Mobile Context Aware Systems: the intelligence to support tasks and effectively utilise resources

Russell Beale[1] and Peter Lonsdale[2]

[1]r.beale@cs.bham.ac.uk
School of Computer Science
[2]p.lonsdale@bham.ac.uk
School of Engineering
University of Birmingham
Birmingham B15 2TT UK

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. We also describe a software architecture to support this model, and evaluate its effectiveness in a mobile learning scenario. Preliminary results from our evaluation suggest important issues for consideration in the continuing development of context aware systems and interfaces, including the need for appropriate representation of contextual data to the user, and maintaining a balance between effective support and intrusion.

1 Context awareness and mobile computing

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.

Having this kind of automated filtering going on is useful for any user. But for a user of a mobile device, this kind of support becomes even more salient. When users are mobile, they are typically involved in other activities not focused on the device itself. For example, visitors to an art gallery would like to maintain their attention on the great works they are admiring, rather than having to perform content searches on their PDA. A tool that can keep up with what is going on with the user and their environment can allow the user to maintain their attention on the world, and can provide timely support for the user's activities. In addition, mobile devices are usually very limited when compared to desktop devices. These limitations mean that steps taken to reduce the quantity and complexity of the information they have to potentially provide allows them to be more efficient and effective. A large amount of interesting work is being done on display approaches to better represent too much information in too little space with too few resources, but effective assistance to that work can be given by systems that reduce the amount of necessary information to be displayed has to be beneficial.

The reason for modelling context is to better understand the user's activity - in our case, of mobile learning. This in turn leads to the design of systems that deliver more appropriate learning content and services. This is useful in three respects. Firstly, it relates the services to time and location, and to the learner's needs and interests, ensuring that they are useful, learnable and enjoyable. Engagement is increased since if the system can provide appropriate information at the opportune time, it can produce a more compelling learning experience. Matching the correct level of information using the most appropriate learning style for a particular user can produce a more effective and enjoyable learning experience. Secondly, it provides for more effective use of resources, which is especially important in the mobile situation, with many different limitations – device processing power, display ability, media capabilities, network bandwidth, connectivity options, intermittent connections – and other aspects of the situation competing for attention. Thirdly, by providing more appropriate information delivered most effectively, it allows the user to focus much less on the technology and more on the actual situation they are in. By producing a system that is responsive to the user's changing attentions and their associated changes in need, mobile learning systems support a much more exploratory, opportunistic and ad hoc approach to learning that potentially suit their users much more.

2 Modelling context

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.

2.1 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 [3, 4]). 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.

3 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 [6], 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.

3.1 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 [7]). 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.

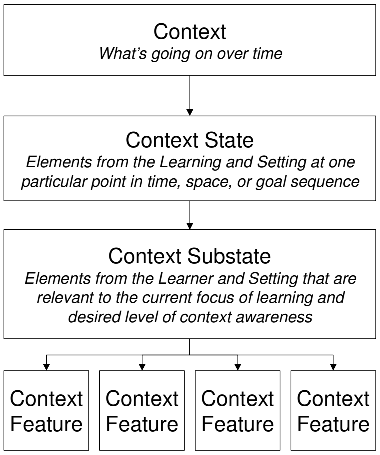

Context
What's going on over time
Context State
Elements from the Learning and Setting at one particular point in time, space, or goal sequence
Context Substate
Elements from the Learner and Setting that are relevant to the current focus of learning and desired level of context awareness
Context Feature
Context Feature
Context Feature
Context Feature

Fig. 1. Context hierarchy

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.

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.

3.2 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.

