Richard Beckwith, Intel

What is to be done?

This document will propose that ethnographic and other qualitative methods can be useful in establishing metrics for selection, analysis, and evaluation of ubiquitous computing systems. After summarizing the work we've done in this field, we will end with some specific recommendations for further research.

We do application-led research because we believe that this is the quickest route to understanding how ubiquitous computing -- and allied its research -- must develop. In answer to the question "What do we need to do to enable ubiquitous computing?"; we say that application-led research is a partial answer. And we do need to know how to enable ubiquitous computing if ON World, Inc.'s prediction of a seven-billion dollar market for wireless sensor networks in 2010 is to happen.

The question framing this workshop is different, though. "What makes for good application-led research in ubiquitous computing?" This is a more difficult question. First, application-led research encompasses research across a broad temporal span ranging from brief concept studies to longer-term trial deployments. We'll restrict comments to the latter since we believe that long-term iterative design is essential. Our multiple year research program with wireless sensor networks in agriculture will be offered as an example. Despite all this work, we will not have discovered how to enable ubiquitous computing until we have gone much further. For example, real, useful deployments will not occur until we have solved issues related to model building and sampling frequencies (to be detailed below).

Simply focusing on long-term iterative design is not enough. Success depends upon research that pushes up against and must conform to the real world phenomena associated with our application. We must, for example, understand the domain of our application. "Good metrics for selection, analysis, and evaluation of ubicomp applications" follow from that understanding. "Good approaches to longer-term iterative design in which applications are refined and scope expanded" require working with and paying attention to the people on whom our emerging technologies are about to emerge.

In some ways, the problem of "what leads to good application led research" is reminiscent of Plato's Meno dialog where Meno asks Aristotle how he can inquire into something he does not know, what should be the subject of inquiry, and how will he know that he's discovered what he set out to learn. Plato goes on to claim that we can do the inquiry if we start with beliefs that we can then evaluate. The point to be made here is that these beliefs should come from good qualitative work in the field and the evaluation should follow from field trials. To that end, I will offer a couple of general points that come from my group's application-led research in wine grape-growing.

What should be the subject of the inquiry?

Metrics for selection. Our research began before we selected the subject of our inquiry. We knew we wanted to look at sensor networks. We looked for a domain where a sensor network would make sense. We chose agriculture because it seemed a likely area of early adoption. In agriculture, sensors are common and sensible data are recognized as central to the enterprise. Focusing on the domain, allowed us to constrain the technological solutions that we might entertain, but understanding the domain was paramount for doing this well.

How do we inquire into something we do not know?

Ethnographic Methods. We used ethnographic methods to determine the kinds of deployments that would be attractive to our potential users. We began with the simplest methods. We spoke with the people working in this area. We began with semi-structured interviews of various people in the wine production value chain. We spoke to vineyard owners, vineyard managers, wine makers and their assistants, wine marketers, wholesalers, and retailers. These interviews addressed not only day-to-day activities but also the economics of their end of the business. This allowed us to focus on areas where there might be potential ROI from a deployment. On the basis of the interviews we identified a large number of potential applications ranging from tracking the work done in the vineyard (e.g., spraying or leaf-stripping) to monitoring conditions of the finished product as it was shipped. In each of these applications we were able to establish the parameters that would be needed for analysis and evaluation of the performance of a deployed system. In addition to this, we were also able to consider the form factors that we could easily deploy for a "deep dive" and selected one sub-domain for our trial deployment. We selected the agricultural side because this was one area that we thought would be shared with many other potential applications.

Participant Observation. Despite our decision to look at the agricultural side, we still needed to know just what we would do. We began participant observation, where we worked alongside those who would be most directly in contact with the technologies that we had in mind. We were involved in various aspects of fieldwork before harvest, harvest, crush, and cellar maintenance. Following this phase of the research, we decided to put sensors in the field to monitor climate and, we hoped, improve fruit quality and/or control over fruit quality for the winemaker. This was an application that was of interest to people throughout the value chain and we hoped would transfer easily to other crops.

Throughout these early phases of the research, it became increasingly clear to us that the kind of data that we could use a sensor network to collect had never been available to these practitioners before. This lack of history with such data had an interesting consequence. Some were interested to see what it could do for them, others doubted that it could add much – the research was just not in. Now we saw that we had to do the research to prove the value of the data.

Trial deployment as participant observation. Normally, a trial deployment would not be considered participant observation. Who would we be observing but ourselves? However, in this case, we needed to work with domain scientists so that we could establish the value of the data. So, as we came closer to understanding what we could do, we started working more directly with domain scientists. In fact, we sought out working scientists (notably, a PhD in plant physiology with a focus on wine grapes) to further refine the application area and make our data useful.

This phase of the research was in some ways the most exciting. We really didn't know what would work and only hoped that our bet would pay off. In the end, we found that we were able to predict a significant amount of the variability of the fruit in the vineyard we monitored. We were able to predict variation in pH, titratable acids, and berry weight. We found that we could predict boundaries for frost damage (and measured parameters for some kinds of damage). We were also able to define areas that would be amenable to growing more valuable crops. We could have done none of this without the input of a scientist working in the field.

We are still working on ways to make these data useful by developing different visualization that allow people with different interests to consume the data in a form they find useful. Still, there's much more to do.

How do we know we've discovered what we set out to learn?

Model Building. In the course of our research, we haven't discovered all that we need to do to enable ubiquitous computing. We didn't learn the answer but we did learn how to get closer to it. We have been able to determine some very specific lacks in the research that we've done. One of the most interesting aspects of working with domain scientists was that we finally really understood what it meant that no one had ever collected this kind of data before. The working scientists finally drove home the notion by explaining that it wasn't clear that finer grained data would be relevant because the existing models of how plants respond to climate used a very loose measurement for climate. Climate was presumed to be homogeneous and could be measured from one point in space, that is, one measurement with no variation. Practitioners and scientists were used to single measures with no variation. Our sensor network offered a measure of the variation in climate with variation.

Consider the way in which extant models have been developed. In order to represent the population of plants in a test plot, agricultural scientists have worked hard to ensure that they have a broad sample of plant products. They use various methods of plant sampling ranging from random to stratified. This ensures that they have the full breadth of variation with which to characterize the crop. On the climate side (where we were introducing technology), scientists had always relied on a small number of measurements to characterize the area the plant products were sampled from. In fact, they usually relied on just one measurement. That is, they had variation in the plants but not the climate. This meant that models predicted a broad range of plant variation that could be expected in a particular climatic situation. These models would hold that our variance in climate could not predict variation in compositional chemistry because the variance in the plants is so high. (Of course, if we believed those models, we would not have gone on with this research.) Which is to say, the models to really support this work will, in many cases, be non-existent. The work that we did started to address the issue of model development but did not go far enough. Two years of data are the norm in agricultural research before findings are even considered publishable. What does this kind of timing mean for the model development that would have to be completed for ON World’s seven billion dollar market to be a reality in 2010? We have to develop those models and there isn’t much time.

Following from this dearth of models we have a related problem. How finely grained shall we sample our variables?

Nyquist Frequencies. Optimal sampling rates are not well understood. At which spatial frequency should we sample? At which temporal frequency should we sample? The simplest way of describing this is that we do not know the spatial or temporal Nyquist frequencies for the domains in which we want to see deployments. This may be more significant than appears at first blush. In our experience, we have found that different services require different densities of measurement. However, you can’t know what the optimum density for any service might be without over-sampling and then looking to see how sparse your sampling could have been while at the same time still reflecting the appropriate level of variance.

Data for each of these issues can be presented at the workshop.

Summary

When we need to understand our application domain enough that we can develop metrics for selection, analysis, and evaluation of ubicomp applications, we believe that ethnographic methods can be reasonable tools. Having set these metrics we can hope to understand when we have learned what we have set out to learn. However, we will add at this juncture that (at least in the case of wireless sensor networks) we will not be done before we have determined exactly how dense our infrastructure needs to be or have built models to support the analysis of the data we have collected.

Emulating the Future with/of Pervasive Computing Research and Development

Abstract

This position paper presents a vision of Pervasive Computing as a complex and user-centric research and development object. Emphasis is put on an "augmented emulation" as a core toolkit for development and assessment of future work and we will place our proposal in the frame of some tools and work that we have undergone. We believe that developing a use-case/scenario methodology on top of this toolkit will allow evaluation of the developed technologies by using a combination of real products, prototype softwares, emulated softwares, hardware and environment in interaction with real users or agents. Those scenarios are ways of describing application needs. The refinement of elements of those large-scale real-time augmented simulations should rely on standard metric (domain specific) and user-related metrics, and will lead to more refined scenarios in a global process.

Keywords

Pervasive/Ubiquitous Computing, Ambient Intelligence, large scale emulation, augmented simulation, model, standard, taskforce

Author

Laurent Ciarletta (laurent.ciarletta@loria.fr)
+33 -3-83-59-20-11
Madynes Research Team,
LORIA,
Technopole de Nancy-Brabois - Campus scientifique
615, rue de Jardin Botanique - B.P. 101
54600 Villers-Les-Nancy Cedex
France

1. Introduction

Pervasive or Ubiquitous Computing can be seen as the point of convergence of four classical Computer Science areas (cf. figure 1): Networking (connecting the elements, accessing data), Embedded Computing (constantly improving software and hardware miniaturization and autonomy), Personal Computing (providing services) and Computer-Human Interaction (with Artificial Intelligence providing the needed context-awareness and automatic customization).

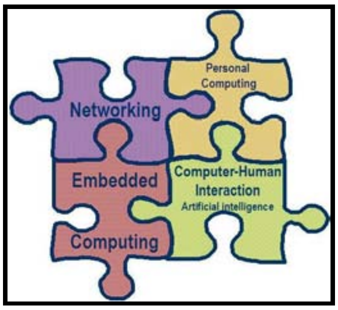

Figure 1: Pervasive Computing = convergence of four classical Computer Science areas

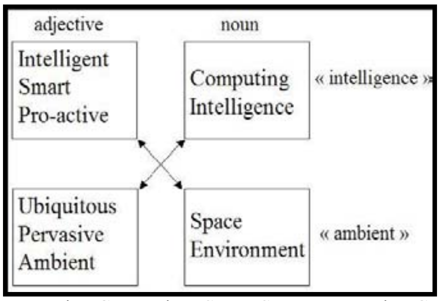

Figure 2: Pervasive Computing, Smart Space, Pro-active Computing, Ambient Intelligence etc.

Pervasive Computing is the idea that computing power ("intelligence") and connectivity will be available everywhere and at anytime ("ambient") and help users in a transparent way in their everyday-life. The association of these two concepts can be seen in the (too?) many expressions that are used to designate it (cf. figure 2). This prospect that computers will be everywhere and give the environment a lot more (computing power) intelligence and interactivity with human users has generated a large amount of interest in the last years as a source of inspiration in Computer Science research and can be considered as the Eldorado where researchers can experiment their wildest ideas.

The paper is organized as follow: limits of real test-beds are evocated to motivate the interest of emulation as a key underlying engine and fuel for Pervasive Computing research. Next, we analyze existing usage scenarios of Pervasive Computing and emphasize the need for making this technology tangible for everyone. Then we suggest building on the classical models and standards to propose a framework that will bring together the pieces of Pervasive Computing and its users to better evaluate pervasive applications.

2. Emulation augments reality

The Limits of Real Test-beds

Research projects, both academics and industrials, have built real test-beds (especially in the US) from a room scale (SmartSpace at NIST [1], Microsoft Easy Living [2]) up to a residential condominium (MIT Changing Places/House_n project [3] or Fraunhofer inHaus [4]). While such environments offer impressive demonstration capabilities, they present important shortcomings:

- Obviously, they are expensive and difficult to set up (hardware cost, need for a handyman…), an investment that most research teams can't afford.
- As a consequence of the previous point, the scale of real test-beds is limited.
- They fall obsolete quickly (sometimes even before they are finished).
- They can not be (easily) moved from one place to another. That means that only the "locals" can experiment on it. It minimizes the chances for collaboration with other research teams around the world.
- It is difficult to replay the exact same scenario over and over into them because many parameters vary (the speed at which a person moves into the corridor, the day light, changes in the test-bed …).
- They are usually tied to a particular usage and it's arduous to adapt them for another (for instance: a smart kitchen environment can't be converted into a smart class room overnight!).
- They are not well-suited for specific applications such as those related to handicapped, seniors or children mainly for safety reason.

