Representations in Pervasive Computing[1]

Christopher Lueg
Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway NSW 2007, Australia
lueg@it.uts.edu.au
http://www−staff.it.uts.edu.au/~lueg/

Abstract

The idea behind pervasive computing is that embedded and invisible technology calms our lives by removing the annoyances. Everyday life, however, is shaped by what people do, how they do it, and how they perceive what they are doing. As a consequence everyday life is difficult to grasp in computational terms. A look at the pervasive computing literature indicates that in a number of pervasive computing scenarios these difficulties are addressed by assuming that it is feasible to build intelligent behavior and common−sense understanding into pervasive computing environments. We believe that such assumptions are prominent among the reasons why many pervasive computing scenarios still sound like science fiction although most of the technologies required are readily available. Making these assumptions along with known difficulties explicit would greatly help pervasive computing become part of everyday life. Example scenarios from the pervasive computing literature will be used to illustrate a number of difficulties and some of the lessons to be learned from related disciplines having investigated similar ideas before.

Introduction

Pervasive computing and ubiquitous computing are synonymous terms (Satyanarayanan 2002) referring to the vision that embedded and invisible technology calms our lives by removing the annoyances (Weiser 1991). Pervasive computing is often seen as a major evolutionary step based on ground breaking work in fields, such as distributed systems and mobile computing (e.g., Satyanarayanan 2001) and it seems that the remaining challenges are mostly technical challenges. The focus on technology is clearly reflected in, for example, what the US National Institute of Standards and Technology (NIST) sees as the synthesis of pervasive computing: "Pervasive computing is a term for the strongly emerging trend toward: numerous, casually accessible, often invisible computing devices, frequently mobile or embedded in the environment, connected to an increasingly ubiquitous network infrastructure composed of a wired core and wireless edges." (from URL http://www.nist.gov/pc2001/).

Certainly there a number of technical issues that still need to be addressed but in general the field is technically mature in the sense that most of the critical technologies are now readily available (e.g., Satyanarayanan 2002). Everyday examples are tiny cameras, powerful handheld computers and wireless LANs. However, a lot of pervasive computing scenarios still sound like science fiction although these crucial technologies are readily available. Difficulties can be attributed to some extent to the problem to ’grasp’ everyday life in well−defined, computational terms as everyday life is shaped by what people do, how they do it, and how they perceive what they are doing. A look at the pervasive computing literature indicates that in a number of pervasive computing scenarios these difficulties are addressed by assuming that it is feasible to build to some extent intelligent behavior and common−sense understanding into pervasive computing environments. We believe that these assumptions are prominent among the reasons why many pervasive computing scenarios still sound like science fiction. An explicit discussion of these assumptions and known difficulties would greatly help pervasive computing become part of everyday life.

1 Proceedings of the Inaugural Asia Pacific Forum on Pervasive Computing. Adelaide SA, Australia, 31 October − 1 November 2002.

We proceed as follows. First, we analyze a number of pervasive computing scenarios by introducing a few relatively simple changes to the scenarios. Then we discuss from a more theoretical perspective that the problems are manifestations of what is known as the frame problem in artificial intelligence. The frame problem and the related problem of building intelligent machines are under investigation for decades which means that there is a body of knowledge directly relevant to the question of what can reasonably be expected from pervasive computing technologies. Finally, we argue that lessons to be learned from the failures and successes of artificial intelligence suggest to keep humans 'in the loop' whenever usefulness of pervasive computing scenarios depends on common−sense understanding or intelligent behavior.

A Discussion of Pervasive Computing Scenarios

In this section we look at three pervasive computing scenarios to be found in the literature in order to illustrate a number of issues that may help explain why many pervasive computing scenarios still sound like science fiction.

