



Thomas Erickson

# Some Problems with the Notion of Context-Aware Computing

Ask not for whom the cell phone tolls.

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

The motivation for context-aware computing [3] 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 between 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, my experiences thus far 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 and integrate a vast range of cues, both obvious and subtle, and interpret them in light of their previous experience to define their context. When at the theater, we know when to talk, to listen, to clap, and to leave. In contrast, context-aware systems detect a very small set of cues, typically quantitative variations of the dimensions for which they have sensors. A (hypothetical) context-aware cell phone might be able to detect 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 theater.

If the context-awareness of systems is so different from that of humans, why even use the phrase? One answer is it serves as a useful metaphor. The phrase "context-aware" highlights an interesting 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 systems able to fit more seamlessly and fluidly into our lives (though a small step for the imagination may require a giant leap in implementation).

However, metaphors obscure as well as highlight. And, as a metaphor, the phrase "context-aware" obscures two critical components of such systems. To examine this, let's consider some examples of context-awareness gone awry:

- Spying a newsrack, Tom pulls his rented car to the side of the street and hops out to grab a paper. The car, recognizing the door has just closed and the engine is running, 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 that gradually dissolve into blackness.
- "What a cretin," Roger mutters as the CEO finishes his presen-

tation, 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 obscured by the metaphor. First, we are not designing context-aware systems because we believe this technology 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 systems to monitor the context, and then act appropriately, so 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 the ability to recognize the context and determine the appropriate action requires considerable intelligence. Why is it a problem to lock the car doors when a car is running? Why is it a problem to invoke a screen-saver during the closing pitch of a presentation? Why is broadcasting a whisper inappropriate? Although the answers are evident to anyone over eight years old, they are not easy to build into a system. If there is one thing we've learned from AI, it is that understanding ordinary

situations is a knowledge-intensive activity, and common sense is difficult to implement.

It is an open question as to whether we can construct context-aware systems so robust 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 in 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 we will dwell in a world where our ability to control, and even understand, what is going on 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,

though that has its problems, too [1].

If we discard the assumption that computers have the intelligence to recognize contexts and then act appropriately, what are we left with? I suggest 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 each is 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. Furthermore, you bear some responsibility for your decision. While it is true you may not always make the correct inference, you will do much better than a computer. The moral is we can have context-aware computing, but to do it well 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. In short, context-aware computing would do better to emulate the approach taken in scientific

visualization, than in trying to reenact AI's attempts at natural language understanding and problem solving.

To me, the adoption of context-awareness as technology's metaphor du jour is a rather unfortunate move. It is not just because it sweeps difficult stuff under the rug (though that itself is disturbing). By invoking the powerful notions of *context* and *awareness*—concepts that people understand very differently from the way they are instantiated in context-aware systems—it also opens a rather large gap between human expectations and the abilities of context-aware systems. Drew McDermott wrote an essay in 1981 called “Artificial Intelligence Meets Natural Stupidity,” in which he took the field of AI to task for the vast gulf between the names of its systems and their ability to perform [2]. Two decades later his essay seems remarkably apropos. **C**

#### REFERENCES

1. Erickson, T. Designing agents as if people mattered. *Intelligent Agents* (J. Bradshaw, Ed). 1997. AAAI Press, Menlo Park, CA.
2. McDermott, D. Artificial intelligence meets natural stupidity. *Mind Design: Philosophy, Psychology, Artificial Intelligence* (J. Haugeland, Ed.) MIT Press, Cambridge, MA, 143–160.
3. Moran, T. and Dourish, P. (Eds.) Special Issue on Context-Aware Computing. *Human-Computer Interaction* 16, 2–4 (2001). ACM New York, NY.

---

**THOMAS ERICKSON** (snowfall@acm.org) is a research staff member at IBM T.J. Watson Research Center, Hawthorne, NY.

---