

Proof-of-Concept Demonstrators and Other Evils of Application-Led Research : A Position Statement

Nigel Davies

Computing Department
InfoLab 21, Lancaster University
Lancaster, England

nigel@comp.lancs.ac.uk

A. Why should *some* people be doing application-led research ?

Application-led research is vitally important – applications provide designers and developers of ubiquitous computing systems with requirements, drive progress towards solutions, validate technology and provide insights into technology use. They offer the chance to demonstrate the usefulness of complex systems components that might otherwise be difficult to appreciate and they are, of course, invaluable in encouraging investment to fund further research.

Application-led research is typically equated with research in which a specific application is created, deployed and evaluated – typically as part of a project to develop an underlying technology or concept. This allows us to develop complete systems that can be trialed in real operational environments. In the fields of mobile and ubiquitous computing this is especially important since the target deployment environment often has very little in common with the laboratory environment in which the system was developed. As a result systems often require substantial redesign once they are deployed. For example, assumptions about network availability and quality, user behaviour or application utility are often shown to be flawed once the system is out “in the wild”. In [1] Kjeldskov argues that, in general, laboratory based user studies provide better and more accurate results than field trials. In my experience this is simply not true – it is impossible to understand ahead of time the impact of the environment on technology (or indeed, the impact of technology on the environment) and this is often critical to system design. Work such as that of Barton [2] and Morla [3] on simulation and test environments for ubiquitous computing are making some inroads on the necessity to deploy. However, progress in this area is slow and these environments provide only a partial sense of what it is like to deploy a ubiquitous computing environment.

B. Why should *most* people *not* be doing application-led research ?

Given all these benefits the reader might reasonably conclude that more people should be carrying out application-led ubiquitous computing research. If application-led research means research that attempts to meet the demands of ubiquitous computing applications (however frivolous these applications are) then the answer is, of course, yes – researchers should always be trying to address some clearly identifiable problem. However, “should more researchers be developing ubiquitous computing applications ?” – categorically not! The first and most obvious reason for this is that developing and trialing applications is a massive undertaking and diverts resources from undertaking more fundamental research.

For those who have not done full scale ubiquitous computing application development and deployment it is hard to convey the enormity of the task. As a specific example consider the GUIDE project [4] – widely considered as an example of a successful piece of application-led research and inspired by the Cyberguide system [5]. To develop the application took two researchers the best part of two years. The project also employed several students to capture content and then to conduct the field trials. On a really good day in Lancaster running a trial we may get 5 users. On an average day we get 2 or 3 and very often researchers can spend all day in the city without collecting any data (of course this could be a function of the appeal of our mobile tour-guide rather than a general observation about application-led experiments!). Systems have to be constructed to be robust and all the little practicalities such as privacy, liability and personal security contribute to the overhead of running sensible trials of applications. For many groups these costs are simply prohibitive. For those that can afford the investment there is still a question of whether application development and deployment are the best use of resources.

However, even if resources are plentiful there is a more fundamental reason why most research projects should not do ubicomp application development and that is simply because it is, in most cases, the wrong

tool for the job! Most projects justify application development as part of developing a “proof of concept demonstrator” or such like. The problem, very often, is that there is no actual concept to be proven. Either the concept has already been proven viable (there really is no need to prove again that we can build a context-aware tour guide), is never in any doubt (we know we can build location-based services) or is not actually proved by the demonstrator (proof is a very strong term!). This is why Kjeldskov’s arguments have some validity – in many cases laboratory based tests will do just as good a job of evaluating a concept as a badly performed field trial – and the resource issues discussed above mean that many field trials are badly performed.

C. And if you *really* have to do application-led research, what should you do ?

Those researchers that feel they really have to build applications to demonstrate their work should look to minimize the cost of doing the application development. In practice this should mean reusing applications developed elsewhere. For example, when benchmarking a new operating system no-one builds an application suite on top from scratch and the same should be true for ubiquitous computing. Of course, in addition to reducing costs, reusing applications has an additional benefit – it allows *comparative* studies to be carried out. One of the core reasons for lack of progress in mobile and ubiquitous application-led research is that researchers rarely compare their own work to that of others and hence it is hard to know if we are making progress forwards or backwards. In some areas of the subject this is not the case. For example, the availability of data from experiments by Intille et al. at MIT (see <http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/>) make it possible for anyone developing algorithms for smart environments to systematically compare their results with those of other researchers. This can be contrasted those labs that do not release trace data (and shall remain nameless) – making it impossible for any third party to validate their work without constructing an identical experimental set-up from scratch.