The pervasive computing system Aura

The first example scenario is described by Satyanarayanan (2001) illustrating a number of technical issues that need to be addressed in pervasive computing. The scenario can be seen as a representative of a class of pervasive computing scenarios in which to some pervasive computing environments are required to act in an intelligent and common−sense oriented way:

> "Fred is in his office, frantically preparing for a meeting at which he will give a presentation and software demonstration. The meeting room is a 10−minute walk across campus. It is time to leave, but Fred is not quite ready. He grabs his PalmXXII wireless handheld computer and walks out of the door. Aura [a pervasive computing system] transfers the state of his work from his desktop to his handheld, and allows him to make his final edits using voice commands during his walk. Aura infers where Fred is going from his calendar and the campus location tracking service. It downloads the presentation and the demonstration software to the projection computer, and warms up the projector. Fred finishes his edits just before he enters the meeting room. As he walks in, Aura transfers his final changes to the projection computer. As the presentation proceeds, Fred is about to display a slide with highly sensitive budget information. Aura senses that this might be a mistake: the room's face detection and recognition capability indicates that there are some unfamiliar faces present. It therefore warns Fred. Realizing that Aura is right, Fred skips the slide. He moves on to other topics and ends on a high note, leaving the audience impressed by his polished presentation." (from Satyanarayanan 2001, page 12).

Many of the proposed activities, such as editing slides while walking to a meeting room, could be implemented by using readily available technology, such as noise−reduction microphones and head−mounted displays. There are a number of usability issues that need to be addressed but the lack of a supportive infrastructure seems to be a major factor preventing such activities from becoming part of everyday life. In these days even the migration of documents to presentation computers from handheld computers may be problematic (surprisingly, it still is problematic to migrate documents from one desktop computer to another as the popular Word document format is a proprietary format which means it is only supported on certain platforms. Even migrating Word documents within their native application frameworks may be problematic, as numerous different versions of the document format exist. Truly platform−independent document formats, such as postscript, have been neglected for too long but are receiving more attention due to the growing recognition that seamless document migration requires well−documented and platform−independent document formats). The Aura scenario quoted above was actually used for arguing that the most important research issue in pervasive computing is the seamless integration of component technology. The brief discussion of document migration in this section clearly supports the view that there are numerous technical issues that still need be addressed.

Apart from illustrating a number of technical issues, the Aura scenario also describes a pro−active computer system ("Aura transfers [...]", "Aura infers [...]") that autonomously implements a number of potentially wide−ranging decisions ("[Aura] downloads the presentation and the demonstration software to the projection computer [...]"). Making such decisions actually requires a considerable common−sense understanding or intelligent behavior. For example, in order to send the right slides to the right presentation computer, Aura needs to ’understand’ the connection between the slides Fred is editing and the upcoming presentation. Otherwise Aura might transfer the wrong set of slides to the projection computer as Fred might be working on slides he intends to use for a talk later in the afternoon. Relying on Fred’s schedule might be problematic, as often people do not keep their schedules updated all the time. Reasons are, for example, that keeping schedules updated all the time requires considerable effort. Also, in certain situations, employees may not want their colleagues to know every detail of their activities (see Want et al 1992 for a discussion of the latter point in the context of the active badge system). Aura might also face everyday problems such as the re−location of a meeting. The original location might have turned out to be too small for accommodating a number of colleagues unexpectedly attending the meeting. A sticky note indicating the change is placed at the door but the secretary has not yet updated the electronic room bookings. As Aura depends on accurate electronic information, such a change could cause Aura send Fred’s slides to the wrong presentation computer which would be the computer in the originally scheduled meeting room. Another everyday experience Aura needs to be able to cope with is that meetings are canceled due to unforeseen events, such as accidents or traffic jams. Fred (and Aura) might only learn about the cancellation while on their way to the meeting. At that time Aura might already have sent the presentation slides to the presentation computer, which means that Aura would need to infer that the slides would have to be removed from that computer. Given the highly sensitive information on one of the slides it is interesting anyway that Aura takes the responsibility that the information will be secure while being transferred to the projection computer and while being stored on the remote system. How does Aura know that the presentation contains highly sensitive information? Did Fred mark the documents as such or is Aura capable of sophisticated text understanding and reasoning about sensitivity of information? At issue is also the face recognition system. What if the Aura did not recognize the unknown persons − who would be responsible for the potential leaking of highly sensitive information? What does ’unfamiliar’ mean in computational terms?