Generalized Emulation

Definitions of "to emulate":

1. Effort or ambition to equal or surpass another
2. To strive to equal/match or excel, especially through imitation
3. (Computer Science) To imitate the function of another system (not necessary at the same speed)

Firstly, the fact that various communities are involved is an opportunity to "emulate" each other, to blend the borders that have been built and to work on common objectives and grounds. There is here a need for common vocabulary,

testing technologies, standard interfacing, interoperability and interaction solutions. Those are the foundations for an open, strong and dynamic Pervasive Computing community.
Secondly, since Pervasive Computing applies to complex systems (from low-level technologies to user interactions), in order to specify good applications, it would be interesting to completely emulate those systems creating fake worlds where the specific piece being developed can be embedded, tested, compared with other solutions and demonstrated in its context, even though some of the technologies have not been developed yet, or are available as prototypes on a small scale. We are voluntarily using generalized emulation instead of augmented (with real world measures) simulations since we have the prospect of gradually interacting with the real-world and real-users in real-time.
For instance, imagine you develop an innovative service, such as an urban emergency system. This system needs pervasive high-speed network connectivity and context-aware support, since its goal is to inform the closest medical and police forces available in the case of an accident. A complete scenario with an emulated environment (physical layout and properties of objects), and users (using intelligent agents for example) as well as devices and sensors emulators would obviously allow focusing on the development of the application and related technologies, and to experiment several solutions. But perhaps more importantly, it would allow others to provide and test the missing bricks (for example future devices, sensors, expert systems), the future environment (for example an architectural project for a given neighborhood) and to demonstrate “how it would and should work” (as far as needed/critical infrastructures, technologies and performances are concerned).
Our inspiration comes from looking at architecture, where entire neighborhoods are simulated, from the mechanical point of view to the environmental impact, or engineering, where behavior of cars and ergonomic are simulated, or from some fields of Computer Science like networking. How can we bridge all of these?
Some of the bricks are already available today, like:

- hardware emulators for PC-like devices (PocketPC/Palm emulators [5], VMware [6], User Mode Linux [7], Bochs [8], to name a few),
- electronic circuitry design toolkit,
- network simulators/emulators (NS2 [9], NCTuns [10], NISTnet [11], etc.),
- low-level wireless signal propagation simulators (WiSe [12]),
- Virtual Reality toolkits [13] and game engines for simulating the environment,
- Tangible User Interface [14] toolkit for prototyping physical interactivity,
- Intelligent agents [15] that can be used to emulate users in well defined constrained contexts.

What’s missing is some common platform where different emulators, real hardware and applications would work together to build complex simulations. Depending on the scale of the distributed nodes and of their performances, bridging the different emulators can go from loosely coupled to constrained real-time architectures.
The need for a common platform starts with some development and testing tools and will naturally lead to (and nourish at one time) the question of modeling, measurement (metrics) and standardization.

3. Putting the user in the center

It’s particularly important for Pervasive Computing to evaluate the perception its users have of it. Looking for “successful” applications and products of Pervasive Computing, and surveying which ones are making titles in magazines and TV shows or are really being commercially offered, one founds: smart fridges [16], connected multicolor “mood” lamps [17], PDAs and smartphones, robotic mowers or vacuum cleaners [18], robotic pets and friends, PAC-MAN in real city [19], simple location-aware wireless phone services and recently RFIDs. Lots of them look like gadgets for nerds, or really expensive toys. So a basic analysis of these applications may show that killer application is all about entertainment, and is it difficult to see the impact on human well-being.
Surely, Pervasive Computing is mainly confined to entertainment because remaining issues (such as security) prevents it from entering a critical field (such as medical), but also maybe because it can not be convincingly demonstrated to “serious users”: one of the pitfalls of promising technologies is that the first targeted/potential users are actual researchers expressing what they would like to see in the future. To other people, this often looks a lot like a Sci-Fi writer having the opportunity to implement his own ideas, but it’s not tangible.
A solid set of modeling and experimentation tools can create a situation where this creativity can express its potential, and where other users (researchers, customers, or sponsors) may experiment with it progressively and bring their input in the process. It can be used at the same time as an educational and demonstration (advertisement) tool. This could also provide understanding and therefore confidence in the technology. Furthermore, all the classical well known objective metrics continue to be available with such tools, and more subjective metrics such as usability and quality feedback based on users “feelings” could be added.

4. A Generic High Level Model for the Pervasive Puzzle

When browsing Pervasive Computing research, subjects range from RFID technology to Graphical Interface Ergonomic passing by agent technologies. It may be difficult at times to see how all of these fit into the global scheme.

Following a natural tendency, almost every field of Computer Science, every community develops its own vision of Ubiquitous Computing, and is somehow envisioning further research through the prism of its domain.
A generic high level model could be used and extended to provide a better understanding of the relative positioning of the domains and where their interactions take place. Tools similar to the HLA (High Level Architecture [20]) used in military simulations, or its lighter versions like the MSI (Multi Simulation Interface [21] could provide this federative capabilities and could help organize distributed emulation by logically placing elements of the simulations relatively to each other.

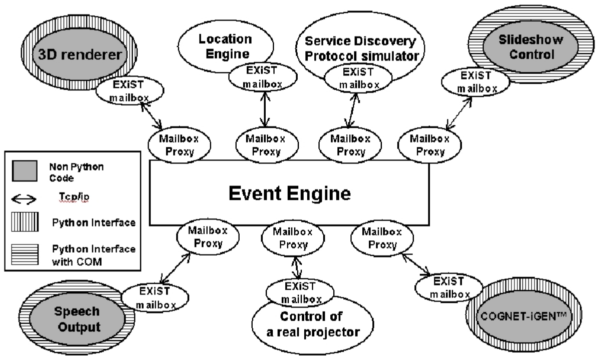

Figure 3: EXiST in action, simulating a smart conference room

In [22] we propose a proof of concept of such a generic toolkit: to demonstrate a smart conference room, EXiST (EXperimental Simulation Tool) combines a real wireless projector (Aroma smart prototype projector using Jini software [23]), intelligent agents to represent users, OpenGL modules (using Blender) to describe and visualize the physical layout of the room and other modules as depicted in figure 3. EXiST is conceptually tied to the LPC (Layered Pervasive Computing) model [24] which generalizes layered models such as the ISO-OSI for Networking taking into account the environment at the bottom part as well as the user goals and application design purpose in the highest layer, but leaving the lower granularity to specialized fields, where specific models remains more pertinent.

5. Synthesis and propositions

The variety of Computer Science fields and technologies involved in Pervasive Computing makes its richness and complexity. Lots of researches are conducted in parallel, and the results of one may be needed by others, which often limits the field of possible applications and the speed of iterations. When considering that Pervasive Computing is User-centric, new evaluation techniques (combining exisiting ones, using emulation) and metrics (such as usability) have to be considered. But researchers should not see themselves as standard users since it can hinder a broader impact of their research seing as hype (or worst). Applications are the visible tip of the iceberg, and the challenges they offer can uncover main issues and lead the underlying tehchnologies development.
We propose to follow two parallel paths to allow the emergence of visible (demonstrable) and evolved application-oriented research and development:

- use large scale (both vertically and horizontally[1]) emulations/simulations based on scenarios to test, demonstrate and emulate creativity for applications. A complete toolkit can be defined reusing lots of already existing tools. The bridging can be done using loosely-coupled to real-time distributed simulation techniques. This kind of tool naturally fosters incremental development cycles where applications or products can be integrated and tested in different versions ranging from emulated ones, to prototypes, and finally to real software and hardware, while waiting for the availability of underlying technologies. It also gives tools and methods to measure performances and to insure safety and usability.
- develop a simple model/classification to place every work and concept where it belongs and good practices to integrate them within the emulation toolkit. It may improve interactions between the different communities and foster collaboration. In the same areas, related-metrics could be better positioned and their impact evaluated.

Our experience on such topics shows that it is not at first an easy path, but that is it promising. What works for individual areas of research and for architecture or engineering should prove highly potent when dealing with the combination of them. Designing and developing an efficient distributed emulation layer and providing methods to optimize communications and relations between elements of the simulations are the core technical issues but we can probably imagine the public impact of a scenario where somehow the Sims are playing in a Doom3 world (with its PDAs) over an ns2 emulated network against human users?
We are therefore proposing the deployment of a large scale community test-bed where everyone could participate and interact with others' researches in a real-time fashion, a PCBone (Pervasive Computing Bone). This could be piloted by a taskforce, that would be in charge of defining the tools (Open Source whenever possible) and standards for emulating the future with/for Pervasive Computing technologies and contributing to monitor and inform researchers and potential users on progress made by the community.

[1] Vertically meaning across abstraction layers and Computer Science fields, and horizontally across large number of possible simulated objects

6. References

[1] Smart Space: http://www.nist.gov/smartspace/
[2] Easy Livinghttp://research.microsoft.com/easyliving/
[3] Changing places: http://architecture.mit.edu/house_n/index.html
[4] inHaus: http://www.inhaus-duisburg.de/en/
[5] Palm Os emulator: http://www.palmos.com/dev/tools/emulator/
[6] VMware: http://www.vmware.com/
[7] User Mode Linux: http://user-mode-linux.sourceforge.net/
[8] Bochs: http://bochs.sourceforge.net/
[9] NS2: http://www.isi.edu/nsnam/ns/
[10] NCTuns: http://nsl10.csie.nctu.edu.tw/
[11] NistNet: http://www-x.antd.nist.gov/nistnet/
[12] WiSe: http://www.bell-labs.com/org/wireless/wisext.html
[13] Diverse, An Open Source Virtual Reality Toolkit: http://diverse.sourceforge.net/
[14] Tangible User Interface Toolkit:
http://dmg.caup.washington.edu/xmlSiteEngine/browsers/stylin/project51.html
[15] iGen: http://www.cognitiveagent.com
[16] Smart Fridge: http://www.lexicle.com/smartfridge
[17] The Dal lamp: http://www.violet.net/dal.jsp
[18] Roomba: http://www.irobot.com/consumer/
[19] Human PACMAN: http://mixedreality.nus.edu.sg/research-HP-infor.htm
[20] HLA: https://www.dmso.mil/public/transition/hla/
[21] MSI: http://msi.sourceforge.net/
[22] *Using Intelligent Agents to assess Pervasive Computing Technologies*, L. Ciarletta, V. Iordanov and A. Dima, International Conference on Intelligent Agents, Web Technology and Internet Commerce (IAWTIC 2001), 07/2001
[23] *Aroma-Wireless JiniTM Connection Technology-Based Adapters for Smart Space Devices*, A. Dima, K. Mills, L. Ciarletta et al., Technical Session 622 of Java One, San Francisco, 06/2000
[24] A conceptual Model for Pervasive Computing, L. Ciarletta and A.Dima, International Conference on Parallel Processing (ICPP 2000), Workshop on Pervasive Computing, 08/2000

Can Early-Stage Tools and Techniques for Iterative Design Help Researchers Understand a Problem Space?

Scott Davidoff [1], Scott Carter [2], Jennifer Mankoff [1]

[1] Human-Computer Interaction Institute
Carnegie Mellon University
{scott.davidoff,jmankoff}@cs.cmu.edu

[2] EECS Department
UC Berkeley
sacarter@cs.berkeley.edu

Abstract. Researchers developing Ubicomp applications often must make ill-informed but irrevocable decisions early in the design process. While desktop computing researchers have multiple methods at their disposal to manage the risk involved in these decisions, the complexity of Ubicomp research affords few alternatives. We suggest that Ubicomp research faces a poverty of effective design process. We explore alternatives that might supplement existing design processes so that designers can make decisions from positions of information. This suggests an opportunity to develop both tools and techniques that support early-stage evaluation.

1 Introduction

Fourteen years after Weiser's vision [18], Ubicomp research has made modest progress towards achieving that end. Now is an appropriate time to reflect on why.

In this paper, we suggest one reason: when developing applications, *researchers cannot evaluate the problem space*. Ubicomp applications are bound by an interesting set of circumstances. Because applications must address multiple, unpredictable contexts of use, researchers need to evaluate them in the field. But developing applications that are field-deployable often involves sufficient cost that researchers are forced to arbitrarily limit the design space early in the design process. Researchers cannot evaluate a problem without first building a technology. But deployment requires knowledge that researchers do not have.

Assuming no "silver bullets" [1] arrive on the scene, what can Ubicomp researchers do? We suggest that new tools and techniques that support early-stage evaluation might help researchers make better decisions, early enough to have impact.

In this paper, we describe knowledge gaps in the design process of one Ubicomp research project we worked on. We then survey evaluation techniques inherited from the desktop world, and describe their shortcomings for Ubicomp. Lastly, we describe the research opportunities afforded by these shortcomings.

2 Case Study

Since considerable barriers of access, skill and intimidation keep elders offline [10], we hypothesized that a device that hides email services within familiar objects might help elders get the benefits of email, without asking them to absorb high learning costs. We quickly identified two candidate technologies to deliver on this promise: an augmented telephone, and a paper-to-email bridge. But how would we know which system would produce superior results?

Traditional exploratory methods failed to meet our needs. Laboratory evaluation – even "future scenario" [2] games of imagination – could never simulate the highly contextual factors central to our investigation. To really know which system would provide superior results would mean developing and evaluating two functioning systems. But to do so would require significant expense.

Unable to commit such substantial resources, we opted to interview elders. After examining elders' communication process and technological comfort zone, we choose letter-writing. We ultimately developed ElderMail [9], a tangible email system that uses a book as a user interface, or BUI. But we cannot say in earnest that our decision produced optimal results. Lacking a low-cost way to evaluate vastly different competing technical alternatives, we had no way to accurately predict the relative efficacy of one solution over another.

3 Early-Stage Evaluation, Evaluated

Evaluation is traditionally characterized as *formative* – the up-front exploration of a problem space – or *summative* – the retrospective measurement of system impact. But since Boehm's spiral model [3] refocused software development, researchers often perform what we describe as *iterative evaluation* – a more rapid and repetitive design-build-test cycle. And though we have inherited multiple iterative evaluation techniques from the desktop world, few meet the diverse needs of early-stage Ubicomp research.

Software toolkits and interface builders evolved to substantially meet the needs of desktop research. And we can already find toolkits designed for Ubicomp, e.g. [12]. But while these toolkits lower the cost of creating field-deployable technologies, they still do not support the kind of low-cost, high-level decision-making early-stage research requires.

Another strategy suggests replacing the user with a computer models that simulate human input, e.g. [6]. But Ubicomp systems exist in a variety of environments that are currently too complex and dynamic to accurately model with the predictive power required to enable design decisions.

