



# From the Editor in Chief

Editor in Chief: M. Satyanarayanan ■ satya@cs.cmu.edu

## Challenges in Implementing a Context-Aware System

M. Satyanarayanan, Carnegie Mellon University and Intel Research Pittsburgh

A unique aspect of this third issue of *IEEE Pervasive Computing* is that it is a joint special issue with *IEEE Wireless Communications*. The two magazines used a single call for papers, a single submission process, and a combined reviewing process. Authors submitted material to the joint issue rather than to a specific magazine, and only in the final stage of the reviewing process did we assign articles to one magazine or the other. I would like to thank Mahmoud Naghshineh, editor in chief of *Wireless Communications*, for his support and cooperation in creating this issue.

This is also this magazine's first guest-edited issue. A team consisting of Maria Ebling, Guerney Hunt, and Hui Lei of IBM Research, Gregory Abowd of Georgia Tech, and Hans Gellersen of Lancaster University put together this collection of articles on *context-aware computing*. Context awareness refers to the properties of a system that make it aware of its user's state and surroundings and help it adapt its behavior accordingly. This topic is central to pervasive computing because a system that strives to be minimally intrusive must be context-aware.

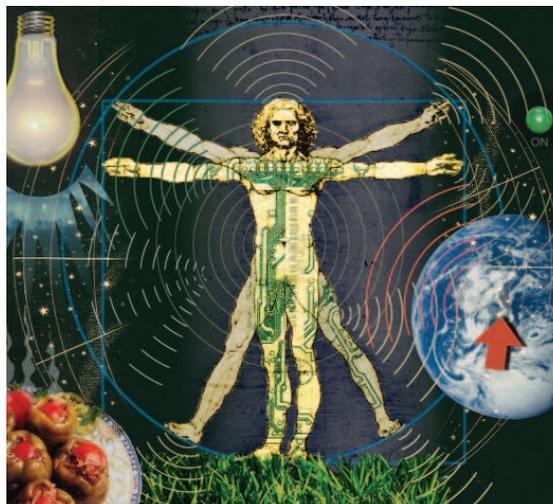
A user's context can be quite rich, consisting of attributes such as physical location, physiological state (for example, body temperature and heart rate), emotional state, personal history, daily behavioral patterns, and so on. If a human assistant were

given such context, he or she would make decisions in a proactive fashion, anticipating user needs. In making these decisions, the assistant would typically not disturb the user at inopportune moments except for an emergency. Can a pervasive computing system emulate such a human assistant?

A key challenge is obtaining the information needed to function in a context-aware manner. In some cases, the desired information might already be part of a user's personal computing space. For example, that space might include schedules, personal calendars, address books, contact lists, and to-do lists. Systems must sense more dynamic information in real time from the user's environment—such as position, orientation, people's identities, locally observable objects and actions, and emotional and physiological states.

Implementing a context-aware system requires addressing many issues:

- How does the system represent context internally? How do we combine this information with the system and application state? Where should the system store context—locally, on the network, or both? What are the relevant data structures and algorithms?
- How frequently does the system need to consult contextual information? What is the overhead of considering context? What techniques can we use to keep this overhead low?
- What are the minimal services that an environment must provide to make context awareness feasible? What are reasonable fallback positions if an environment does not provide such services?
- What are the relative merits of different location-sensing technologies? Under what circumstances should we use one and not another? Should we treat location information just like any other contextual information, or should we handle it differently? Is historical context useful?



The articles in this issue explore these and related issues. The guest editors have done a fine job of soliciting, reviewing, and shepherding these articles. It has been a pleasure to work with them, and I would like to thank them for their time and effort. ■