The home of the future

Intille (2002) describes a pervasive computing scenario in the context of a ’home of the future’. Compared to the Aura scenario, its implementation appears to be technically straightforward:

> "One way to reduce resource consumption is to design a home environment that controls environmental conditions. The home’s occupant informs the system via some type of user interface that he or she wishes to stay comfortable while saving as much energy or money as possible. The home then uses a set of optimization algorithms to simultaneously maximize savings and comfort by automatically controlling the HVAC systems, windows, and blinds. For instance, on a day when the temperature is predicted to shift from warm to cool, the home might determine that the optimal cooling strategy is to shut down the AC and automatically open a set of blinds and windows so as to create an efficient cross breeze." (from Intille 2002, page 77)

However, the implementation of the presumably simple automation setting would require a house exhibiting significant intelligence and common−sense understanding. Intille (2002) illustrates the complexity: opening the window might be inappropriate as it might be noisy outside; someone might be smoking in front of the window to be opened; someone in the house might be allergic to pollen and the pollen count is high; opening the blind might throw glares on a computer screen, and so on. Intille (2002) concludes that it would be an immense challenge to implement this pervasive computing scenario in a real home setting.

Agent−based communication mediation

A number of different issues can be highlighted by discussing another apparently simple pervasive computing scenario described by Henricksen et al. (2002):

> "Bob has finished reviewing a paper for Alice, and wishes to share his comments with her. He instructs his communication agent to initiate a discussion with Alice. Alice is in a meeting with a student, so her agent determines on her behalf that she should not be interrupted. The agent recommends that Bob either contact Alice by email or meet with her in half an hour. Bob's agent consults his schedule, and, realizing that he is not available at the time suggested by Alice's agent, prompts Bob to compose an email on the workstation he is currently using, and then dispatches it according to the instructions of Alice's agent.
> A few minutes later, Alice's supervisor, Charles, wants to know whether the report he has requested is ready. Alice's agent decides that the query needs to be answered immediately, and suggests that Charles telephone her on her office number. Charles' agent establishes the call using the mobile phone that Charles is carrying with him." (from Henricksen et al. 2002, page 168)

This scenario is particularly interesting as the query of an employee's supervisor is put through although the recipient is in a meeting whereas a colleague's query is blocked. It seems as if the pervasive computing system relies on information about a person's position within an organization when calculating whether to interrupt the recipient or not. The problem with such an approach would be that there are more exceptions than rules to follow. Henricksen et al.'s scenario would differ significantly from a 'standard situation' if Alice were waiting for the feedback (nearby deadline for submitting the paper) or if her colleague Bob were about to leave for an extended business trip. On the other hand, Alice's supervisor Charles might have queried information about a report he would not need before end of the following week. These minor changes to the scenario would imply that despite Charles being Alice's supervisor his request might be less important than Bob's.

This brief discussion again indicates that describing in computational terms 'what matters' in a situation may be extremely difficult as considerable common−sense understanding may be required even in presumably simple situations. Of course, it would be possible to consider further information sources, such as electronic travel schedules, electronic vacancy lists, and so on but there is some evidence suggesting that more information would not change the nature of the problem, which is related to representation, change and interpretation.

Pervasive Computing and Representations

In the previous section we have analyzed pervasive computing scenarios by introducing a number of changes to the scenarios and by discussing impacts of these changes. The resulting problems can be explained from a number of different perspectives, such as logic and epistemology

From a logic−oriented perspective, the problems are manifestations of what is known as the frame problem (e.g., Pylyshyn 1987) in classical, representation−based artificial intelligence (AI). Roughly, the frame problem is about what aspects of the world would have to be included in a sufficiently detailed world model and how such a world model could be kept up−to−date when changes occur is know as the frame problem in artificial intelligence. The frame problem is under investigation for more than two decades and it seems to be reasonable to state that the frame problem is intractable in realistic settings (e.g., Dreyfus 2001). Put in a nutshell, the real world is constantly changing, intrinsically unpredictable, and infinitely rich (Pfeifer and Rademakers 1991).