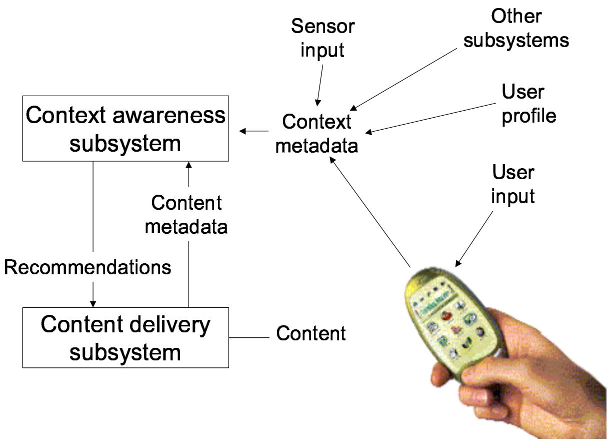


Fig. 2. Context awareness in action

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.

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.

3.3 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.

3.4 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. For example, we might choose to have a context feature object that excludes content based on its file-size – such CFO should be active if the learner is using a low-bandwidth connection, but should remain quiescent if a high bandwidth connection is available. By creating a context feature that responds to bandwidth availability and allowing it to control the status of the context feature that responds to file-size, we can easily create a pair of context features that respond to a more complex context. This linking process is transparent to the user and to individual CFOs, so long chains can easily be created to cope with complex situations.

3.5 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).

3.6 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.

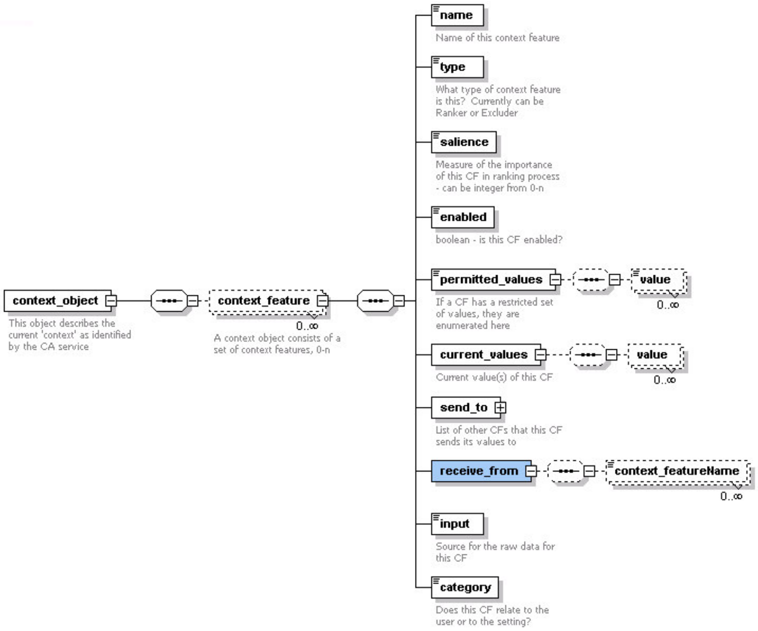

context_object
This object describes the current 'context' as identified by the CA service
context_feature
0..∞
A context object consists of a set of context features, 0-n
name
Name of this context feature
type
What type of context feature is this? Currently can be Ranker or Excluder
salience
Measure of the importance of this CF in ranking process - can be integer from 0-n
enabled
boolean - is this CF enabled?
permitted_values
value
0..∞
If a CF has a restricted set of values, they are enumerated here
current_values
value
0..∞
Current value(s) of this CF
send_to
List of other CFs that this CF sends its values to
receive_from
context_featureName
0..∞
input
Source for the raw data for this CF
category
Does this CF relate to the user or to the setting?

Fig. 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 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, as well as the set of other context features that this feature can send to or receive from. 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.

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 [5], 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 [2].

4 User Trials

We have run some small scale user trials to assess the impact of context aware content delivery on user's experience and to provide some formative evaluation of our work so far. The results of the these trials, presented below, are being used to inform the design of our next prototype.

4.1 Software setup

All participants used a prototype of the context awareness subsystem implemented in Java. The prototype comprised a single server or management application connected to several clients.