Other researchers, e.g. [1], turn to social science theory for guidance. But social science theory is currently too limited to provide accurate predictions for complex behaviors in situations involving multiple, and still largely-unknown variables and effects.

Paper prototyping [16] has evolved as an extremely low-cost system proxy. Because paper is such a naturally flexible medium [17], it shows distinct promise for prototyping smaller Ubicomp systems. Some researchers are exploring how paper can be applied to more complex Ubicomp scenarios, e.g. [7]. But paper currently falls short when it meets the dynamic needs of many Ubicomp systems. It cannot scale well to distributed, multi-user applications, or situations that require vast or dynamic input from multiple channels.

Some researchers have explored simulation as a means to explore a complex problem space. We more commonly find these so-called Wizard of Oz (WOz) studies in speech and multimodal interface, e.g. [15], or intelligent user interfaces e.g. [8]. Despite their simulated components, WOz studies often involve substantial programming investments. Researchers still need to create largely working systems, and replace particular components with methods for experimenters to simulate machine input or output.

Some researchers are addressing this particular issue by providing tools to help researchers build WOz simulations for particular domains, such as location-aware applications [13], or speech [11]. Such tools, though helpful, are limited to a single domain. This forces researchers evaluating applications in multiple domains to re-implement simulations to suit the semantics of each prototyping toolkit.

4 Conclusion

To understand how burgeoning Ubicomp applications function in context, researchers will have to find lower-cost ways to evaluate emerging design alternatives. We see an opportunity to develop both tools and techniques to better support this process.

Paper has proven it can capably simulate the user interface. But what techniques are available when researchers want to simulate an environment? What techniques can help researchers design representative tasks, and select appropriate performance metrics for poorly-understood domains? And what techniques are available to compare, analyze, and visualize the complex behavioral variables produced through such evaluations?

When tools are appropriate, what tools might provide the structure to design, evaluate, and analyze multiple design iterations. We suggest that a WOz toolkit might fill this role. A WOz toolkit would have to generalize WOz patterns across multiple application domains. It would also have to provide low-cost ways to integrate multiple input streams, both real and simulated. and support the wizard's complex real-time performance needs during testing. It might also provide special tools to visualize, analyze and explore data from complex and multiple semantics.

Ultimately, any tools and techniques developed would seek to lower the development costs so that researchers can make fewer assumptions about the new and unpredictable contexts of use for which we are only just now beginning to explore, and develop applications that can meet the real and demonstrated needs of users, as observed in their environment.

References

1. Barley, S.R., Dutton, W.H., Kiesler, S., Resnick, P., Kraut, R.E., Yates, J. (2004) Does CSCW Need Organization Theory?, in *Proceedings of CSCW'04*, 122-124.
2. Beyer, H., & Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufman Publishers.
3. Boehm, B. W. (1988) A Spiral Model of Software Development and Enhancement, in *IEEE Computer*, 21(5): 61 - 72.
4. Brooks, F. P. (1995) The Mythical Man-Month: After 20 Years, in *IEEE Software*, 12(5): 57-60.
5. Buchanan, R. (1995) Wicked Problems in Design Thinking. In V. Margolin and R. Buchanan (Eds.) *The Idea of Design*, MIT Press.
6. Card, S. K., Moran, T.P. & Newell, A. (1983) *The Psychology of Human-Computer Interaction*, Erlbaum.
7. Chandler, C. D., Lo, G., & Sinha, A. K. (2002) Multimodal Theater: Extending Low Fidelity Prototyping to Multimodal Applications, in *Proceedings of CHI'02*, 874-875
8. Dahlbäck, N., Jönsson, A., Ahrenberg, L. (1993) Wizard of Oz Studies: Why and How, in *Proceedings of IUI'03*, 193-200.
9. Davidoff, S., Bloomberg, C., Li, I. A. R., Mankoff, J., & Fussell, S. R., (2005) The Book as User Interface: Lowering the Entry Cost to Email for Elders, in *Proceedings of CHI'05*, to appear.
10. Fox, S. (2004) Older Americans and the Internet, *Pew Internet & American Life Project*, http://www.pewinternet.org/pdfs/PIP_Seniors_Online_2004.pdf
11. Klemmer, S. R., Sinha, A. K., Chen, J., Landay, J. A., Aboobaker, N., Wang, A. (2000) Suede: a Wizard of Oz Prototyping Tool for Speech User Interfaces, in *Proceedings of UIST'00*, 1-10.
12. Klemmer, S., Li, J., Lin, J., & Landay, J. (2004) Papier-Mache: Toolkit Support for Tangible Input, in *Proceedings of CHI'04*, 399-406.
13. Li, Y., Hong, J. I., & Landay, J. A. (2004) Topiary: A Tool for Prototyping Location-Enhanced Applications, in Proceedings of UIST'04, 217 – 226.
14. Luff, P., & Heath, C. (1998) Mobility in Collaboration, in *Proceedings of CSCW'04*, 305-314.
15. Oviatt, S., Cohen, P., Fong, M., & Frank, M. (1992) A Rapid Semi-Automatic Simulation Technique for Investigating Interactive Speech and Handwriting, in *Proceedings of Spoken Language Processing'92*, 1351-1354.
16. Rettig. M. (1994) Practical Programmer: Prototyping for Tiny Fingers, *Communications of the ACM*, 37(4): 21-27.
17. Sellen, A. J., Harper, R. H. R. (2003) *The Myth of the Paperless Office*, MIT Press.
18. Weiser, M. (1991) The Computer for the Twenty-First Century, In Scientific American, 3(3): 94-104.

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 Confer-ence on Mobile Computing (Mobicom). 1996.

[6] Weiser M. "The computer for the 21st century" Scientific American, Sep 1991, pp. 94-

A Framework for Mobile, Context-Aware Trails-based Applications: Experiences with an Application-led Approach

Cormac Driver* Eamonn Linehan† Siobhán Clarke Andrew Jackson Shiu Lun Tsang* Mike Spence†

Distributed Systems Group,
Computer Science Department
Trinity College Dublin,
Ireland.

{FirstName.LastName}@cs.tcd.ie

Abstract. The *Hermes* project is addressing the development of a generic framework to support the design and implementation of mobile, context-aware applications. Our initial focus is on trails-based applications. This paper discusses our experiences with an application-led approach to framework development and presents the issues encountered to date. Remedies for the issues are proposed as a step towards enhancing the benefits of application-led ubiquitous computing research.

1 Introduction

The development of ubiquitous computing applications poses numerous challenges to software developers. Issues inherent to the ubiquitous computing paradigm must be tackled during each application development effort, meaning that developers repeatedly encounter the same or similar issues, regardless of the application under consideration. These issues range from low-level programming issues to high-level usability issues.

Hermes [5] (http://hermes.dsg.cs.tcd.ie) is a software framework for mobile, context-aware trails-based applications which will support developers by providing generic components containing structure and behaviour common to all trails-based [4] applications. Mobile, context-aware applications are those that run on devices such as PDAs and mobile phones, and have an awareness of the physical and social situation in which they are deployed. A trail can be thought of as a collection of locations, together with associated information and activities, and a dynamically reconfigurable recommended visiting order. Trails underpin a wide range of useful applications for a mobile user who has a set of activities that may or should be carried out throughout the day at different locations. Combining the trails concept with mobile, context-aware technology creates opportunities for innovative activity-based application development. Examples of trails applications that are both mobile and context-aware include courier management systems, basic route planners, treasure hunt games and student support systems.

The Hermes framework will facilitate the development of a diverse range of realistic trails applications. This research is relevant to the field of ubiquitous applications development as a whole because we consider trails applications to be archetypal mobile ubiquitous applications which exhibit the ubiquitous computing characteristics described in [2].

This paper presents our experiences to date with following an application-led approach to framework development. The remainder of the paper is as follows. Section 2 contains a description of our framework development approach and work to date. Section 3 describes the issues we encountered. Section 4 suggests remedies to these issues. Section 5 contains a summary.

2 Framework Development Approach

The Hermes framework is being developed using the "Three Examples" technique for framework evolution described by Roberts et al [14]. We began the process by specifying requirements for three mobile, context-aware trails-based applications. These are listed below in order of increasing complexity as regards ubiquitous computing issues addressed:

1) Student support system for new students at Trinity College Dublin.
2) Trails-based mobile treasure hunt game.
3) Delivery courier support system.

We have implemented a student support system called Oisín which provides campus-wide trails including both compulsory tasks such as registration and optional activities such as visiting college buildings. The trails are based on and affected by environmental and personal context and are automatically dynamically reconfigurable. We have recently initiated a period of application testing involving a user trial preceded and followed by written questionnaires and interviews.

3 Issues and Experiences

This section contains the issues encountered by the Hermes development team during the design, implementation and initial user testing of Oisín.

3.1 Lack of Interdisciplinary Expertise

Lack of Psychological Expertise

As the user trial is based on a questionnaire design, the data produced depends entirely on the quality of the questionnaire employed. Our experience illustrates that computer scientists are typically not the people best qualified to conduct user trials. For example, some of our questions proved confusing to the test subjects, even though those tested thus far have a computer science background. These questions yielded no data,

* Supported by the Irish Research Council for Science, Engineering and Technology (IRCSET) Embark Initiative
† Supported by Intel Corporation

leaving our data sets incomplete as a result of our own short-comings in the area of questionnaire design.

Lack of Graphical User Interface (GUI) Expertise

Poor GUI design has prevented the collection of data on application features we are interested in. An objective of our trial is to gauge user opinion on Oisín's dynamic trail reconfiguration behaviour. We wish to ascertain whether or not users feel the decisions made by the application on their behalf are appropriate and timely. However, we found that the implementation of the trail adaptation behaviour was deficient from a GUI perspective. This resulted in cases where the users did not notice that the trail had been reconfigured, due to the lack of appropriate notification. It is apparent to us that although the dynamic reconfiguration decisions may make sense, the user can be turned off the concept if the GUI is not designed in a manner which maintains the user's feeling of control.

3.2 Social Computing

The social attraction of the application was overlooked when developing Oisín, most notably in terms of the hardware form factor and the social acceptance of the devices used. Users of Oisín are required to carry a GPS receiver and a PDA, which are connected by a cable. The GPS receiver requires line of sight to satellites so must remain in the user's hand. The PDA requires two hands to operate. This leads to a very difficult to manage deployment platform.

We also found that repeated consultation with a PDA attracted inquisitive and sometimes disparaging glances from passers-by. We accept that levels of social acceptance of such devices vary from country to country but believe our experience to be representative of the general case in our environment. This factor coupled with the difficult to manage form factor has lead to a negative social impact on system users, resulting in negative comments in our user trials.

3.3 Implementation issues

Implementation issues arose that detracted from getting unbiased results on potentially more important aspects of the research. In our experience, two main implementation issues affected our results: 1) limited resolution and unpredictable error of our chosen location system and 2) resource limitations of mobile devices.

Oisín uses a GPS receiver to obtain location data. During user trials the location data was only accurate to twenty meters, resulting in user location being inaccurately displayed on the map. This confused users and caused the application to make incorrect reconfiguration decisions. This inaccuracy is attributed to the urban environment in which Oisín operates, where GPS functionality can be lost because buildings occlude signals or scatter them in multipath reflections.

The conflict between the limited processing power of the mobile device and the calculations required by the dynamic reconfiguration algorithm caused a delay each time a trail reconfiguration decision was made. Delays in the order of 30 seconds were experienced while the application updated the user's trail. During this time the application was "busy" and could not be interacted with, leading to user frustration.

3.4 Application Scenario Choice

A motivating factor in our decision to implement a student support system as the first prototype application was ready access to a pool of test subjects. We intended to use two classes of students on their first day at college. However, due to delays in equipment acquisition it was not possible to meet this deadline, meaning that another application scenario had to be defined. We sent subjects on an information gathering trail around Trinity College's historical artefact displays and art galleries. We observed that subjects exhibited a lack of motivation when following the trail as doing so is not necessarily in their interest. They were purely participating in a user trial, unlike the first day students that would have followed a different trail involving activities such as registration and compulsory lectures.

4 Ideas for Remedies

This section puts forward possible solutions which may overcome the difficulties we have encountered.

4.1 Interdisciplinary Involvement

Psychology

There is an area of psychological research known as psychometrics [12] that is concerned with the design of tests and questionnaires and the compilation and evaluation of their results. Psychometric knowledge is required in order to validate that a user trial is suitable for gathering the type of data required by a development team. It is asserted that "metrics for designing and evaluating pervasive systems are still lacking" [7]. Our approach of gathering user trial results via questionnaires is a common one and increasing the quality of such questionnaires should be a goal of researchers in this area. In parallel with the development of applications we foresee a need for the development of a standardised questionnaire for measuring user-perceived quality of context-aware mobile applications. This could provide data for an evaluation of users' experiences under various conditions and across a range of independently developed applications. Such evaluation techniques can only be developed in conjunction with experts in the area of psychology. For this reason we recommend the inclusion of such experts in the development of ubiquitous computing applications.

GUI

"A frustrating interaction with a computer system can also leave a user feeling negatively disposed toward the system and its makers." [9]. This finding suggests that when evaluating an application via a user trial, the ease of use of the application's interface can colour the results of questionnaires.

It is essential that experts in the field of human-computer interaction are consulted on design issues where possible at an early stage of development. For example, when we consulted Human Computer Interaction (HCI) researchers they were able to advise us on how to reduce the feeling of lack of control regarding trail adaptation by making a small number of improvements to the user interface.