The frame problem is often considered a more technical problem as it is about keeping models of the world up−to−date. However, the frame problem can also be interpreted from an epistemology−oriented point of view as such a world model defines what is known about the world. The discussion of the Aura scenario has indicated the necessity to use the model for reasoning about the state of the world and about the implications of changes (e.g., implications of cancellations and re−locations). This indicates that the frame problem is also an epistemological problem as richness of the model determines what can be inferred based on the model: aspects of the world not included in the model and not derivable from the model do not exist in the world of the model (e.g., "information sensitivity": if Aura's world model would not include some notion of information sensitivity then Aura would not be able to conclude that Fred should be warned that some slides contain highly sensitive budget information. Similarly, Aura might not be able to conclude that the presentation slides need be removed from the presentation computer as the meeting had been relocated to a different meeting room).

A closely related issue is the idea that pervasive computing systems need to be context−aware. Gupta et al. (2001), for example, argue that achieving invisibility in pervasive computing will require tremendous progress in user interfaces, context−awareness and other technologies. The idea behind context−awareness is that computational artifacts are able to sense the context in which they are being used so that they can adapt their functionality accordingly. The problem with implementing context−awareness in artifacts is that features of the world are not context because of their inherent properties. Rather, they become context *through their use* in (human) interpretation (Winograd 2001). Context is shaped by the specific activities being performed at a moment; these activities also influence what participants treat as *relevant* context (Goodwin and Duranti 1992). Agre (2001) explains that people use the various features of their physical environment as resources for the social construction of a place, i.e., it is through their ongoing, concerted effort that the place −opposed to space− comes into being. An artifact will be incapable of registering the most basic aspects of this socially constructed environment. Accordingly, a context−aware artifact may fail annoyingly as soon as a context−aware system's (wrong) choices become significant. Elsewhere (e.g., Lueg 2002a) we have outlined that context−awareness in any non−trivial sense also involves the frame problem discussed earlier in this section.

Early experiences reported in the context of the fielding of the active badge location system (Want and Hopper 1992) in a research lab can be used to illustrate some of the considerations above. The active badge location system (Want et al. 1992) was primarily used by the lab's receptionist when trying to forward phone calls to the location of a recipient's current location. Want et al. report that staff wearing badges found it useful to have phone calls accurately directed to their current location. However, staff also wanted to be able to exhibit some control over when calls were forwarded to them. Want and Hopper (1992) report that an extended version of the active badge location system allowed users to write personal control scripts that would control phone forwarding based on aspects of the environment, such as location or time.

Want and Hopper's (1992) control scripts could be seen as a way of adding a notion of context−awareness to the active badge location system (see Lueg 2002a for a discussion of context−awareness in the more technically−oriented literature). In a scenario similar to what has been outlined by Henricksen et al. (2002) the active badge control scripts could be used to block phone calls if the recipient's current location is in a meeting room unless the caller's position in the company's hierarchy is beyond the recipient's. Want et al.'s (1992) experiences with control scripts suggest, however, that it may be extremely difficult to pre−define under which conditions phone calls should be or forwarded or blocked.

Related problems have been investigated in the context of intelligent desktop agents and personal assistants (e.g., Maes 1994). Such agents were expected to take over boring, or repetitive and time−consuming tasks in order to increase human productivity and creativity (Hoyle and Lueg 1997). Examples for mundane tasks were meeting scheduling and email filtering. Promises made during the early agent hype have been reviewed and the conclusions were rather disillusioning:"[...] not much discernible progress has been made post 1994 [the year in which the ACM special issue on software agents was published], perhaps because researchers have failed to address the practical issues surrounding the development, deployment and utilization of industrial−strength production systems that use the technology. We note that once greater effort is placed on building useful systems, not just prototype exemplars, the real problems inherent in information discovery, communication, ontology, collaboration and reasoning, will begin to be addressed."

