Workshop on Context-awareness: Position Paper/Extended Abstract

Peter Brown and Gareth Jones

Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK
pjb@ukc.ac.uk

ABSTRACT

We discuss two characteristics of context-awareness, and argue that, as context-aware applications become more mature, simplistic approaches need to be replaced by richer algorithms. Our perspective is from context-aware retrieval of information. This contribution relates most closely to questions 3 and 4 of the Call for Participation, though it relates, we think, to the other questions too.

The place of context-awareness

We would like to make a general point first, as this will put our more specific points in perspective.

If you take a wide view by comparing an established software capability, 'word processing', with context-awareness, you see that the former relates directly to a user task whereas the latter does not. If users want context-awareness, they only want it as part of a wider application that relates to their task.

Given that there is unlikely to be a pure context-aware application, we should not spend too much time trying to define what is context-wareness and what is not. Moreover precise definitions are a problem even in more self-contained disciplines. Continuing the parallel with word processing, the characteristics of word processing systems vary considerably, particularly if you include such systems as Latex and *troff*. Moreover people's conception of word processing is continually developing over time, especially with web influences.

Although trying to find a watertight characterisation of context-awareness may be a fruitless task, trying to gain a better understanding of its most common components certainly is not. Hence this workshop. We would like to concentrate on two common components: proactiveness and the way context changes. These two are both fundamental. Arguably proactiveness follows directly from the adjective 'aware': if an application is context-aware, this implies some element of proactive behaviour. Clearly change of context is fundamental too, since applications with a constant context are not interesting.

Our perspective

Our interest is in the field of context-aware retrieval of information. We have recently been working in relating it to the established fields of Information Retrieval (IR) and Information Filtering. Most context-aware applications involve the retrieval of information, and that drives our interest. The retrieved information may be presented directly to the user, or it may indirectly affect the application, e.g. by changing the user interface or triggering a program in the background.

Context-aware retrieval

We will look at context-aware retrieval in a little detail, because of the issues we believe it raises for context-awareness in general. The elements of context-aware retrieval are:

(1) the user's current context; in some applications there are many simultaneous users and thus many current contexts.

(2) the collection of information to be retrieved from. In IR the term *document* is commonly used for each item in the collection, though it is not assumed an item is necessarily a textual document.

(Actually there is a third element: the recipient of the information. In applications such as tracking, or applications where the 'users' are, say, farm animals or widgets on a production line, the recipient is different from the user. However for simplicity we will here assume the the user is the recipient of the retrieved information.)

Proactiveness can be embodied in (1) or (2) above or in both. Case (1) is a very common one: for example a location field of the user's current context may be set and continually reset proactively via a GPS sensor. Additionally any document in (2) can have a context attached to it, with the mechanism that the document is proactively 'triggered' if this context matches the user's current context. Such proactive behaviour is taken for granted in many context-aware applications.

We believe that in most real applications proactive behaviour needs a considerable degree of intelligence — though not necessarily of the AI kind — rather than being a simple mechanical process.

If we consider setting the user's current context, there is often a mismatch between the sensors, which tend to provide data at a low level of abstraction, and the user, who works at a higher level. Clearly the application must aim to meet the user's level of abstraction. As an example, a context-aware office application might want to record if a user was busy or not. The application would then need to derive this information intelligently using low-level data from sensors that recorded whether doors were open, seats sat upon, phones in use, etc [1, 2].

If we consider the other element of proactive behaviour, triggering documents when their context is matched, we have become increasingly convinced that the matching process needs to be intelligent. This mirrors the development of IR, where, driven recently by web search engines, huge amounts of research and development have been devoted to making the retrieval process *efficient* and *effective*. Efficiency involves retrieving from massive amounts of information quickly and with minimal use of resources; this will surely be a need in future industrial-strength context-aware applications — though most current applications are pilot ones and deal with small amounts of data. Effectiveness is concerned with providing just those documents that the user needs to see. If anything, there is an even bigger need for relevance in context-aware applications: for example, experiments by Rhodes [3] show that if users are presented with documents of marginal relevance they will soon abandon the application.