4.2 Social Computing

Device satisfaction and social attraction need to be considered when selecting hardware. Neither of these factors featured in our hardware requirements. We are preparing new hardware requirements for the next prototype application featuring a subset of factors from [10], which we believe have the most significant social computing impact for our class of application:

1) **Familiarity**. Is the form of the device one that is familiar and appropriate for the context of its use?
2) **Appeal**. Is the device something that the user is comfortable being seen using?
3) **Disruption**. Does the device disrupt individuals' natural social behaviours?
4) **Pervasiveness**. Is the device mobile or otherwise convenient to use in social settings?

The above considerations have led us to begin evaluating and researching the possibility of basing future applications on mobile phones. It should also be an aim of context-aware applications to produce more socially intelligent applications that adapt not just to physical context such as location and time but to a users social environment providing for a more positive social impact.

4.3 Overcoming Implementation Issues

The main implementation issue hindering our user trial is an inaccurate positioning system. Other researchers have also encountered this issue [1]. The failure of systems beyond our control to supply accurate contextual information highlights a general problem with context-aware mobile application evaluation. Because context-awareness clearly distinguishes ubiquitous computing applications from other mobile applications, the ability to evaluate how applications react to context changes is critical in a test environment. To make this possible a test environment in which developers have some degree of control over available context for the purpose of logging and reasoning about applications correctness and for the purpose of enabling repeatable experiments is required.

A possible approach to achieving such control is to use a simulated environment. Similar efforts in the area of ubiquitous computing are now emerging with immersive 3D games engines being used to simulate the physical environment [3,6,11]. These test environments are designed to simulate different aspects of ubiquitous computing environments. The GeoNotes project [6] modified a games engine to output a player's position via the Context ToolKit. UbiWise [3] modelled mobile devices in the environment and Morla et al. integrated network simulation into a virtual environment [11].

Trinity College Dublin's Knowledge and Data Engineering Group are working towards a ubiquitous computing simulator combining aspects of all these previous works called TATUS [13]. TATUS allows for easily configurable interactive 3D virtual environments to be built using a games engine. Context is exported from the virtual environment to the software under test, simulating the software running within the virtual world. An environment like TATUS with the addition of standardised context acquisition interfaces would be useful for the development of context-aware mobile applications. Such an environment could provide a level of control not easily attainable in the real world and would have even more utility when collaborative context-aware applications are considered. A TATUS-like simulator could allow multiple researchers to simultaneously test the interactions between different devices in a ubiquitous computing environment.

4.4 Application Scenario Choice

To avoid having to carefully select trial subjects so that they are motivated for the particular application under consideration, we are turning our attention towards investigating how the properties of the application itself influence user motivation. Opportunities for creating engaging applications exist in the computer gaming field. Games have a motivational appeal that distinguishes them from other forms of computer interaction [8]. Games provide a sense of control, opportunity for strategy and discovery of information. These qualities are desirable in an application that will be evaluated by user trial. For these reasons we are currently working on the next prototype application which is a mobile, context-aware trails-based treasure hunt game.

5 Summary

This paper discusses issues we have encountered to date with using an application-led approach to ubiquitous computing research. We learned that experts from the fields of psychology and HCI should be involved to validate the work done in those areas. We also suggest increased consideration of the social issues surrounding ubiquitous computing. Finally, development of virtual environment-based ubiquitous computing simulators may aid evaluation of prototype applications.

References

[1] Anhalt, J. et al. "*Toward Context-Aware Computing: Experiences and Lessons*". IEEE Intelligent Systems, Wearable AI, 2001.

[2] Banavar, G. et al. "*Challenges in Design and Software Infrastructure for Ubiquitous Computing Applications*". Advances in Computers, Vol. 62, 2004.

[3] Barton, J.J. et al. "*UbiWise: A Ubiquitous Wireless Infrastructure Simulation Environment*". HP Labs, 2002.

[4] Clarke, S., et al. "*Context-Aware Trails*". IEEE Computer, Vol. 37, No. 8, 2004.

[5] Driver, C. et al. "*Hermes: Generic Designs for Mobile, Context-Aware Trails-based Applications*". Workshop on Context Awareness, MobiSys 2004.

[6] Espinoza, F. et al. "*GeoNotes: Social and Navigational Aspects of Location-Based Information Systems*". Ubicomp, 2001.

[7] Grimm, R. "*One.world: Experiences with a Pervasive Computing Architecture*". IEEE Pervasive, July, 2004.

[8] Johnson, C. "*Taking Fun Seriously: Using Cognitive Models to Reason about Interaction with Computer Games*". Personal Technologies, Vol. 3, No. 3, 1999.

[9] Klein, J. et al. "*This Computer Responds to User Frustration: Theory, Design, and Results*". Interacting with Computers, Volume 14, Issue 2, Elsevier, 2002.

[10] Marti, S. "*How does the user interface design of mobile devices influence the social impact of mobile communication?*" MIT Media Lab, February, 2002.

[11] Morla, A. et al. "*Evaluating a Location-Based Application: A Hybrid Test and Simulation Environment*". IEEE Pervasive Computing, 2004.

[12] Murphy, K.R. et al. "*Psychological Testing: Principles and Applications*". Prentice-Hall, 1998.

[13] O'Neill, E. et al. "*A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments*". To appear Proceedings of Tridentcom 2005.

[14] Roberts, D. et al. "*Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frame-works*". Pattern Languages of Program Design 3, 1997.

Design and Evaluation of a Ubiquitous Computing Application for Law Enforcement †

Andrew L. Kun[1], W. Thomas Miller, III[1], Peter A. Heeman[2]

[1] Electrical and Computer Engineering Department, University of New Hampshire
[2] Center for Spoken Language Understanding, Oregon Health & Science University
[1] {andrew.kun, tom.miller}@unh.edu [2] heeman@cslu.ogi.edu

Abstract. We present approaches we have taken in designing a ubiquitous computing application. We propose several metrics that can be used to evaluate this application and other existing and potential applications.

Integrating devices in police cruisers

In his article "The computer for the 21st century", Mark Weiser describes a world where ubiquitous computers blend into the background [1]. Humans rely on information streaming between these computers and they interact with them effortlessly. Our world of course is very different from the world of Weiser's vision – our computers often cannot share data and we often need extensive training to be able to interact with them. Work at the University of New Hampshire has made the inside of a police cruiser look more like the world in Weiser's article [2]. We designed a system (called Project54) that integrates in-car devices into a single, voice-controlled system where all devices can talk to each other, as well as to remote computers, to share data. The system also provides an elegant speech user interface that is easy to learn and easy to use in the hands-busy, eyes-busy environment of the police cruiser. The system is deployed in about 300 police cruisers in the US.

Approaches taken and metrics for assessing results

In creating and deploying the Project54 system, we solved some (and identified other) device interfacing problems and speech user interface problems. We also created a system that serves as an example of how useful ubiquitous computing can be.

Device interfacing

We have integrated electronic devices using the CAN 2B standard for hardware interconnections and the Microsoft COM standard for software modules (these control individual devices). We feel that adopting open standards is what makes affordable

† Work presented here was funded in part by the US Department of Justice under grant 2003CKWX0151 and the National Science Foundation under grant IIS-0326496.

ubiquitous computing possible. Without open standards, ubiquitous computing would rely on proprietary solutions or on intelligent software and hardware that learns how to communicate with other devices. While the latter may become a long term solution, it is not practical or cost effective today. Proprietary solutions limit access to know-how to certain organizations and geographic regions and thus preclude the development of ubiquitous computing systems that span multiple domains and locations.

What standards should one adopt? Standards change and the system may use more than one standard for a certain type of activity. To prepare for modifying or adding standards, the system software and hardware should be modular. The Project54 system is modular. For example, it has a module for communicating with handheld devices that uses the 802.11b standard and another module for communicating with remote servers using the Project 25 digital radio standard.

Four measures of the success of device interfacing efforts are:

– **Do the hardware and software of the system follow open standards?** We adopted and extended existing standards. Our extensions are freely available.
– **How many domains do the interfaced devices cover?** The domain of our system is law enforcement. However, we are working on interfacing devices for home automation, which would add another domain.
– **How many activities within a domain do the interfaced devices cover?** The types of devices (each type covering an activity) we integrated within cruisers are: lights and sirens, radios, radars, GPS devices, barcode scanners, video systems and database software.
– **How many different devices have been enabled for integration into the system?** We enabled over 30 devices for integration into the Project54 system.

Speech user interface

The in-car environment is an eyes-busy, hands-busy environment. While driving, officers often use our system's speech user interface (SUI). A GUI is also available, as well as the original user interfaces of the devices. The SUI uses a press-to-talk button, a directional microphone, a commercial recognizer and text-to-speech engine and a set of grammar files. Grammar files prescribe the form of valid user utterances.

The next generation of SUIs should support multi-threaded dialogues in order to allow concurrent interaction with multiple devices or programs in ubiquitous computing applications. The SUI will need to support interruptions and resumptions of individual spoken interactions. To discover what conventions are natural for people to use, we are running experiments in which pairs of subjects need to complete multiple tasks at the same time, and where the tasks require the two subjects to converse [3]. These studies are inspiring our approaches to developing the new SUI.

Four measures of the success of a SUI implementation are:

– **Is the SUI being used?** We completed a field study of the Project54 SUI and found that officers in the field do use it for certain tasks.
– **What is the SUI recognition rate?** Our field study showed that the average SUI recognition rate is 85%.
– **How much training does using the SUI require?** Officers are trained to use the Project54 system, including our SUI, during one 2-3 hour training session.
– **Does the SUI allow natural speech?** Officers learn set phrases for each application.

The effect of a successful example

For the hundreds of people who use our system, ubiquitous computing is an everyday reality. This success created a pool of sophisticated users who expect electronic devices to perform in a ubiquitous computing environment. In the public safety domain, this has created pressures on industry to adopt standards that will promote ubiquitous computing applications. It also created interest for similar efforts in other domains. For example, our work on police cruisers sparked the interest of firefighters, freight train engine operators and one major US auto manufacturer.

Three measures of how an application can be expected to promote ubiquitous computing research, development and deployment are:

– **How many related domains are there for the application?** Our application domain is law enforcement. Related domains are, in general, domains in which humans interact with multiple devices in hands-busy, eyes-busy environments (e.g. other emergency response applications, human extra-vehicular activities in space, some home automation applications).
– **How many industrial partners are involved in creating the applications?** The Project54 effort has about ten major industrial partners. None of these support the effort financially but collaborate on development and deployment.
– **How many people use the application?** The Project54 system is used by over 500 officers in the field (many of the over 300 deployed cruisers are used by more than one officer) and is being adopted by police agencies throughout the US.

Selecting applications for ubiquitous computing research

We expect that successful ubiquitous computing applications will use open standards and modular software and hardware. If the user is involved in eyes-busy, hands-busy tasks, speech user interfaces will play an important role. Industry participation in the development process is important because of the promise of explosive deployment. Industry participation is more likely if there is existing interest from potential customers and if the application has multiple related domains of use. Starting with these ideas we propose metrics for evaluating the Project54 system. These metrics can be used to evaluate and compare existing ubiquitous computing applications. They can also be used to systematically evaluate potential applications. We can assess the ranges in which we expect the answers to be, once an application is complete, and use these assessments to evaluate the prospects for success of the application.

References

1. Weiser, M. The computer for the 21st century. In *IEEE Pervasive Computing,* 1, 1 (2002), 19-25.
2. Kun, A.L., Miller, W.T., and Lenharth, W.H. Computers in police cruisers. In *IEEE Pervasive Computing,* 3, 4 (2004), 34-41.
3. Heeman, P., Yang, F., Kun, A.L., Shyrokov, A. Conventions in Human-Human Multi-Threaded Dialogues: A Preliminary Study. In *Proc. Intelligent User Interfaces Conf.* (2005).

APPLICATION-LED RESEARCH IN UBIQUITOUS COMPUTING: A WIRELESS SENSOR NETWORK PERSPECTIVE

Joshua Lifton and Mathew Laibowitz

Responsive Environments Group
MIT Media Lab
Cambridge, MA 02139

1. INTRODUCTION

The broad vision of ubiquitous/pervasive computing has inspired several fields of more narrowly defined research, among them wireless sensor networks. Although more narrowly defined, the field of wireless sensor networks is nonetheless stuck in a similar application limbo. In what follows, we discuss this problem in more detail, suggest possible routes of escape, and relate the lessons learned back to the more general ubiquitous/pervasive computing community.

2. STATE OF THE APPLICATION SPACE IN WIRELESS SENSOR NETWORKS

Wireless sensor network research as a whole suffers a dearth of viable application scenarios for which wireless sensor networks are the best solution. For example, more than a few authors mention forest fire detection as an application of wireless sensor networks. In this scenario, sensor nodes are dropped from an airplane into a forest and then route temperature information back to civilization. This is untenable for a number of reasons.

Firstly, by the time a change in temperature can be detected, the fire is most likely well under way. Secondly, even with today's most environmentally friendly technology, it is unacceptable to litter forests with sensor nodes containing heavy metals, solvents, and other toxins. Finally, a single low-cost graduate student or forest ranger stationed at a fire watch tower can monitor hundreds of square miles of forest much more effectively than any sensor network thus far proposed. To our knowledge, the forest fire detection application has never been deployed, but is rather touted solely as an easily understood application with which to motivate simulation or theory.