(Nwana and Ndumu 1999). Given the apparent overlap of topics under investigation in agent research and pervasive computing research (e.g., context−specific information delivery and meeting scheduling), the statement is relevant to pervasive computing research as there is also a lack of reports on experiences with pervasive computing environments. Similar to agent research, only long−term, real−world experiences with pervasive computing environments will reveal whether certain technologies work or not. User frustration is almost guaranteed if user−modeling techniques, such as statistical models of user behavior, are good at supporting 'typical' users but fail to support individual users in their specific ways of interacting with the world. We are confident, however, that the lack of reports will be addressed as a number of pervasive computing researchers are stressing the importance of real world experiences. Abowd and Mynatt (2000), for example, have argued that "[d]eeper evaluation results require real use of a system, and this, in turn, requires a deployment in an authentic setting. The scaling dimensions that characterize ubicomp systems− device, space, people, or time−make it impossible to use traditional, constrained usability laboratories". More recently, Consolvo et al. (2002) have argued that "[t]o be successful, ubicomp applications must be designed with their environment and users in mind and evaluated to confirm that they do not disrupt the users' natural workflow". Examples for recent reports on subjects interacting with pervasive computing technology in different ways than envisioned by designers of the technologies are guidebook (Fleck et al 2002) and e−graffiti (Burrell and Gay 2002). The guidebook study is interesting as it demonstrates that even in enclosed environments with presumably well−defined roles, such as museums, it is difficult to envision how technology will actually be used. The e−graffiti study nicely demonstrates how users make use of technology according to their needs even if the technology has been designed for different purposes. A similar effect had been observed by Carroll et al. (2001) when investigating youngsters using (or not using) features of mobile phones to change certain aspects of their social life, such as the need to meet as specific times at specific places. Using mobile phones allows these youngsters to meet 'on the run', ultimately fragmenting their lives. Howard et al. (2001) draw from work on the task−artifact cycle (Carroll et al. 1991) to explain this usage of mobile phones as an instance of technology appropriation.

Socially Responsible Design

Experiences with artificial intelligence techniques, such as common−sense reasoning and plan recognition, in realistic settings suggest that there is always the risk that pervasive computing technologies relying on these techniques fail when situations do not develop as expected by developers. Plans, for example, are often used by humans to guide their behavior but plans do not determine human behavior (Suchman 1987). This means that even if resources, such as plans or electronic schedules, are available they need to be interpreted and contextualized if used by pervasive computing systems like Aura. Interpretation and verification, however, require deep understanding of human behavior which means that such tasks are everything but trivial (Lueg 2002b). User−modeling techniques, such as statistical models of user behavior, may catch 'typical' user behaviors (and provide support in 'typical' situations) but they may fail annoyingly in non−typical situations.

A way to circumvent many of the problems associated with trying to develop pervasive computing systems that exhibit intelligent behavior or common−sense understanding is to keep humans 'in the loop'. As Erickson (2002) puts it: "[...] we need to consider people as part of the system. Computers detect, aggregate, and portray information, constructing "cue−texts" that people can read, interpret, and act on."

Applied to the pervasive computing scenarios discussed previously, the idea of keeping humans in the loop in order to avoid (possibly flawed) computation of decisions could be realized as follows. For example, in the Henricksen et al. (2002) scenario (interrupt callee or not) keeping users in the loop could be realized by providing to the caller information about the callee's current situation (e.g., callee in a meeting with a student) and by leaving the decision whether to interrupt or not to the caller. Then the decision would depend on the caller's understanding of the situation. At the end of the day, it is the caller who has to justify the interruption (or the missed opportunity) anyway. An implementation of a related approach has been described by Pedersen (2001).