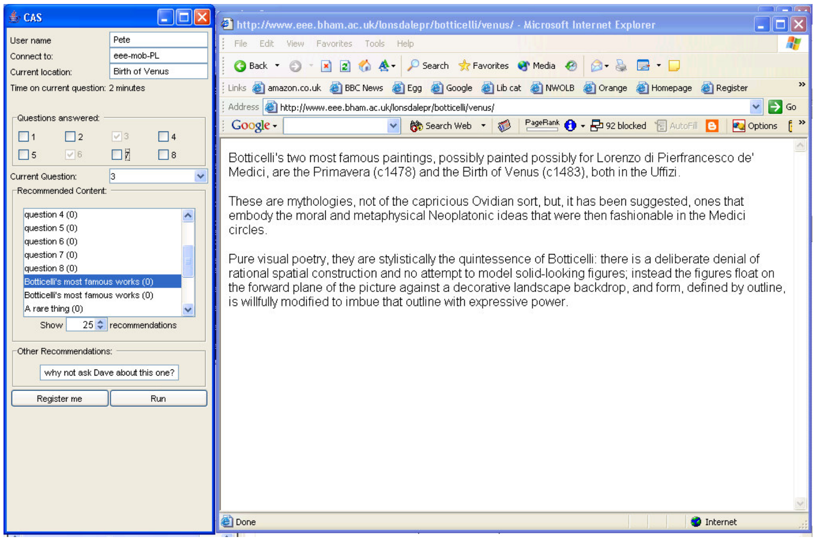
The manager application allowed the experimenter to manage and monitor several participants simultaneously, updating their location and observing the current question they were working on as well as previous completed questions. This application ran on a laptop PC that was used by the experimenter during the session. The laptop was connected to the client software over the wireless network using socket communications.

The CAS client ran on the tablet PCs (Fujitsu Stylistics) that were given to the participants for the session. This application ran beside the Internet Explorer browser and offered recommendations of content, questions, and people that were deemed relevant to the learner's current context. Context was determined from a combination of location of user, location of other users, current question being answered, and previous questions answered.

The system provided recommendations of content, questions, and communication with other learners, depending on the participant's current location and question. For example, a learner standing in front of La Primavera would see content relevant to that painting near the top of their content list, with the top item being most relevant to the La Primavera *and* their current question. If another participant who had already answered the current question was also at La Primavera, the system would suggest talking to them.

4.2 Method

Participants were divided into groups of 2-4 and asked to play the role of art history students following a study guide in an art gallery. They were each given a Fujitsu Stylistic tablet PC and were told that they would get help in finding the answers to the questions from the context awareness system running on the tablet. The basic functionality of the system was explained to them and they were given a brief demonstration of how to use it. Participants were asked to move around the simulated art gallery containing 6 paintings whilst trying to find the answers to 8 questions given to them at the beginning of the session. Participants walked around a set of 6 paintings located in a small room intended to represent an art gallery. They were told that their location relative to specific paintings would affect the recommendations given to them by the system.



User name: Pete
Connect to: eee-mob-PL
Current location: Birth of Venus
Time on current question: 2 minutes

Questions answered:
□ 1 □ 2 ☑ 3 □ 4
□ 5 ☑ 6 □ 7 □ 8

Current Question: 3

Recommended Content:
question 4 (0)
question 5 (0)
question 6 (0)
question 7 (0)
question 8 (0)
Botticelli's most famous works (0)
Botticelli's most famous works (0)
A rare thing (0)

Show 25 recommendations

Other Recommendations:
why not ask Dave about this one?

http://www.eee.bham.ac.uk/lonsdalepr/botticelli/venus/ - Microsoft Internet Explorer

Address: http://www.eee.bham.ac.uk/lonsdalepr/botticelli/venus/

92 blocked

Botticelli's two most famous paintings, possibly painted possibly for Lorenzo di Pierfrancesco de' Medici, are the Primavera (c1478) and the Birth of Venus (c1483), both in the Uffizi.