This and other similar situations exemplify several obstacles to application-led wireless sensor network research. For example, researchers are generally unfamiliar with the application domains they are trying to address and therefore cannot accurately assess the efficacy of a wireless sensor network solution relative to a more traditional solution. In addition, most researchers do not have the resources to design, build, deploy, and maintain a wireless sensor network application. This lack of hands-on experience has contributed to the commonly accepted assumption that there is a sea of applications waiting to make use of the results of simulation and theory, thus leading many researchers to only minimally motivate their work.

This is not to say there aren't applications, simply that more effort should be focussed on fleshing them out. The remainder of this paper is a (necessarily limited) starting point for doing just that. We discuss the role applications can and should play in wireless sensor network research, suggest some simply guidelines for evaluating potential applications, examine application identification, outline four concrete categories of wireless sensor network applications, and finally summarize some high-level obstacles to application-led research.

3. ROLE OF APPLICATIONS

The role of applications is four-fold.

3.1. Validate Theory and Simulation

The ultimate test of any theory or simulation is experiment, and building real applications is a clear path toward experimentation. These type of applications are not particularly prevalent in wireless sensor network research since the scale (e.g., node count and physical size) and complexity of readily built wireless sensor network applications pale in comparison to the scale and complexity called for by most of the theories and simulations in need of testing. For example, given it is currently quite challenging to build, deploy, maintain, and monitor an application with only 100 nodes, it is not reasonable to expect to test a theory whose main result is arrived at only as the number of nodes in the network goes to infinity.

3.2. Motivate Theory and Simulation

Theory and simulation require motivation. In the context of wireless sensor networks, this often comes in the form of a specific application or class of applications. Unfortunately, more often than not, the specifics of the application are not discussed. Furthermore, the same set of example applications (e.g., forest fire detection) seem to be repeatedly cited without a critique of their plausibility or usefulness.

3.3. Sample User Needs

In the end, the killer applications of a technology are decided by the users, not researchers or developers of technology. Creating and deploying applications is a very direct way to gain insight into what users want and need (as opposed to, for example, focus groups or statistics gathered from similar domains). Very few wireless sensor network applications exist, let alone are designed for non-expert users.

3.4. Build a Base for Future Applications

One of the dominant, if understated visions of wireless sensor networks is that their utility is derived from their versatility. Accordingly, no single application has been identified that would alone warrant the widespread deployment of wireless sensor networks. Rather, the synergy between multiple diverse applications is supposedly what will motivate their deployment. Thus, every application developed has the potential to incrementally bring closer the day when it is worth the cost of building and deploying wireless sensor networks for widespread use. In this sense, there is a parallel with desktop computers; few people are willing to buy a desktop computer only for the utility a word processor program provides, but many people are willing to buy desktop computers for the aggregate utility provided by all the programs available to them.

4. METRICS FOR SELECTION, ANALYSIS AND EVALUATION OF APPLICATIONS

In the context of wireless sensor networks, there are several ways to select, analyze and evaluate applications. Here is a non-comprehensive list of points to keep in mind:

- Can the problem be solved better by centralized approaches? If there is no benefit to implementing a wireless sensor network solution, then don't.
- Interesting problems do not imply interesting applications. It may take more effort to find an interesting application than to solve an interesting problem.
- Useful algorithms and tools are not themselves applications. For example, data aggregation is useful, but not itself an application.
- Favor interesting applications over optimal applications. Interesting applications will further the field more than optimal applications at this point. Scalability, energy consumption, speed, bandwidth, etc. can be optimized afterward.

5. APPLICATION IDENTIFICATION

This is certainly the most difficult problem facing wireless sensor network application developers. In part, this is because there is still a sizable gap between what technologies are available with which to develop and envisioned applications. More seriously, however, is the problem of finding compelling applications at all. Time will hopefully take care of the former problem, but only imagination and ingenuity can cure the latter. To that end, it behooves researchers to expand their definition of a wireless sensor network to include, for example, a great diversity of physical scales. Why not microns (e.g., super dense artificial skin) or parsecs (e.g., interplanetary navigation networks)? The definition could also be expanded by considering actuators on each node, closed-loop versus interactive systems, and tiered networks.

6. EXAMPLE APPLICATION DOMAINS

We present here four broad categories of applications with examples of each category.

6.1. Augmented Sensing

The structural similarity between wireless sensor networks and biological sensor networks suggests that wireless sensor networks may be well-suited to augment biological sensor networks. For example, an extremely dense, skin-like sensor network might be embedded in a body suit in order to process incoming tactile data and then route high-level features to an off-body receiver for use in a telepresence application. Such a sensor network could also be applied as skin for robots, aiding in kinesthesis. Sensor networks distributed on a larger physical scale could also augment our natural senses. For example, a sensor network distributed throughout a building or construction site might augment a building manager or site foreman's perception of what is happening in the building.

6.2. Instant Infrastructure

Wireless sensor networks are often touted as having the potential to provide infrastructure on short notice in uncertain environments. Localization, tracking and communication services are examples of applications of use in situations arising in military operations, space exploration, and disaster relief.

6.3. Distributed Infrastructure

Situations currently employing centralized permanent infrastructure may benefit from a distributed solution enabled by wireless sensor networks. Power generation and distribution is a prime example. At present, power supplied by large generators is centrally controlled to carefully match power demanded by end users. This precludes widespread adoption of household power generators (e.g., solar panels, flywheels, and wind turbines) connecting directly to the power grid and deciding as a network when to generate or store power. On one level, the power generators and storage devices could be considered as nodes in a sensor network. On another level, each household's power generation and storage unit might have access to information culled from a wireless sensor network distributed throughout the household in order to monitor, mitigate and predict electricity use by the inhabitants and therefore make more informed decisions as to how much power to request or offer the rest of the grid.

6.4. Physically Situated Information

Embedding digitally accessible information into the physical environment (e.g., RFID tags and IR beacons) has long been a goal of the ubiquitous computing community. At the most basic level, such information could be used to support localization services. Information may also only have meaning or use in the context of a particular physical location. Graffiti is an analog example of physically situated information. A digital example might be movie posters that digitally store feedback about the advertised movie entered by passersby and/or collected from remote sources. In essence, this is an example of physically situating the viral consumer and social networks already prevalent on the Internet, thus magnifying their effect by making information available at the time and place users most want access to it.

7. OBSTACLES TO APPLICATION-LED RESEARCH

By far the most formidable obstacle to application-led research is the host of limitations imposed by using a real hardware platform. Either the researcher can develop her own platform at considerable time and financial expense, or she can use one of the very few available experimental platforms at the expense of being constrained by hardware not designed for her application and also at considerable financial expense.

Another obstacle to application-led research is the extreme emphasis on communication protocols and energy conservation. Clearly, these will be the limiting factors in the end, but applications should be pushing the bounds of the state-of-the-art in communication protocols and energy conservation, not lagging far behind. For example, sensing problems (e.g., calibration) are just as important to most potential applications, but command relatively little research focus.

Usability is another obstacle of our own making. Wireless sensor networks will not become widespread until average people can use them. We already have an idea of the applications Big Brother would like, but which applications exist that an average person would find compelling?

8. CONCLUSION

In many ways, wireless sensor networks are positioned to become the machinery on top of which ubiquitous/pervasive computing operates. Thus, the issues surrounding wireless sensor networks outlined here apply equally well to ubiquitous/pervasive computing.

We've given a brief outline from the perspective of wireless sensor networks research as to the role of applications in research, heuristics for evaluating possible applications and research directions, promising categories of applications, and obstacles to application-led research.

In addition to technological limitations, application-led research also suffers from an over-emphasis on optimization. On the other hand, pulling real users into the equation can only further the field and should be encouraged.

The potential for wireless sensor networks and ubiquitous/pervasive computing is greater than it has ever been. When all is said and done, developing compelling applications is the only way to realize this potential.

Ubiquitious Computing needs to catch up with Ubiquitous Media

Anil Madhavapeddy[1], Nick Ludlam[2]
University of Cambridge Computer Laboratory[1], Interceptor Communications Ltd.[2]
{avsm2@cl.cam.ac.uk[1], nick@intercomms.co.uk[2]}

1 Introduction

The field of Ubicomp is, in a word, "varied". A typical Ubicomp research team is highly multi-disciplinary, consisting of hardware, software and social engineers collaborating to achieve the vision of invisible computing in our day-to-day environments. This variety has led to research in a myriad of fields such as context-awareness, sensor networks, low-power computing, activity inference, and location sensing infrastructure. However, the field currently lacks a unifying factor that will drive this scattered research into real-world deployments.

This position paper points out a vertical market that could hugely benefit from the expertise of the Ubicomp research community—the media industry, currently in the throes of a digital revolution. Analogue broadcast systems such as radio and television are moving to digital formats, and consumers are starting to get familiar with "digital lifestyle appliances" promoted by industry such as the popular iPod[1] and Tivo. We discuss some of the problems created by this transition, and point out a golden opportunity for the Ubicomp research community to get involved in creating an effective platform for the future of ubiquitous media for consumers.

There are traditionally two mechanisms of delivering video content to users: (*i*) wireless or cable broadcast medium (e.g. live television); and (*ii*) physical media such as video cassettes or DVDs. More recently, a third delivery mechanism has been added to this list: the Internet, driven by the rising penetration of broadband services to US homes. With more bandwidth, consumers are able to search for and download specially encoded video files which let them view selected programs "on-demand". Aside from the copyright and legality issues, the sheer volume of activity on this front (e.g. the rise of peer-to-peer software such as Kazaa) demonstrates the strong consumer desire to break out of the "push model" of television into a more interactive and flexible content consumption model.

This paper does not attempt to create a taxonomy of problems facing the media industry. Instead, we highlight some of the more interesting problems that have arisen in recent years and examine some relevant Ubicomp projects in the field that apply to them. Finally, we describe the beginnings of a project in conjunction with the British Broadcasting Corporation (BBC) that will start to tackle some of these problems.

[1] In an effort to cut down on references, we do not cite terms which can be easily looked up via an Internet search engine.

1.1 Television On-Demand

Television has traditionally been a broadcast medium, with the consumer selecting a video stream to watch from a limited set of channels. The advent of cable television has increased the number of channels into the thousands, but the basic broadcast model is still intact. If consumers wanted to watch a video "on-demand", they would normally purchase, rent or borrow a DVD or cassette.

Recently, companies such as Tivo have been selling Personal Video Recorder (PVR) appliances which hook into a television feed and record content based on user selection to a local hard-drive. When consumers wish to watch television, they can simply flick through all the recorded programs and search until they find the content they feel like watching. The Tivo attempts to record content based on user's past preferences, e.g. "I like the Simpsons" resulting in all episodes of the cartoon to be stored. This approach is limited to programs the user actually watches on that Tivo, and breaks down when the user's interest rapidly change in the outside world (e.g. moving house or falling ill).

Problem 1: *Video appliances need to be able to infer user demand and record the content that they will want to watch based on their current activities in the wider world.*

Activity inference has been looked at by a number of Ubicomp projects. Fishkin et al discuss how objects with RFID tags attached could allow interactions with items in the home to be detected automatically [3]. Koile at al use vision to infer "activity zones" of user actions [5]. The PlaceLab project [6] uses commodity wireless hardware to retrieve user location, another useful beacon for context inference. Microsoft's AURA [9] lets users scan barcodes on everyday objects and register them on a central website—this information could be used to trigger video recording in the home (for example, scanning organic health food at the supermarket could allow the predictive recording of news snippets about that topic).

If these systems were integrated with a digital media platform, users would see the real effect of activity inference as television they want to watch is made available to them in the background.

The shift from live broadcast will have a profound impact on the multi-billion dollar advertising industry, who are already finding users capitalising on PVR systems to filter out advertising during automated recording. However, as Google has demonstrated with their successful AdWords business, users are not averse to advertisements, only irrelevant ones. Providing advertisements to users *peripherally* instead of obtrusively, e.g. when they are engaged in activities other than watching television may win back user interest.

Problem 2: *Context-aware and peripheral advertisements may drive the future of consumer video-on-demand advertising.*

Ubicomp offers a good breath of research in this field. Project Aura's vision of Distraction-free Ubiquitous Computing [1] seeks to minimise the amount of interruption a user suffers. Mankoff et al also offer insights into the use of "ambient displays" to occupy the periphery of user attention [7]. When combined with activity inference, these approaches could result in an advertising experience for the user that is very different from the existing interruptive delivery mechanism.

1.2 Physical Storage

Traditionally, the standards governing rich media have been dictated by the physical storage format being used. For example, the MPEG-2 bit-rate of 3-10Mb/s allows a dual-layered commercial DVD with a capacity of 8.5 GB to store around 3-4 hours of content (enough for a typical Hollywood movie with additional features).

Internet delivery, and the ability of consumers to encode their own content onto digital media such as hard-drives eliminates this requirement. For instance, the Apple iTunes music program allows the creation of "MP3 CDs" or "audio CDs". Audio CDs can be played back by normal CD players, while MP3 CDs use a more advanced codec that packs more music onto the CD, but can only be played by MP3-aware players. Similarly, modern mobile phones are capable of playing video that has been transcoded to the 3GPP format, and the DivX format is popular for the storage of television programs without a large degradation of image quality.

Problem 3: *The proliferation of digital media clients used by consumers has created the need for a flexible way of storing large amounts of high-quality data on their person or at home, and transcoding it on-demand to other devices such as mobile phones or televisions.*

In Ubicomp, the Personal Server project [11] provides an insight into how users can effectively carry around large amounts of data on their person. However, such a device would need high-bandwidth wireless links to the outside world, and more computation power to be able to transcode content on-demand into different formats as the user requests them (or by automatically sensing the class of device requesting the media).