In a second Aura scenario (Satyanarayanan 2001), the pervasive computing systems Aura needs to prioritize a bunch of emails in order to be able to send the most important ones first:

> "Jane is at Gate 23 in the Pittsburgh airport, waiting for her connecting flight. She has edited many large documents, and would like to use her wireless connection to e−mail them. Unfortunately, bandwidth is miserable because many passengers at Gates 22 and 23 are surfing the Web. Aura [a pervasive computing system] observes that at the current bandwidth Jane won’t be able to finish sending her documents before her flight departs. [...] Aura discovers that wireless bandwidth is excellent at Gate 15, and that there are no departing or arriving flights at nearby gates for half an hour. A dialog box pops up on Jane’s screen suggesting that she go to Gate 15, which is only three minutes away. It also asks her to prioritize her e−mail, so that the most critical messages are transmitted first. [...]". (from Satyanarayanan 2001, p. 12)

In this scenario, Aura asks the user to assign priorities to emails. By doing so, the designers of the scenario elegantly circumvent a number of hard problems (text understanding, relevance computation, etc.). Moreover, by keeping the user in the loop the problem is solved by the ’expert’ who probably knows best which emails are the most important ones.

Two recent examples for research projects keeping users ’in the loop’ are Stanford University’s interactive workspaces project (Johanson et al. 2002) and the already referenced ’home of the future’ project at MIT (Intille 2002). At Stanford University, researchers explore interaction with wall−size displays and outline their motivation for keeping users in the loop as follows: "Rather than have the room react to users, we chose to focus on letting users adjust he environment as they proceed with their tasks [...] users and social conventions take responsibility for actions and the system infrastructure is responsible for providing a fluid means of executing those actions" (Johanson et al. 2002, p. 68). MIT researchers express a similar view maintaining that "[...] the home of most value in the future will not use technology primarily to automatically control the environment but instead will help its occupants learn how to control the environment on their own" (Intille 2002, p. 76).

Summary

In this paper, we have analyzed a number of pervasive computing scenarios. By introducing a few relatively simple changes to the scenarios we were able to highlight the omnipresent problem of brittleness which results from the fact that everyday life is shaped by what people do, how they do it, and how they perceive what they are doing. Many pervasive computing scenarios seem to address this problem by assuming that it is feasible to build intelligent behavior and common−sense understanding into pervasive computing environments. However, after decades of research in artificial intelligence it is reasonable to assume that brittleness of such systems would not be a technical problem that will be solved sooner or later. Rather, it is likely that brittleness would remain a characteristic of such systems. Brittleness, however, would seriously impact success and overall acceptance of pervasive computing systems.

The discussion of examples illustrating the alternative approach of keeping users ’in the loop’ suggests that brittleness could often be avoided. Moreover, giving users a sense of control often supports acceptance of new technologies. In this sense, we see the future of pervasive computing in creating interactive environments and not so much in trying to create intelligent environments. We also believe that an explicit discussion of problems likely to come up when pursuing the idea of intelligent environments would help pervasive computing researchers make deliberate decisions regarding their options when planning new research projects.

References

Abowd, G. and Mynatt, E. (2000). Charting past, present, and future research of ubiquitous computing. ACM Transactions on Computer−Human Interaction Vol. 7 Number 1, pp. 29−58.

Agre, P. (2001). Changing places: contexts of awareness in computing. *Human Computer Interaction* 16.

Bellotti, V. and Edwards, K. (2001). Intelligibility and accountability: human considerations in context aware systems. *Human Computer Interaction* 16 (2−3).

Burrell, J. and Gay, G. (2002). E−Graffiti: evaluating real−world use of a context−aware system. *Interacting with Computers* 14, pp. 301−312.

Carroll, J.; Howard, S.; Vetere, F.; Peck, J., and Murphy, J. (2001). Identity, power and fragmentation in cyberspace: technology appropriation by young people. In G. Finnie, D. Cecez−Kecmanovic, and B. Lo (eds.) *Proceedings of the Twelth Australasian Conference on Information Systems* (ACIS 2001). pp. 95−102. ISBN 1−86384−483−X.

Carroll, J.; Kellogg, W. and Rosson, M. (1991). The task−artifact cycle. In J. Carroll (ed.) *Designing interaction. Psychology at the human−computer interface*. Cambridge University Press. pp 74−102.

