From the Editor-in-Chief . . .

Beyond Geography

Munindar P. Singh • singh@ncsu.edu

As the Internet extends its reach, people want to do more and more with gadgets attached to it: check e-mail messages from the airport, monitor home security systems, and set room temperatures on the way home from work. The motivation is natural: you want to keep in touch—and in *control*—from wherever you are. And who hasn't had a nagging feeling, after leaving on a long trip, of having left a back door open or the stove on?

Systems that support these kinds of functionality are starting to emerge. For instance, the first products using Bluetooth wireless technology are scheduled for market next year (http://www.bluetooth.com). However, current prototypes and product visions seem quite ad hoc. Applications are essentially modeled in a disjointed manner that loses the natural connections among them. User interfaces tend to be superficial, and changing locations is about the only subtlety underlying design ideas.

An embedding of a space governs how much of it can be observed or controlled from elsewhere.

The common theme among these upcoming applications is "action at a distance." Let me propose a simple idea for modeling this theme computationally, so that where an action originates and where the action occurs can be related systematically.

Spacing Out

Let's begin with some basic assumptions. We are given a world—a rich world populated with users and devices and humming with activity. Let's refer to a portion of the world, including its users and devices, as a *space*. Important characteristics of a space are the kinds of devices associated with it and the quality of its network connection. Thanks to devices and connectivity, users can observe (and control) spaces from other spaces.

Let's refer to the placing of spaces in the world as *embedding*. Each space has its own embedding, which reveals some interesting aspects of the space. What makes spaces interesting is that they can be based on more than just geography. When the space is based on a location, the embedding may tell us its geographical coordinates and also facts pertaining to the location. For example, in the home location, the facts would be about the security system, temperature settings, and whether a back door is open. When the space is based on an interaction—say, a dialog or e-mail exchange—the embedding gives us the user's progress in it. When the space is based on an ongoing activity, the embedding tells us things like the name and description of the activity, how far along the user is in it, and what the user plans to do.

What a user sees depends not only on the space he or she observes, but also on the space from which he or she observes. Let's refer to the view of a space from a space as a *representation*. Essentially the representing space is a window into the world and filters out some details of the viewed spaces. For example, using a home location embedding, you might be able to call in to set your room temperature, but not to disable the security system.

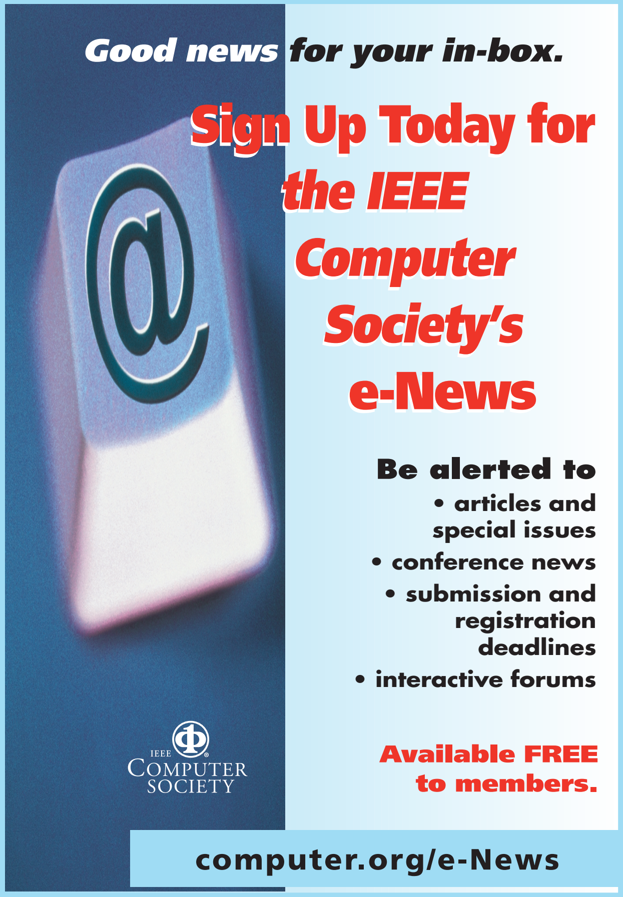
Spaces Computationally

An embedding of a space governs how much of it can be observed or controlled from elsewhere. Embeddings are based on the devices that exist in a space and how they are accessed. Think of embeddings as *computational capabilities*. Likewise, a representation governs how much of an embedding can be observed or controlled from the user's current space. Think of representations as *computational authorizations*. Representations are based on the devices that exist in a space and how their credentials are set up. Pass the given portion of the world through an embedding-representation path to compute what the user can see and control.

When you can represent a space which itself represents a space, you effectively tunnel through one space to get to another. From my workstation, I can view all lab files and read my e-mail. From my home computer, I can't do either directly. But when I telnet to my workstation, I can view all lab files and read my e-mail, effectively acquiring the authorizations of my office space. Tunneling helps us hack around the limitations of how the authorizations and capabilities are set up. Tunneling is a cheap way to achieve application integration. But it is not always desirable. For example, in the near future, I might leave my e-wallet on a server, change its settings from my desktop (wider authorizations), and use it for only small purchases from my phone (limited authorizations). Of course, tunneling from the phone to the server via the desktop would be a security risk, not to mention confusing to the user. Thus, tunneling may sometimes be essential, but would often not be desirable. However, minimizing tunneling presupposes a clean application design that includes precisely the embeddings and representations that users need.

Habitation

The space metaphor provides a useful way to partition the design requirements of applications where users are fundamentally acting at a distance. Partitioning helps us build applications that keep track of where a user is. And where a user is is not just a question of geography. ■



IEEE INTERNET COMPUTING

IEEE Computer Society Publications Office
10662 Los Vaqueros Circle, PO Box 3014
Los Alamitos, CA 90720-1314