Consider GUIDE once again. Since the project completed there have been dozens of other tour guides developed. At the ubiquitous computing summer school held in Dagstuhl in 2002 I surveyed the audience and was dismayed to discover that approximately half the audience were working on developing some form of ubiquitous or context-aware tour guide! All subtly different of course! However, to date (and to the best of my knowledge) we have not received a single request from any other research group to participate in any form of comparative study – no-one has requested a copy of the GUIDE system from us for such a purpose, nor has anyone submitted their system and asked us to evaluate it against GUIDE. This despite the fact that we designed it specifically to be portable to other cities and that we have successfully reused components for some years (though to be fair we tend to reuse content, design and UI features more than core code). Similarly, none of the pervasive computing middleware platforms that have been developed have been used to support GUIDE. Of course, it should be noted that we have not gone out of our way to request systems to evaluate GUIDE against either – blame should be apportioned in equal measure!

D. The way forward ?

As researchers we are in a strong position to influence how research is conducted through the peer review process. To improve application-led ubiquitous computing research I suggest that we adopt the following four point action plan.

- (i) **Clarify the distinction between application-led research and application development.** As a first stage we need to remind people that application-led research does not necessarily mean carrying out application development.
- (ii) **Stop most researchers from developing applications.** Whenever a grant proposal, student dissertation outline or research workplan that contains an application development phase comes our way to review we should rigorously evaluate whether application development is really the best way to conduct the research. Proposals for “proof of concept demonstrators” should be viewed with particular suspicion.
- (iii) **Stop those that need to do application-led work from developing applications from scratch.** One easy way to do this is to demand that papers present the results of comparative experiments. In other words, how did your system compare to existing systems in the same operational environment. Of course this also means that as application developers we will need to develop our applications such that they can be reused by other researchers.

- (iv) **Develop metrics for ubicomp applications.** Comparative experiments of the type suggested in (iii) need metrics and the development of these should be a high priority goal for our community.

D. Closing Thoughts

In order to change from a subject driven by a vision to one with clearly defined goals we have to be able to measure our progress towards such goals. This, inevitably, means we have to do more comparative analysis between systems. This in turn means we need to develop common metrics and test environments – an important component of which is applications. By encouraging application reuse and comparative analysis we can accelerate progress towards a deeper understanding of what makes a good ubiquitous computing system and thus make steady progress towards Weiser's vision [6].

E. Acknowledgements

This position statement is based on my experiences of the GUIDE project and I would like to acknowledge the contribution of all those who have worked on the project and my colleagues at Lancaster for their insights.

References

- [1] J., Skov M. B., Als B. S. and Høegh R. T., "Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field", Proc of the 6th International Mobile HCI 2004 conference, Glasgow, Scotland. Lecture Notes in Computer Science, Berlin, Springer-Verlag, pp. 61-73
- [2] Barton, J. and Vijayaraghavan, V., "UBIWISE: A Simulator for Ubiquitous Computing Systems Design", tech report HPL-2003-93, Hewlett-Packard Labs, 2003.
- [3] Morla, R. and Davies, N., "Evaluating a Location-Based Application: A Hybrid Test and Simulation Environment", IEEE Pervasive Computing, Vol, 3, No. 3, July 2004.
- [4] Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C., "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences", Proc of CHI'00, Nether-lands. 17-24 March 2000.
- [5] Long, S., Kooper, R., Abowd, G.D. and Atkeson, C.G. Rapid Prototyping of Mobile Context-Aware Applications: The CyberGuide Case Study. Proc. of 2nd ACM International Conference on Mobile Computing (Mobicom). 1996.
- [6] Weiser M. "The computer for the 21st century" Scientific American, Sep 1991, pp. 94-