Consolvo, S., Arnstein, L, and Franza, R. (2002). User study techniques in the design and evaluation of a ubicomp environment. *Proceedings of the Fourth International Conference on Ubiquitous Computing*, 29 September−1 October, Goeteborg, Sweden.

Erickson, T. (2002). Some problems with the notion of context−aware computing. *Communications of the ACM* Vol.45, No.2, pp. 102−104.

Fleck, M., Frid, M., Kindberg, T., O'Brien−Strain, E., Rajani, R., and Spasojevic, M. (2002). From informing to remembering: ubiquitous systems in interactive museums. *IEEE Pervasive Computing* April−June, pp. 13−21.

Goodwin, C. and Duranti, A. (1992). Rethinking context: an introduction. In Duranti, A., Goodwin, C., eds.: *Rethinking context: language as an interactive phenomenon*. Cambridge University Press.

Gupta, S., Lee, W.C., Purakayastha, A., and Srimani, P. (2001). An overview of pervasive computing. *IEEE Personal Communications* pp. 8−9.

Henricksen, K., Indulska, J., and Rakotonirainy, A. (2002). Modeling context in pervasive computing systems. In Mattern, F. and Naghshineh, M. (eds) *Proceedings of the International Conference on Pervasive Computing* 26−28 August 2002, Zurich, Switzerland. Springer LNCS 2414, pp. 167−180.

Howard, S.; Carroll, J.; Vetere, F.; Peck, J., and Murphy, J. (2001). Young people, mobile technology, and the task−artefact cycle. In W. Smith, R. Thomas and M. Apperley (eds.) *Proceedings of the Annual Conference of the Computer Human Interaction Special Interest Group of the Ergonomics Society of Australia* (OZCHI 2001). pp. 63−69. ISBN 0−7298−0504−2.

Hoyle, M. and Lueg, C. (1997). Open Sesame: A look at personal assistants. In *Proceedings of the International Conference on the Practical Application of Intelligent Agents and Multi−Agent Technology (PAAM 97)*, pp. 51−60.

Intille, S. (2002). Designing a home of the future. *IEEE Pervasive Computing* April−June, pp. 76−82.

Johanson, B, Fox, A., and Winograd, T. (2002). The interactive workspaces project: experiences with ubiquitous computing rooms. *IEEE Pervasive Computing* April−June, pp. 67−74.

Lueg, C. (2002a). Operationalizing context in context−aware artifacts: Benefits and pitfalls. *Informing Science* Volume 5 No 2, pp. 43−47. ISSN 1521−4672.

Lueg, C. (2002b). On the gap between vision and feasibility. In Mattern, F. and Naghshineh, M. (eds) *Proceedings of the International Conference on Pervasive Computing* 26−28 August 2002, Zurich, Switzerland. Springer LNCS 2414, pp. 45−57.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM* 37, pp. 31−40.

Nwana, H. and Ndumu, D. (1999). A perspective on software agents research. *Knowledge Engineering Review*.

Pedersen, E. (2001). Calls.calm: enabling caller and callee to collaborate. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems Extended Abstracts: Short Talks*.

Pylyshyn, Z.W., ed. (1987). *The robot's Dilemma: the frame problem in artificial intelligence.* Ablex Publishing Corporation, Norwood, NJ.

Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *IEEE Personal Communications* pp. 10−17.

Satyanarayanan, M. (2002). A catalyst for mobile and ubiquitous computing. *IEEE Pervasive Computing* Vol. 1, Number 1 (January−March), pp. 2−5.

Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems* 10, pp. 91−102.

Want, R. and Hopper, A.(1992). Active badges and personal interactive computing objects. *IEEE Transactions on Consumer Electronics* 38, pp. 10−20.

Weiser, M.(1991). The computer for the 21st century. *Scientific American* Vol. 265 No. 3 pp. 66 75. Reprinted in *IEEE Pervasive Computing* Vol. 1 No. 1 2002 (January−March), pp. 19−25.

Winograd, T. (2001). Architectures for context. *Human Computer Interaction* 16 (2−3).