Ask Not for Whom the Cell Phone Tolls: Some Problems with the Notion of Context-Aware Computing*

Thomas Erickson
IBM T.J. Watson Research Center

Let not the cell phone ring at the theatre. Or at the symphony during the beautifully quiet beginning of Beethoven's ninth. Or as my mother-in-law serves the first course of Thanksgiving dinner. The cell phone, and its potential for intrusion into social situations at inauspicious moments, has become a symbol of the blunt impacts of technology upon the fluid and subtle social structures within which we construct our lives.

The idea of context-aware computing [2, 4] springs from a natural desire to cushion ourselves from these infelicitous impacts of technology. It is not coincidental that the notion of context-awareness has become increasingly popular as technology has become ever more pervasive and entwined with our lives.

As an interaction designer I continually grapple with the awkward relations amongst technology, users, and context. I would like nothing more than technology that was context-aware, able to sense the situation in which it was immersed and adjust its actions appropriately. However, thus far, my experiences leave me skeptical and increasingly concerned that the phrase "context-aware" is deeply misleading.

The root of the problem is that the context-awareness exhibited by people is of a radically different order than that of computational systems. People notice a vast range of cues, both obvious and subtle, and interpret them in light of their previous experience to define their contexts. Thus, when at the theatre, we know when to talk, when to listen, when to clap, and when to leave.[1] In contrast, context-aware systems detect a very small set of cues, typically quantitative variations of the few dimensions for which they have sensors. A (hypothetical) context-aware cell phone might be able to detect that it is motionless and in a dark place with high ambient noise — but that is very different from the human awareness of being in a theatre.

If the context-awareness of systems is so different from that of humans, why use the phrase? One answer is that it serves as a useful metaphor. The phrase "context-aware" highlights an attractive and saleable characteristic of such systems: the ability to use sensors to detect and respond to features of their surroundings. From that point it is only a small step to imagining a cell phone that will refuse to ring in the theatre. Unfortunately, what is a small step for the imagination is a giant leap in implementation.

[1] While some may choose to talk on their phones at times others may find inappropriate, the problem is not that people are unaware that they're in the theatre, but rather that social norms about cell phone use have not yet congealed.

Metaphors obscure as well as highlight. And, as a metaphor, the phrase “context-aware” obscures two critical components of the systems it is used to describe. To examine this, let’s consider some examples of context-awareness gone awry:

> Spying a newspaper rack, Tom pulls his rented car to the side of the road and hops out to grab a paper. The car, recognizing that the door has closed and the engine is running, alertly locks its doors.
>
> In the midst of her finely honed closing pitch, Susan’s prospective clients watch intently as her screensaver kicks in and the carefully crafted text of her slide slowly morphs into flowing abstract shapes.
>
> ‘What an idiot,’ Roger mutters as the CEO finishes his presentation, unaware —for the moment— that the high tech speaker phone in the table’s center has triangulated on his whisper and upped its gain to broadcast his remark to the meeting’s remote audience.

These examples highlight two aspects of context-aware systems that are obscured by the metaphor. First, we are not designing context-aware systems because we believe that, somehow, context-awareness is a good thing in and of itself. Rather, the purpose of context-awareness is to allow our systems to take action autonomously. We want the car to safely lock us inside. We want our computers to preserve their battery power. We want our speaker phones to pick up our voices when we’re speaking softly. We want our systems to monitor the context, and then act appropriately, so that we don’t have to be in the control loop. Context-awareness is ultimately about action.

The second obscured aspect of context-aware systems is that the ability to recognize the context and determine the *appropriate* action requires considerable intelligence. Why is it a problem to lock the car door when a car is running? Why is it a problem to invoke the screensaver during the closing pitch of a presentation? Why is broadcasting the whisper inappropriate? Although the answers are evident to anyone over eight, they are not easy for a system to deduce. If there is one thing that we’ve learned from Artificial Intelligence, it is that understanding ordinary situations is a knowledge-intensive activity, and that common sense is (at best) awfully hard to implement.

It is an open question as to whether we can construct context-aware systems that are so robust that they will rarely, if ever, fail. I am skeptical. In lieu of true AI, the approach to making context-aware systems more robust is to add new rules. For example, I am told that today’s rental cars no longer lock the doors only when the engine is running; they will wait until the accelerator is depressed or there is a weight on the driver’s seat. While that rule may have averted my particular problem, if we assume an active dog or toddler in the car the doors may still lock at an inauspicious moment. There are two points here: First, regardless of whether adding rules solves the problem, piling heuristic upon heuristic in an attempt to map a sparse array of sensor inputs to an actionable interpretation is very different from human awareness. Second, as the set of rules becomes larger and more complex, the system becomes more difficult to understand. The consequence, I fear, is that we will find ourselves dwelling in a world where our ability to understand and control what is going around us is diminished. Using a metaphor that obscures the centrality of control and intelligence in context-aware computing will only exacerbate the problem. In many respects we’d do better with yesterday’s metaphor of intelligent agents; although that has its own problems [1], at least it left people with an appropriate sense of unease.

If we discard the assumption that computers have the intelligence to recognize contexts and then act appropriately, what are we left with? I suggest that rather than trying to take humans out of the control loop, we keep them in the loop. Computational systems are good at gathering and aggregating data; humans are good at recognizing contexts and determining what is appropriate. Let each do what they're good at. Imagine the following scenario: You call a colleague's cell phone number. Rather than getting an immediate ring, an 'answering machine' comes on the line and says, 'Lee has been motionless in a dim place with high ambient sound for the last 45 minutes — continue with call or leave a message?' Now you have some basis for making an inference about your colleague's situation and deciding whether to try to interrupt[2], furthermore, you bear some responsibility for your decision. While it is true that you may not always make the right inference, you will do a lot better than a computer. The moral is that we can have context-aware computing, but to do it well we need to make people part of the system: computers can detect, aggregate and portray information, constructing "cue-texts" that people can read, interpret and act on. In short, context-aware computing would do better to emulate the approach taken in scientific visualization rather than trying to reenact AI's attempts at natural language understanding and problem solving.

The adoption of "context-aware" as the metaphor *du jour* for technology is an unfortunate move. It is not just because it sweeps the difficult stuff under the rug, though that in itself is disturbing. It is also because, by invoking the powerful notions of "context" and "awareness", concepts which people understand very differently from the way in which they are being instantiated in context-aware systems, it opens a rather large gap between human expectations and the abilities of 'context-aware' systems. In 1981 Drew McDermott wrote an essay called "Artificial Intelligence Meets Natural Stupidity," [3] in which he took AI to task for the vast gulf between the names of its systems and their ability to perform. Two decades later his essay seems remarkably apropos.

References

1. Erickson, T. Designing Agents as if People Mattered. In *Intelligent Agents* (ed. J Bradshaw). Menlo Park, CA: AAAI Press, 1997.

2. Harter, A., Hopper, A., Steggles, P., and Webster, P. (1999) The Anatomy of a Context-Aware Application. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.

3. McDermott, D. Artificial Intelligence Meets Natural Stupidity. In *Mind Design: Philosophy, Psychology, Artificial Intelligence* (ed. J. Haugeland). Cambridge, MA: MIT Press, 1981.

4. Moran, T. and Dourish, P. (editors). Special Issue on Context-Aware Computing. *Human Computer Interaction*, Volume 16, Numbers 2-4 (entire issue). New York: ACM Press: in press, 2001.

* To appear in *Communications of the ACM*. New York: ACM Press, in press, 2001.

[2] This is an instance of a common strategy in face to face situations, as when I hover outside a colleague's office trying to discern how 'interruptable' the current activity is.