As the user's collection of personal data increases, the consolidation of storage becomes a double-edged sword. The user benefits from centralizing large cassette and DVD collections into a single portable drive, but then has to deal with the long-term backup of unreliable physical media such as hard drives. Dealing with how to securely synchronize rich media across a home network and ensure there are no single points of failure is a difficult problem across heterogeneous networks and devices.

Problem 4: *There is a need for storage devices to infer and indicate how much of the information they contain is "unique"; that is, not backed up or available elsewhere in the user's storage network.*

Many Ubicomp researchers are very familiar with programming low-power embedded devices; storage synchronization is an active research area in the networking community with efforts such as the Co-operative File System [2]. Combining this research with work on tangible interfaces [10] on personal storage gives us an exciting glimpse into how a future personal storage device might look: a wireless device with a physical indicator as to how "unique" the data it contains is. Giving users an immediate notion of the value of a storage device would warn them to purchase more storage to back it up, not put critical data on it, or even just be very careful handling it. Anyone who has accidentally erased the last copy of a picture from a digital camera will appreciate how valuable this is as more of our media storage shifts into digital intangibility.

1.3 Publishing for All

The digital media revolution extends to production equipment as well as playback. The majority of cameras sold in recent years have been digital, and camcorders to record video have dropped to affordable prices. This has led to consumers having an increasing need for video editing, conversion and viewing software. In addition, they also need to easily share it with friends and family.

Problem 5: *There exists no standard mechanism for "hyper-linking" into video based on a variety of factors such as its location, time and content, in order to allow other Ubicomp interaction techniques such as visual tags to be used by consumers to easily share rich media.*

Universal video links might be links to local devices on the wireless network, or if that device is unavailable, a central Internet server for a slower download. One scheme proposed by Kindberg [4], which allows users to access local physical resources as Web resources, could be extended to solve this problem. Once these hyperlinks can be expressed, traditional Ubicomp interaction techniques such as visual tags [8] can be used to make it easy for consumers to print out "link cards" to give to their friends, or just dynamically generate them between two compatible devices such as mobile phones (both of which are equipped with a screen and a camera).

2 One Platform to Rule them All

Open-source software support for video processing is rapidly gaining ground. Projects such as MPlayer and FFmpeg offer backends for performing video conversions to and from many formats such as DivX, MPEG or 3GPP.

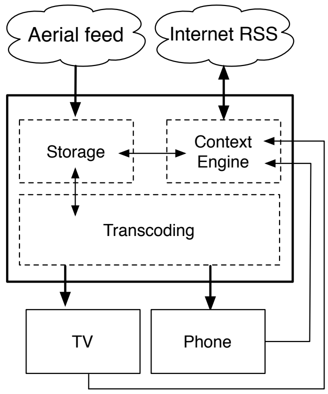

Figure 1: Architecture for our prototype Contextual PVR

MythTV is a PVR project which uses Linux and video hardware to record and manage streams of television data.

However, none of these individual elements mesh together to genuinely move the user experience beyond the industry state-of-the-art. We have begun a joint collaboration with the British Broadcasting Corporation (BBC) to investigate the potential impact that large-scale local storage of digital content provides. We have constructed a platform with enough storage to continuously record 7 days worth of television across all BBC channels broadcast in the UK. This platform will be used to explore the socio-technical challenges introduced to *both* the broadcasters and consumers in the emerging digital market.

As described earlier in this paper, there are numerous challenges to solve: (*i*) allowing users to search the vast amount of recorded television without requiring a keyboard and mouse; (*ii*) analyse how viewing habits and user expectations change as media becomes truly "on-demand" by being continuously recorded and stored; (*iii*) classify videos from sources such as Internet information feeds via RSS or from the XML meta-data held in the broadcasts themselves; and (*iv*) a "context-feed framework" to allow external devices such as televisions or mobile phones to provide feedback about user activity in the real world. Underpinning all of these is the requirement for "invisible content security" via a Digital Rights Mechanism that protects intellectual property while still granting fair use rights and not suppressing spontaneous interaction between individuals.

We urge the Ubicomp research community, consisting as it does of a rich variety of hardware, software and social engineers, to step forward and examine how their research could help improve the handling of digital media for the next generation of home appliances. Ubicomp has so far been a scattered research field, with several discrete (though interesting) areas of research such as low power computing, context-awareness, location-awareness, sensor networks and activity inferencing. Demonstrating our ability to solve real user issues through the unified deployment of our technology into a rapidly transforming digital society would finally initiate Ubiquitous Computing as a viable long-term research area.

3 Acknowledgements

This work was sponsored by Intel Research Cambridge and Interceptor Communications Ltd. We are grateful to Richard Sharp, Eleanor Toye, David Scott and Alastair Beresford for their caffeine-fuelled "ubi-rants" which shaped the thoughts presented here.

References

[1] Project Aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 1(2):22–31, 2002.

[2] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, Chateau Lake Louise, Banff, Canada, October 2001.

[3] K. Fishkin, B. Jiang, M. Philipose, and S. Roy. I sense a disturbance in the force: Long-range detection of interactions with rfid-tagged objects. In *Ubicomp*, September 2004.

[4] T. Kindberg. Implementing physical hyperlinks using ubiquitous identifier resolution. In *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, pages 191–199. ACM Press, 2002.

[5] K. Koile, K. Tollmar, D. Demirdjian, H. E. Shrobe, and T. Darrell. Activity zones for context-aware computing. In *Ubicomp*, pages 90–106, 2003.

[6] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device positioning using radio becons in the wild. In *Proceedings of Pervasive 2005*, LNCS. Springer-Verlag, May 2005.

[7] J. Mankoff, A. K. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *CHI '03: Proceedings of the conference on Human factors in computing systems*, pages 169–176. ACM Press, 2003.

[8] D. Scott, R. Sharp, A. Madhavapeddy, and E. Upton. Using visual tags to bypass bluetooth device discovery. *ACM Mobile Computer Communications Review*, January 2005.

[9] M. A. Smith, D. Davenport, H. Hwa, and T. Turner. Object auras: a mobile retail and product annotation system. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 240–241. ACM Press, 2004.

[10] B. Ullmer and H. Ishii. The metadesk: models and prototypes for tangible user interfaces. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223–232. ACM Press, 1997.

[11] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light. The personal server: Changing the way we think about ubiquitous computing. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages 194–209. Springer-Verlag, 2002.

Experiencing Technology before it exists: A Case Study

Florian Michahelles
Perceptual Computing and Computer Vision Group,
ETH Zurich,
Switzerland
michahelles@inf.ethz.ch

Bernt Schiele
Multimodal Interactive Systems
Darmstadt University of Technology,
Germany
schiele@informatik.tu-darmstadt.de

1 Introduction

Ubiquitous and wearable computing have been around for more than a decade now [15]. However, there is still no consensus on what the new technologies are going to be used for. Traditionally, application development in ubiquitous and wearable computing communities has been mostly technology driven. This has lead to a variety of applications of which only few are really being used in practice, e.g. [2, 4, 6]. This may be seen as an indication that current practices are too unsystematic and rely too much on developers' and designers' intuition alone.

We found that there is a latent tension between a technologically advanced solution and the focus on the user. For the introduction of advanced new technologies, such as ubiquitous or wearable computing, users usually cannot be familiar with this new opportunities and often have difficulties realizing its benefit [1]. Simultaneously, designers and developers often lack the understanding of the relevant issues in specific working domains. In order to create solutions that really go beyond incremental changes of established routines, these limiting boundaries have to be overcome: Real innovation is not generated by technology itself, but technology can act as a vehicle to create new opportunities that innovate and change established routines and work practices.

This paper consists of three main sections. Section 2 lists and discusses a set of requirements which, in our experience, help to introduce and generate new applications that aim to depart radically from today's work practices. Section 3 describes the notion of *X'treme Prototypes* as a method to generate innovative applications. Prototypes, referred to as *X'treme Prototypes*, are used as vehicles to provide a grasp of how future implementations could look like regardless of their immediate feasibility. Finally, section 4 reflects on how well the *X'treme Prototype* method fulfills the initially posed requirements.

2 How to build innovative applications

This section discusses various requirements an application development method should meet in order to produce innovative ubiquitous and wearable computing applications. This list of requirements is based on our own experience to introduce innovative applications as well as on discussions with other researchers.

Early stage support: Currently, designing ubicomp and wearable computing still means designing for the early stage (which is different than writing task descriptions for creating use cases as suggested by software engineering disciplines [13]): designers are still looking for ideas and user needs applications might solve. Whereas methods from software engineering [13] and HCI have a long tradition in describing efficient processes for developing customized applications, they do not support the search for compelling problems that radically depart from today (as ubiquitous and wearable computing aim).

Balanced user involvement: Instead of the typical asymmetry between developers and users (users have needs, developers fulfill them), both parties have to productively learn from each other in a complementary partnership.

Radically depart from today: The most important challenge is to create innovative solutions which may radically differ from current practice. It is not enough to 'just' fulfill the user's 'incremental' needs. Rather the solution may enable and stimulate a fundamental and effective change in work practice.

Multiple stakeholder involvement: In the course of different projects we realized that meeting the end-users of applications is necessary but not enough. The important point is to talk not only to the final end-users who may be most affected but also to other parties and decision makers in order to gain a more holistic view and obtain stimulating input.

Stimulation of stakeholders: We experienced that users and other stakeholders are often locked into their world and work practices. This makes it difficult for them to look beyond daily practice and imagine new and innovative solutions. However, presentation of new concepts that are concrete and graspable (but not necessarily ready-to-use) can initiate fruitful discussions .

Mediation between users and developers: Users and developers have to be empowered to view beyond established routines and to envision prospective opportunities for change. This information has to be shared in a way which both parties understand.

Developer guidance: A development process should provide clear single steps that guide the developer towards a final development goal. Only if a development method provides enough guidance for the developers, it will be useful for a wide audience.

[1] as Don Ballman mentioned in a panel at InterCHI'93 [1]: *users are afraid of disrupting established routines and are unaware of technological advances.*

Exploratory nature: In contrast to traditional product/software development for creating innovations the goal is rather to exceed the user's expectations. This means, there has to be more than only validating specifications with the customer. One has to discover relevant problems by experimentation, by trying out, and touching new concepts: stakeholders should be able to explore and reflect on alternative approaches.

Yield feasible results in the end: For a successful development method it is essential that, after a number of iterations, it yields feasible results. This does not necessarily mean a ready-to-use product, but at least an output that developer and user consider to be useful.

Allow for multiple cycles with the user: The basic idea of multiple cycles [**?**] allows the developer to take advantage of what was learned during the development of earlier versions of a system. However, learning should come from both the development and the cooperation with the user.

Rapid development cycles: Many development methods such as extreme programming [3] or rapid prototyping [14] aim at meeting the user's expectations in short converging cycles. Short and fast cycles are the premise for several iterations during a project life.

Use of toolboxes: Toolboxes are a technical prerequisite for rapid developments. Re-using frameworks and architectures over and over again is a key factor for short development cycles. Though re-usability in ubicomp is still in it's infancy the benefit has been acknowledged and partly addressed for software (e.g. Context Toolkit [12], Context Fabric [8]) and hardware (e.g. Smart-Its [5]).

3 The *X'treme Prototyping* Method

Many if not all of the requirements listed in section 2 are addressed at least individually in various approaches of software engineering [13], user-centered design [11], and participatory design [7]. We feel, however, that none of the current approaches does respect all of them sufficiently well in order to generate and create innovative applications and application scenarios for ubiquitous and wearable computing. This section therefore presents a method called *X'treme Prototyping* which is synthesized from various well-known approaches.

To overcome the tension between user-focus and the introduction of radically new concepts and technologies the application of so-called *X'treme Prototypes* are used to explore new principles and future user needs in cooperation with stakeholders of a specific domain.

1. **Choose a compelling problem domain.**
   This suggests to deliberately choose a problem domain that simultaneously allows for a successful integration of research results and a meaningful grounding in a user's domain.

2. **Understand the user's application domain.**
   The goal of this phase for the developers is to gain a clear understanding of the user's ultimate goals, driving forces and constraints of established routines, and the current implementation of work practices.

3. **Distill radically new concepts by identifying new opportunities for change.**
   The result of this phase should be a new concept that helps users to solve at least one of their goals in a new way regardless of the current implementation looks. Obviously, this phase is the most creative part of the presented approach.
4. **Develop an *X'treme Prototype* to present one or several radically new concepts to the user.**
   This step develops a prototype that can represent and illustrate the strengths of one or several concepts to the user.

5. **Provide experience for the users and stimulate users and stakeholders.**
   This phase aims to provide hands-on experience of new concepts by using an *X'treme Prototype*. As such users and stakeholders should be stimulated to articulate future needs and imagine new possibilities beyond their daily practice and knowledge.

6. **Iterate, several iterations may be necessary to let the process converge.**
   Deliberate incorporation of user feedback (from the previous phase) should be used to revise the concept (Step 3) and result in changes of the prototype (Step 4). The iteration process can be stopped if, firstly, developers are confident with the new innovative application and if, secondly, the users judge the application to be generally feasible and useful.

4 Discussion and Conclusion

By posing a list of requirements (Section 2) we characterized how an ideal development process should be composed for innovative application development in ubiquitous and wearable computing. Then *X'treme Prototypes* where introduced as a development method for ubiquitous and wearable computing aiming to respect those requirements. We now want to reflect on the challenges these requirements pose when applying *X'treme Prototypes* in practice. Our experienced is based on two projects: avalanche rescue using wearable sensing technology (A-Life) [10] and applying sensors in professional skiing [9].