Thus we believe that the strides made in IR will need to be matched in context-aware retrieval — indeed some of the IR technology can be carried over directly. Note that most IR systems do not provide their intelligence by AI methods: instead they use methods based on statistical probabilities and on preprocessing of static information.

The need for efficiency and effectiveness leads us into our second topic: changing context. On the face of it, context-aware retrieval is much harder than IR, and indeed represents what some have called the 'grand challenge' [4]. The difficulties come from two directions. The first difficulty is change: the current context may be changing very rapidly, and there may be a continuous need to retrieve new information — perhaps more frequently than once a second; furthermore the collection of documents may be changing too, e.g. when it relates to road traffic or share prices. The second difficulty is multiplicity: there may be both multiple information sources and multiple users.

These difficulties will, we believe, make some potential context-aware applications impractical, at least in the short term. There is, however, a knight in shining armour that

can often save the application. The knight's motto is: *change is often gradual and semi-predictable*. Our work [5] has indicated that when change is gradual and semi-predictable, this can be exploited in many ways. Examples are:

- the retrieval process can be speeded up, because some documents are more likely, whereas others need not be monitored regularly. Thus in a location-aware application one only needs to look at the documents that are 'close'.
- caches can be built to contain the documents that are likely to be needed in the near future.
- relevance can be improved by extrapolating the way context is changing. Thus if the user is moving in a regular way, information just ahead is likely to be more relevant than information just behind or out to one side. There is corresponding work here relating to users of cellphones [6]: tracking users and successfully anticipating their future location can lead to great gains in efficiency; some of this work is based on layers, reflecting how closely predictions can be made.

To make the point more strongly: we believe that *many applications will only be possible when context is changing gradually*, and these applications will break down, either for performance reasons or relevance reasons, if change is lumpy and unpredictable.

Conclusion

We have concentrated on two common characteristics of context-aware applications: proactive behaviour and change of context. Our conclusions are that

(1) proactive behaviour in real applications is likely to require intelligent rather than simple mechanical algorithms, though these algorithms may not relate to traditional AI.

(2) a key characteristic is how context is changing, and many applications only become feasible if change is semi-predictable. In cases where it is, the knowledge of change can be exploited in a large number of ways.

In essence, if we want the richness of communication highlighted by Abowd and Dey[7], we need to plant some of our riches into the above two areas. These conclusions are driven by experience in context-aware retrieval, but our guess is that they apply equally well to other fields of context-awareness: it will be of considerable interest if workshop participants can throw light on this.

References

[1] Newman, W.M., Eldridge, M.A. and Lamming, M.G., 'Pepys: Generating Autobiographies by Automatic Tracking', *Proc. ECSCW '91*, Amsterdam, September 1991.

[2] Cooperstock, J., Fels, S., Buxton, W. and Smith, K.C. 'Reactive environments: Throwing away your keyboard and mouse', *Comm. ACM*, **40**(9), pp. 65-73, 1997.

[3] Rhodes, B.J. 'The Wearable Remembrance Agent: A system for augmented memory', *Personal Technologies*, 1 pp. 218-224, 1997.

[4] Oard, D.W. and Marchionini, G. *A conceptual framework for text filtering*, Report EE-TR-96-25, Univ. of Maryland, 1996.

[5] Brown, P.J. and Jones, G.J.F. 'Context-aware retrieval: exploring a new environment for information retrieval and information filtering', to be submitted for publication, 2000.

[6] Das, R.E. and Sen, S.K. 'Adaptive location prediction based on a hierarchical network model in a cellular mobile environment', *Computer Journal*, **42**, 6, pp. 474-486, 1999.

[7] Abowd, G.D. and Dey, A.K. 'Towards a better understanding of context and context-awareness', panel statement in Gellerson, H.-W. (Ed.) *Handheld and Ubiquitous Computing*, Springer, pp.

304-5, 1999.