These are mythologies, not of the capricious Ovidian sort, but, it has been suggested, ones that embody the moral and metaphysical Neoplatonic ideas that were then fashionable in the Medici circles.

Pure visual poetry, they are stylistically the quintessence of Botticelli: there is a deliberate denial of rational spatial construction and no attempt to model solid-looking figures; instead the figures float on the forward plane of the picture against a decorative landscape backdrop, and form, defined by outline, is willfully modified to imbue that outline with expressive power.

Fig. 4. Context awareness client

During the session the experimenter employed a "Wizard of Oz" evaluation method, monitoring each participant's location and updating their client software using a remote management application. The experimenter was also able to monitor which question each participant was currently working on and which questions had already been answered. At the end of the session, the concept of context aware content delivery was discussed with the participants and they were asked for feedback about their use of the system in an informal, free-form interview.

4.3 Results

Feedback was gathered from users about the usefulness and usability of the system. This feedback was used to derive a formative evaluation of our current implementation. The following issues were identified:

- *It worked*: Most users were able to quickly find relevant information and successfully answer the questions.
- *Interface and representations:* Many users were confused about what we were trying to represent with the interface, and were not sure why their recommendations were changing or how they could best use the recommendations list to answer the questions.
- *Understanding*: some people weren't quite sure why the system did what it did, and were surprised by the constantly changing list of options. Demonstration and explanation did not seem to help with this – when there was a misunderstanding it was due to a lack of intuitiveness about the display of the context-dependent recommendations
- *Distraction vs Engagement*: offering multiple choices either led to sidetracking or encouraged people to further their exploration of the content. Both of these suggest that users were engaging with the experience, but this could become a concern if we are trying to design a specific programme of learning. Options that distract users from their current task focus need to be avoided, and so it is possible that some limits need to be set on exactly how much contextually based recommendation is done.
- *Mixed content:* there is a need to distinguish questions, content, physical resources. Offering recommendations of all of these in a single, integrated display seemed to be confusing, especially in combination with the lack of an intuitive, easily grasped model of what was actually going on and why.
- *Temporal context*: Context is often used in a snapshot sense: what is happening now, where am I at this moment, and so on. However, there are many much longer-term aspects to context (e.g. task, learning progress, life goals) and it is not clear how to best represent and use this information in the context system. The fundamental issue is that we need to be able to model and then provide support for users across multiple activities, episodes and projects, with the history of previous support playing an integral role in determining future actions.

5 Conclusions and Next Steps

The context awareness system that we have developed demonstrates that it is useful and works so that people are supported in their actions. However, it is clear that there is not a sufficiently effective model of context communicated to the user, so that they are often confused as to why the system is changing its recommendations to them. This is partly an interface issue, where the separation of the different parts of the system is unclear, and partly a conceptual one, since users are not used to systems being dynamically adaptive on such a scale. It may be that hiding more of the workings of the context engine and simply presenting the results using an appropriate metaphor would be more effective; for example, using an avatar gallery guide, which users could easily ascribe some form of 'intelligence' and hence more easily accept changing suggestions.

We are now focussing on extending the context engine and the sensor inputs, and integrating them with the meta-data schema and set of learning objects, to provide a rich environment for more extensive user trials and evaluations. We will also continue our related work on developing new knowledge-based approaches to implicit modelling, to provide more effective models without overloading users with questions. The intention is to allow the system environment to develop models without overtly intruding on its users. Knowledge-based systems inevitably make mistakes, and we are exploring how to resolve conflicts between parts of the system that reach different conclusions and how to cope with interpreting and reconciling heterogeneous sources of data.

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. 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*.
3. Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research: *Technical Report TR2000-381*, Dartmouth College, Hanover, NH
4. Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, **8**(1): 19-30.
5. IEEE 1484.12.1-2002 (2002). Draft standard for learning objects metadata, IEEE 1484.12.1-2002.
6. 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.
7. 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.