In both projects we experienced that the two requirements *Radically depart from today* and *Balanced user involvement* are both central but at the same time create a tension due to their very different nature: For example during early discussions about the benefits of wearable sensors in avalanche rescues with mountaineers – as the actual end-users – we felt a strong and general reluctance against technology. This reluctance, while being an important issue, was a major obstacle during the early stages of the project which hindered progress significantly. From that we learnt that the distillation of opportunities in isolation from users and the preparation of an *X'treme Prototype* illustrating new concepts can help to overcome this tension. Furthermore, when we presented our concept to alpine emergency physicians we were encouraged to continue with our project. *Multiple stakeholder integration* proved important in order to broaden our scope to the entire field instead of depending on perspectives of single stakeholders. In both projects *X'treme Prototypes* helped our stakeholders to think beyond established routines. The presentation of a radar sensing heart-rate in the A-Life project and sensing platform for skiers immediately yielded in a *stimulation of stakeholders*: a black-box device recording life signs, different sensor placements etc. were proposed. This clearly showed that providing the context of work through an *X'treme Prototype* can help stakeholders to better express their needs and think beyond. As such, this immediately provided an understanding of constraints posed by practical limitations, e.g. velocity is a very desired feature in skiing, but sensing is very difficult. This grounding of our concept into the users' work context supported the *mediation between users and developers*. Regarding *developer guidance* we experienced that there is a limit of providing systematic guidance towards innovative development, since creating innovation is a process of creativity. This creativity must not be hindered by the process. Nevertheless, as much guidance as possible without eliminating creativity is a desired goal. Working towards several prototypes sets clear goals without restricting creativity. *Exploratory nature* is key for stimulating stakeholders. Nevertheless, especially during the skiing project we had to cope with the difficulties of providing a small light-weight early robust prototype to be worn by a skier outdoors in a cold environment. For future developments it would be useful to reuse this gained knowledge, as such *use of toolboxes* is a very important goal to follow for ubiquitous and wearable computing in order to enable *rapid development cycles*. Unfortunately, this is not yet the case. *Yield feasible results in the end* is difficult. Nevertheless, in the A-Life project we could arrive at a stage that a Swiss avalanche beacon manufacturer plans to integrate the final results of that project into the new generation of devices. In the skiing project we have the commitment of Swiss Ski to continue with further testing and application in training sessions with the national team.

In conclusion, using the proposed *X'treme Prototype* method we experienced open discussions revealing new and innovative issues since the *X'treme Prototype* provides hands-on experience. In our opinion, this combination of technological change and the revision of established practices may lead towards real innovation. However, other researchers and practitioners from HCI and design might argue that processes similar to the one presented are current practice. We strongly argue however that the ubiquitous and wearable communities are very seldom using those techniques and many researchers might not even be aware of the potential benefit using any of them. This is why we think listing and discussing the above requirements is beneficial.

References

[1] D. Ballman. User involvement in the design process: Why, when and how. In *INTERCHI*, 1993.

[2] J.E. Bardram. Applications of context-aware computing in hospital work. In *ACM symposium on Applied computing*, 2004.

[3] K. Beck. *Extreme programming explained: embrace change*. Addison-Wesley, 2000.

[4] N. Davies, K. Mitchell, K. Cheverst, and G. Blair. Developing a Context Sensitive Tourist Guide. In *MobileHCI*, 1998.

[5] Lars Erik Holmquist et al. Building intelligent environments with smart-its. *IEEE Comput. Graph. Appl.*, 24(1):56–64, 2004.

[6] S. Fuller, Z. Ding, and A. Sattineni. A Case Study: Using the Wearable Computer in the Construction Industry. In *ISARC*, Taipei, Taiwan, 2000.

[7] J. Greenbaum and M. Kyng. *Design at work: Cooperative design of computer systems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

[8] Jason I. Hong and James A. Landay. An Infrastructure Approach to Context-Aware Computing. *Human-Computer Interaction*, 16, 2001.

[9] F. Michahelles, E. Crameri, and B. Schiele. Design, Implementation and Testing of a Wearable Sensing System for Professional Downhill Skiing in Cooperation with Trainers. In *2nd International Forum on Applied Wearable Computing (IFAWC)*, Zurich, Switzerland, 2005.

[10] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele. Applying Wearable Sensors to Avalanche Rescue. *Computers & Graphics*, 27(6), 2003.

[11] D. A. Norman and S. W. Draper. *User centered system design*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[12] D. Salber, AK. Dey, and GD. Abowd. The context toolkit: Aiding the development of context-enabled applications. In *CHI*, pages 434–441, 1999.

[13] A. Sutcliffe. Scenario-based requirement analysis. *Requirements Engineering*, 3:46–65, 1998.

[14] S. Tripp and B. Bichelmeyer. Rapid prototyping. *Educational Technology Research & Development*, 38(1):31–44, 1989.

[15] M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94–104, September 1991.

An Application-led Approach for Security Research in Ubicomp

Philip Robinson
Telecooperation Office (TecO), University of Karlsruhe
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany

Abstract. The difficulties of scoping security-related research within ubiquitous and pervasive computing are discussed. The paper provides a condensed background to this research domain, and shows how a generalized, application-oriented research methodology is being applied to a thesis on Intrusion Detection, such that a good balance of theory, technology and scenarios may be obtained.

1 Introduction

Application-led research encompasses theory, technology and scenarios. Nevertheless, problems arise when there is too much focus by researchers on a specific aspect of the research. Having both reviewed and contributed research in the area of security in Ubicomp[1] [9], it has often been observed that the content and focus of security-related research are either purely theoretical and hence not practically realizable, present too much technical details (e.g. equipment specification, cryptographic key-sizes, standards) and leave the reader without an explicit scientific conclusion, or describe scenarios/ stories that present very "special-case" problems with very limited solutions and outlandish assumptions. This paper offers a methodology for balancing these three aspects of research, using security as a case study. The particular area of security being considered is Intrusion Detection, as it is still relatively unexplored in UbiComp but has very clear analogies with real world social interactions and concerns.

Before proceeding to the central theme of the paper, it is necessary to have a clear understanding of some terminology. The first term that must be understood is that of "Application", as there tends to be a common misperception that an application is equivalent to a storyboard-like description or software. The description of "Application" being used in this paper is *the way in which processes, tasks and information are organized in order to optimally and consistently achieve specific objectives.* A scenario is a very specific instance of an application with very specific properties, assumptions and a storyline. Software and technology are tangible solutions for enhancing the way that the everyday objectives of people and organizations are met i.e. the application. Nevertheless, rapid deployment of technology into society and businesses often incurs problems for usability and management [10]. This is a particular concern for security, as new technologies and ideas may introduce new risks and opportunities for intrusion.

The paper proceeds by describing the proposed methodology, followed by section 3, where it is applied to a thesis on Intrusion Detection.

2 An Iterative Methodology for Application-Led Research

Application-led research should commence with clearly stated objectives and criteria by which the research will be evaluated. An approach of "iterative refinement" is suggested, as this allows a researcher to separate theory, technology and scenarios into different foci of research, and progressively refines the argumentation and results. Börger proposes strategies for iterative refinement of systems engineering using ASMs (Abstract State Machines) [5], from which similar principles are adopted for motivating iterative, objective-driven research. The resultant, iterative, four-step methodology proposal is described below:

Step 1: (Scope) Identify application domain and objectives to be realized, as well as the conditions under which the objectives are considered satisfied. Identify the *subjects* (entities with management roles in order to meet objectives), *utilities* (mechanisms employed by subjects) and *objects* (entities managed by subjects in order to meet objectives).

Step 2: (Theory) Postulate a ground model that proposes a conceptual strategy for meeting the application's objectives. Secondly, specify rules governing the interaction between subjects and objects based on how the objectives are decomposed.

[1] Ubicomp is used as a placeholder for both ubiquitous and pervasive computing. Despite the different origins of the two communities, there is no real distinction between the two today.

Step 3: (Technology) Propose the hardware and software that can either extend or newly implement mechanisms for meeting the objectives. Mechanisms are affiliated with functionality of subjects, utilities or objects.

Step 4: (Scenario) Evaluate the theory and technology proposals based on the objectives and constraints identified. This step is also useful as a "reality check", to validate claims made by the theory and technology with reference to enhancing the application needs of people and organizations. A good scenario should consider the target audience but make sure that the application objectives and scope specified in step 1 are maintained or qualified, without becoming superficial or overly imaginative.

The iterative property of the methodology suggests that an outcome of the scenario analysis (or feedback), may serve to refine the scope of the research, the theoretical assumptions and the technology considered. Furthermore, the scenario can be used to both make problems clear as well as present solutions.

3 Applying the Methodology to Intrusion Detection

In the well-known 1991 position paper of Weiser [12], he discussed the possibility of well-implemented ubiquitous computing systems offering enhancements to the way information privacy is traditionally handled, along with the observation that cryptographic techniques were already in existence for securing messages passed between computers. In a later paper [6] published in the 1999 "Pervasive Computing" edition of IBM Systems Journal, Weiser and the group at PARC issued another statement on the topic, identifying "the lack of control" as the principal problem for privacy, as it becomes increasingly harder to manage dynamic and complex interconnections, information flows, usages, failures and actions, characteristic of Ubicomp systems. Additionally, there have also been several theses and publications related to the usage of sensor-derived context information for the enhancement of security, such that security becomes more adaptive, representative of the circumstance of its subjects and based on a broader spectrum of attributes [3, 8]. The topic of Intrusion Detection has not been considerably addressed within Ubicomp, apart from what could be considered related work in the areas of mobile ad hoc networks [7] and wireless communications [1]. Nevertheless, the above citations provide a foundation for considering how Ubicomp can be applied to detecting and controlling intrusions, and why this is an important topic. An "Intrusion Detection System (IDS)" can be considered as an "Application", in that people and organizations often express the objective to protect their assets against theft or their privacy against intrusion. The proposed research methodology can therefore be applied as follows:

Step 1 (Scope): The *objects* of an IDS may include but are not limited to data, services, and physical items, as these are the ultimate goal of an intruder. The *subjects* of an IDS are therefore owners, administrators and users, while utilities are required for specifying rules and profiles, monitoring the success of these *rules*, and appropriately notifying and responding to intrusion alerts/ alarms that arise if a specified rule fails. This therefore describes the scope of the application being considered and already lends to developing a ground model.

Step 2 (Theory): The original papers on IDS were written by Dennings and Neumann in 1987 [4]. There is also a detailed and more recent taxonomy of IDS research available from Axelsson, from which the general properties of an IDS can be extracted [2]. These reliable citations were used to derive the requirements for an IDS, depicted as an ASM in figure 1.

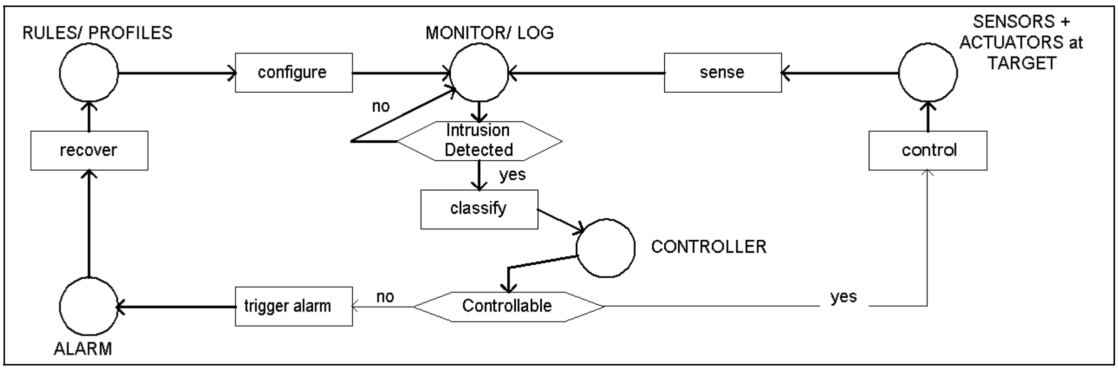

Figure 1: Proposed ASM ground model for an IDS

An intrusion detection system is bootstrapped with a definition of profiles of "normal or accepted" behaviour. The system then monitors in real-time or does inspections of system logs, with the goal of classifying sensed activity data from the targets against the profiles. When an intrusion is detected, a controller is selected based on the classification of the intrusion. If there is no controller capable of controlling the intrusion, then an alarm is triggered until the system can be recovered (typically by a site security officer), which may entail an enhancement in the system's detection profiles.

Step 3 (Technology): What happens to the above theoretical model when Ubicomp is introduced? Kindberg and Fox have identified two key features of Ubicomp systems, namely, spontaneous interaction and physical integration [9], leading to the *volatility* and *boundary* principles respectively. Using these as refinement parameters of the model, the following requirements have been derived:

- *Configuration* – cannot assume central administration nor fixed detection profiles
- *Sensing and Classification* – the availability and validity of sensors and classification schemes change as the boundary changes.
- *Logging* – there is the issue of ownership of and access to logged data after the security boundary has been "dissolved" or modified
- *Controls and Alarms* – decisions about control and alarms need to be efficiently coordinated, in the case of shared ownership, to minimise false-positives and false-negatives
- *Recovery* – the feasibility and validity of a recovery plan has to be weighed based on the stability of the target and configuration of the security boundary

The working solution for the thesis suggests a model for selecting and reconfiguring specific roles in the IDS in response to changes in the security boundary and interactions.

