mistake, and make inappropriate recommendations, its output does not override selections made elsewhere in the system (for example, the user might specify a particular page of content and then not want that item to be replaced by another).

Metadata schema

We have developed a metadata schema to facilitate the appropriate storage and transfer of contextual data among the different components in the MOBIlearn system. This schema maps on to our hierarchical description of context itself and offers a generic and reusable template for exchanging data about the current context. This schema is also intended to map very closely onto the underlying design of our current software architecture – all context feature objects in the system are implemented as Java objects with attributes that mirror those shown in the schema. Translating from Java object attributes to XML is therefore an efficient way for the system to make its current state available to other system components. A diagrammatic representation of this context schema is shown in Figure 3.

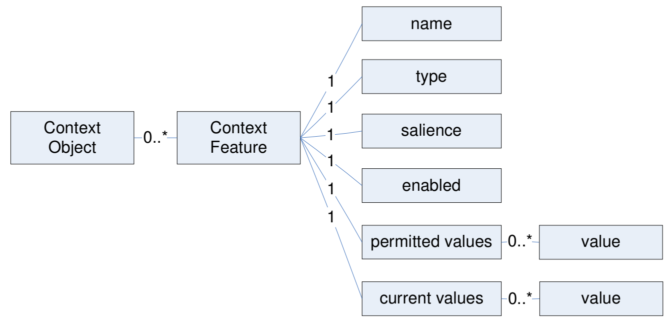

Figure 3: Context meta-data schema

The root element, ContextObject, is the entire set of all of the contextual features that are currently maintained by the system. Each context feature element corresponds to a software object that listens for changes in a specified feature of the real-world context and responds accordingly. Typically this response will be a re-ranking of the available content to match the new context. This schema is deliberately designed so as to not be prescriptive in itself about which elements of the context state we are currently interested in. Each context feature element contains sub-elements (not all shown in figure) that allow the description of each software context feature in terms of its name, type, enabled status, current and permitted value(s), salience value, input source, and category. The element 'category' is used to indicate whether this feature relates to environmental or user data – we have identified both of these sources as important for enabling context aware learning applications. Link objects have a similar schema that is used internally by the context system.

We address the need to monitor and respond to context over time by storing a series of context objects, each of which has its own timestamp and can be marked with any other data that relates it to a particular episode, activity, or task. Our aim is to use these 'context traces', made up of groups of context objects, to influence the context of a future use of the system. For example, a learner who has already visited an art gallery on a previous occasion would be able to retrieve their previous context trace and use it to better guide the system for this visit. The previous context would become part of the current context, thus satisfying our identified need for historical dependencies.

With contextual metadata available in this XML format, it is a relatively easy process to apply the exclusion and ranking process outlined earlier. Metadata relating to available learning objects is read into the system as a series of XML documents adhering to the IMS 1.2 schema for learning object metadata [6], and comparison of these two sources of metadata yields contextually relevant recommendations. As we have already found, metadata relevant to *mobile* learning are not fully addressed by the IMS schema, and so we worked on extending this and other schemas to rectify this problem. For more details of this work, see [3].

Applying context awareness

The output from the context-awareness subsystem can be used in a number of ways:

- to automatically provide directed routes or relevant materials
- to affect ordering and presentation of the same material
- to provide supplemental information to be accessed by user only if they are stuck

These different uses represent different points on the spectrum of system automaticity vs. user control. There are pros and cons at each end of this spectrum, exaggerated by the nature of interacting with mobile devices. For example, limited screen size means that it is desirable to keep automatically generated recommendations in the background as much as possible, but because mobile devices are typically embedded in a changing context we also need to be able to quickly notify the user of any changes. There are challenges for user interface design, as well as representation issues and the need to support user goals without being intrusive or disruptive.

CURRENT STATUS AND NEXT STEPS

The MOBIlearn context awareness subsystem is currently deployed as a web service that can be used by other components in the MOBIlearn system (these components are by design geographically distributed). We have an integrated an ultrasound based positioning system [2] and a basic user profiling mechanism to provide context aware support to visitors to an art gallery. This is due for user trials in September [8]. We will explore the issues surrounding context representation and user interface design through a series of comparative trials. Our research questions centre on the use of context-awareness as a metaphor for content navigation, as opposed to a typical web-model with which users may be more familiar. There are issues therefore relating to maintaining the user's locus of control and providing adequate visibility of system status, whilst at the same time providing effective context-aware content delivery.

ACKNOWLEDGEMENTS

This research is supported by the European Commission's 5th Framework Programme on Information Society Technologies, project *MOBIlearn* IST-2001-37187, http://www.mobilearn.org.

REFERENCES

1. Bormida, G. D., Lefrere, P., Vaccaro, R., and Sharples, M. (2002). The MOBILearn Project: Exploring New Ways to Use Mobile Environments and Devices to Meet the Needs of Learners, Working by Themselves and With Others. In *Proceedings of European Workshop on Mobile and Contextual Learning*, Birmingham, UK, The University of Birmingham, UK.

2. Byrne, W., Lonsdale, P., Sharples, M., Baber, C., and Arvanitis, T. (2004). Determining location in context-aware mobile learning. In *Proceedings of MLEARN 2004*, Bracciano, Rome, LSDA.

3. Chan, T., Sharples, M., Vavoula, G., and Lonsdale, P. (2003). Educational metadata for mobile learning. In *Proceedings of Second IEEE Workshop on Wireless and Mobile Technologies in Education*.

4. Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research: *Technical Report TR2000-381*, Dartmouth College, Hanover, NH

5. Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1): 19-30.

6. IEEE 1484.12.1-2002 (2002). Draft standard for learning objects metadata, IEEE 1484.12.1-2002.

7. Lonsdale, P., Baber, C., Sharples, M., and Arvanitis, T. (2003). A context awareness architecture for facilitating mobile learning. In *Proceedings of MLEARN 2003*, London, LSDA.

8. Lonsdale, P., Baber, C., Sharples, M., Byrne, W., Arvanitis, T., Brundell, P., and Beale, H. (2004). Context awareness for MOBIlearn: creating an engaging learning experience in an art museum. In *Proceedings of MLEARN 2004*, Bracciano, Rome, LSDA.

9. Vavoula, G. and Sharples, M. (2002). KLeOS: A personal, mobile, knowledge and learning organisation system. In *Proceedings of IEEE International Workshop on Wireless and Mobile Technologies in Education*, IEEE Computer Society Press.