Step 4 (Scenario): An area where Ubicomp technologies show commercial fortitude is that of shipping and logistics. Goods are transported between different points and are placed in intermediate holding areas along the way. Each holding area has different conditions and provides different services and appliances for the care of the goods. Different models could be applied to detecting and responding to intrusions, where an intruder is defined as someone or something whose presence or behaviour threatens the progress of the goods being delivered and intact. One model could be *localized*, where each item is responsible for detecting and responding to intrusions, but this would imply that each item would need to be very expensive in terms of communications, sensing and processing. A *centralized* model could be considered, where all processing is undertaken by one node, but this would result in complex detection logic at an overloaded and vulnerable central point of attack. Using scenarios to aid in understanding the problem, the proposed model follows progress in the area of "collaborative intrusion detection". However, the differentiating contribution of the thesis is the dynamic configuration and operation of a collaborative IDS.

4 Summary

This paper has provided a general methodology for performing application-oriented research primarily in Ubicomp. This methodology has been shown using the example of security, namely Intrusion Detection, which is a thesis under development by the author. By considering Intrusion Detection as an Application, based on the definition given above, it was possible to provide very clear research objectives and parameters by which the models and solutions could be evaluated. In addition, the theoretical, technological and scenario elements of the research complement each other, which, as stated at the beginning of the paper, should be the goal of application-led research.

References

[1] Adelstein F., Alla P., Joyce R., and Richard G. G., "Physically Locating Wireless Intruders" in Proceedings of IEEE 2004 IAS Conference, Las Vegas, Nevada, April 2004.

[2] Axelsson S., "Intrusion Detection Systems: A Survey and Taxonomy". Technical report 99-15, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, March 2000

[3] Covington M. et. al, "Securing Context-Aware Applications Using Environment Roles", SACMAT 2001

[4] Denning D.E., "An Intrusion Detection Model," IEEE Trans. Software Eng., vol. 13, no. 2, Feb. 1987, pp. 222–232.

[5] E. Börger, "The ASM Ground Model Method as a Foundation for Requirements Engineering." Verification: Theory and Practice 2003

[6] Kindberg T., Fox A., "System Software for Ubiquitous Computing",IEEE Pervasive Computing, 2002

[7] Parker J., Undercoffer J.L., Pinkston J. and Joshi A., "On Intrusion Detection in Mobile Ad Hoc Networks", IEEE Workshop on Information Assurance 2004

[8] Robinson P., Beigl M., "Trust Context Spaces: An Infrastructure for Pervasive Security in Context-Aware Environments". Springer, SPC 2003

[9] Robinson P., Vogt H., Wagealla W., "Privacy, Security and Trust within the Context of Pervasive Computing." ISBN: 0-387-23461-6, Kluwer 2005

[10] Smith S.W., Spafford E.H., "Grand Challenges in Information Security: Process and Output" IEEE Security & Privacy, 2004

[11] Weiser M., "How Computers Will Be Used Differently in the Next Twenty Years". IEEE Symposium on Security and Privacy 1999

[12] Weiser M., "The Computer for the Twenty-First Century. Scientific American". September 1991

Deploy or Die: A Choice for Application-Led Ubiquitous Computing Research

Richard Sharp

[1]Intel Research Cambridge,
15 JJ Thomson Avenue, Cambridge, CB3 0FD, UK.
richard.sharp@intel.com

1 Introduction

For the past fifteen years Ubiquitous Computing researchers have been exploring how computing can be pushed beyond the traditional desktop environment and seamlessly *"woven into the fabric of our everyday lives"* [13]. Yet, despite such grand vision, the vast majority of UbiComp research is published and forgotten long before it has any impact on everyday life, let alone woven into its fabric. I believe that the UbiComp community urgently needs to address this issue before the disparity between the now cliched rhetoric and the lack of real world impact drives the field into disrepute.

In this position paper I argue that recent technological developments make it possible for Ubiquitous Computing applications to be deployed on a global scale and that, with a few exceptions, the research community is currently failing to embrace this opportunity for real-world impact (Section 2). I propose a direction for application-led UbiComp research that makes widespread deployment its success criteria (Section 3) and discuss the attitudes and practices within the community which are currently hampering the pursuit of deployable applications (Section 4). Finally, I highlight one pioneering research project that has already demonstrated that building deployable UbiComp applications can generate both global impact and research contributions (Section 5).

2 The Computer for the 21st Century is Here. Where is UbiComp?

In 1991, Weiser stated that the hardware required for ubiquitous computing would come in two parts: (*i*) cheap, low-power computers that include equally convenient displays; and (*ii*) a network that ties them all together [13]. These two requirements have arguably been satisfied for the past five years: programmable cell-phones, PDAs and laptops have become commonplace and a plethora of wireless networking technologies (e.g. Bluetooth, 802.11, 3G) are now standard.

So, given that the necessary infrastructure now exists, where are the context-aware applications [10], augmented homes [5], smart offices [12], geo-annotation systems [4], electronic tourist guides [3] (and dangling pieces of string [14]) that UbiComp promised? By contrasting the claims of UbiComp papers with our everyday experiences it is clear that the former are detached from the latter.

The UbiComp community has survived by continually adapting its research program (application domains chosen in an ad-hoc manner) in order to save its hard core assumption (that computers will become invisible, automatically inferring and catering for our every need). To remain credible, at least some of the technologies, applications, user-interfaces and usage models that the field has predicted must soon be seen in the real world. Like AI before it, UbiComp risks being categorized as what Imre Lakatos terms a "*degenerating research program*" [7].

3 A Deployment-Driven Methodology for UbiComp Research

Given this urgent need for real-world impact, I propose that application-led UbiComp research projects adopt the following deployment-driven methodology: (1) build a well- engineered, robust UbiComp application that leverages existing infrastructure (WiFi, PDAs, Laptops, Cell-phones etc.); (2) release this application publicly; (3) build a real- world user-base around the application; and (4) study this user-base and learn from users? experiences.

The vast majority of UbiComp projects that attempt to follow this methodology will inevitably fail to progress to stage 3; many uncontrollable factors contribute to whether a real-world user-base can be established. However, the hope is that, if the community as a whole makes a concerted effort to build deployable applications (and if UbiComp research is indeed relevant to people's everyday lives) then some will attract significant user-bases. These successful applications will be lucky enough to proceed to stages 3 and 4, generating both much-needed impact for the UbiComp community and valuable insights into how or why the applications were adopted by users (on a potentially global scale).

Projects that do not manage to build real-world user-bases should not be regarded as failures. By merely achieving stage 1 they will have encountered and solved interesting research challenges, e.g. how was the application designed and adapted to work on existing infrastructure? What engineering approaches were used to facilitate scalability?

This proposal is hardly revolutionary; indeed, many would argue that most technical computer science research already proceeds in this way. UbiComp, however, is certainly not embracing a deployment-driven methodology. Of the 26 projects presented in the proceedings of UbiComp04, not one describes a publicly released application that users can download and benefit from. A single project, Krumm's *NearMe Wireless Proximity Server* [6], achieves stage 1 but fails at stage 2: as I write this article I can find no way of downloading either his client- or server-software onto my laptop.

4 Attitudes Within UbiComp Hampering Pursuit of Deployment

Given that other Computer Science research disciplines have successfully generated impact by applying the above 4-stage methodology, why has UbiComp not followed suit? I believe that there are two common attitudes in the community that are hampering the development of deployable UbiComp applications. These are discussed below.

4.1 Repeating ParcTab: An Obsession with Building Custom Hardware

The philosophy of using custom hardware to support application-led UbiComp research dates back to the ParcTab Ubiquitous Computing Experiment where Want et. al. argued that it allowed them to "*glimpse into the future* [11]. At this time researchers had no alternative but to build custom hardware in order to explore the potential of Ubiquitous Computing; there was no existing infrastructure that could support UbiComp applications. The success of the pioneering ParcTab project greatly influenced the UbiComp community. In particular, its approach of deploying custom hardware to explore futuristic applications became generally accepted as the de-facto methodology for technical UbiComp research.

Today, when Weiser's requirements for Ubiquitous Computing infrastructure have been met, this approach must be challenged. Building custom hardware infrastructure still allows one to *glimpse the future*. However, it also eliminates all possibility of public release, widespread deployment and, therefore, real-world impact. In a time when these goals are attainable, researchers should think long and hard about the opportunity-cost of building custom infrastructure.

4.2 Hundreds of Small, Disparate Projects; No Community-Wide Efforts

As evidenced by the proceedings of Ubiquitous Computing conferences, the UbiComp community has typically focused on small, disparate projects. Although researchers have sometimes collaborated in lab-sized teams to engineer larger projects, there have been few (if any) community-wide efforts. If a deployment-driven methodology is to be adopted, greater collaboration is essential.

Designing and implementing robust deployable applications is expensive and time-consuming requiring, in many cases, more resources than a single research group can provide. Other CS research disciplines have successfully tackled this problem by using open source development techniques. For example, consider the Operating System community's BSD UNIX [9] and, on a smaller, but still impressive scale, the

Programming Language Community's *Haskell* language and its associated compilers and interpreters [1]. It is vital that the UbiComp community mirror this approach, publicly releasing source code and actively seeking to build on each other's implementations.

5 An Exemplary Deployment-Driven UbiComp Project

Although I have argued that the UbiComp community is largely failing to embrace the opportunities for real-world impact, there is one notable exception: PlaceLab [8].

PlaceLab is a successful UbiComp project that has followed a deployment-driven methodology. The project aims to enable commodity hardware clients like notebooks, PDAs and cell phones to locate themselves by listening for radio beacons that already exist in the environment (e.g. 802.11 access points, GSM cell phone towers, and fixed Bluetooth devices). A robust software implementation has been developed and made available for public download in a variety of formats including Windows XP, Linux, Mac OS X, Windows CE/Pocket PC and Nokia Series 60 Phones. As a result a significant global user-base has now been established.

The PlaceLab project did not invent the idea of using existing radio beacons for location: there is earlier literature on this topic [2]. The contribution of PlaceLab was to take this embryonic idea and produce a well-engineered platform capable of scaling to a global deployment. In this process the PlaceLab researchers encountered and solved problems that initial proponents of radio beacon location systems had not envisaged. Furthermore, they generated impact for the UbiComp community by developing a global, genuinely ubiquitous location infrastructure.

6 Conclusions

For many years UbiComp researchers have imagined an age when low-power computing devices, display technology and wireless networking capability would be truly ubiquitous. That time is now and, by exploiting this existing infrastructure, the community at last has a chance of achieving real-world impact. However, the current culture of lab prototypes and small, disparate research programs is preventing UbiComp from reaching its potential. In this paper I have presented a deployment-driven methodology which attempts to address this issue and have highlighted areas in which community-action is required if the approach is to be successful.

The next few years will determine the success or failure of Ubiquitous Computing research. The choice is simple: we must deploy (thus demonstrating our relevance to 21st century computing and silencing our critics) or die.

References

[1] `http://haskell.org/`.

[2] BAHL, P., AND PADMANABHAN, V. N. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of INFOCOM* (2000), pp. 775–784.

[3] CHEVERST, K., DAVIES, N., MITCHELL, K., AND FRIDAY, A. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *Proceedings of MobiCom '00* (2000), ACM Press, pp. 20–31.

[4] ESPINOZA, F., PERSSON, P., SANDIN, A., NYSTRM, H., CACCIATORE, E., AND BYLUND, M. Geonotes: Social and navigational aspects of location-based information systems. In *Proceedings of UbiComp '01* (2001), Springer-Verlag, pp. 2–17.

[5] KIDD, C. D., ORR, R., ABOWD, G. D., ATKESON, C. G., ESSA, I. A., MACINTYRE, B., MYNATT, E. D., STARNER, T., AND NEWSTETTER, W. The aware home: A living laboratory for ubiquitous computing research. In *CoBuild '99: Proceedings of the Second International Workshop*

on Cooperative Buildings, Integrating Information, Organization, and Architecture (1999), Springer-Verlag, pp. 191–198.

[6] KRUMM, J., AND HINCKLEY, K. The NearMe wireless proximity server. In *Proceedings of Ubicomp 2004*, vol. 3205 of *LNCS*, Springer-Verlag.

[7] LAKATOS, I. History of science and its rational reconstruction. In *PSA* (1971), R.C.Buck and R.S.Cohen, Eds., Reidel.

[8] LAMARCA, A., CHAWATHE, Y., CONSOLVO, S., HIGHTOWER, J., SMITH, I., SCOTT, J., SOHN, T., HOWARD, J., HUGHES, J., POTTER, F., TABERT, J., POWLEDGE, P., BORRIELLO, G., AND SCHILIT., B. Place lab: Device positioning using radio beacons in the wild. To appear, Pervasive 2005.

[9] MCKUSICK, M. A berkeley odyssey. *Unix Review 3*, 1 (January 1985).

[10] SCHILIT, B., ADAMS, N., AND WANT, R. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA, US, 1994).

[11] WANT, R., SCHILIT, B. N., ADAMS, N. I., GOLD, R., PETERSEN, K., GOLDBERG, D., ELLIS, J. R., AND WEISER, M. The ParcTab ubiquitous computing experiment. Tech. Rep. CSL-95-1, Xerox Palo Alto Research Center, March 1995.

[12] WARD, A., JONES, A., AND HOPPER, A. A New Location Technique for the Active Office. *IEEE Personal Communications 4*, 5 (October 1997), 42–47.

[13] WEISER, M. The computer for the 21st century. *Scientific American 265*, 3 (January 1991), 66–75.

[14] WEISER, M., AND BROWN, J. Designing calm technology. *PowerGrid Journal 1*, 1.