HUMAN-COMPUTER INTERACTION, 2001, Volume 16, pp. 323–336

Mobile Devices and Mobile Data—Issues of Identity and Reference

Peter Lucas
MAYA Design

ABSTRACT

As we begin to develop architectures to guide the engineering of context-aware computing systems, we will need to apply significantly more precision to the notion of "context" than is afforded by common usage of this term. In this essay, I identify three distinct realms of contextual reference. First, the *physical context* allows us to imbue our machines with a sense of "place" in the most literal sense of that term. Second, the *device context* concerns the relations among information processing systems as such. Finally, computing systems have an *information context*. The study of information contexts is the province of the discipline of Information Architecture, which we may define as the design of information entities abstracted from the machines that process them. Although these topics raise diverse issues, they arguably all share a need for a uniform basis for dealing with matters of identity and naming. The first step in developing such a basis is the adoption of a uniform scheme for universally unique identifiers, both for identifying digital objects and for referring to physical phenomena.

Peter Lucas is a cognitive psychologist with interests at the intersection of computer architecture and product design; he is co-founder and CEO of MAYA Design, a product design consultancy.

CONTENTS

1. INTRODUCTION

Perhaps the pinnacle of 18th- and 19th-century "high technology" was the commercial sailing ship. The design and operation of these craft represented an exquisitely subtle and challenging enterprise. This challenge was rooted in the fact that these devices needed to operate simultaneously in three distinct but interacting realms. A merchant ship is at once in the ocean, in the atmosphere, and "in" a constantly flowing current of passengers and cargo. (It may seem peculiar to speak of the ships as being "in" the flow of passengers rather than the converse, but this is a perfectly sensible and useful description of the situation. From the point of view of the ship and its designers, the stream of commerce is as real and as important a context as wind or current: Its pressures shape the holds and staterooms and decks just as surely as the imperatives of hydro- and aerodynamics determine the streamlines of the hull and the catenaries of the sails.) The need to balance the often-conflicting design constraints imposed by these three contexts, combined with the reality that sailing ships spent most of their lives operating without infrastructural support, raised naval architecture to a level of art unmatched in its day. The sense of appreciation—even awe—that many people feel in the presence of a gathering of "tall ships" reflects an intuitive recognition of the success of their designers in meeting the challenges of this multiply constrained design space.

As designers of information devices, we are embarking on a great enterprise whose challenges are, in certain ways, reminiscent of those of our seafaring past. No longer can computer designers make the simplifying assumption that their creations will spend their lifetimes moored safely to the desk of a single user or plying the familiar channels of a particular local area network (LAN). It now seems clear that, before long, most computing will happen in situ and that most computing devices will find themselves operating in diverse and changing contexts. Necessarily, context awareness will soon be a hallmark of effective computation.

2. THREE REALMS OF CONTEXT

Such observations have become commonplace, but they significantly understate the issue. The thesis of this essay is that, like our naval architect predecessors, the designers of mobile computing devices will have to concern themselves simultaneously with not one, but three distinct kinds of "context" in three separate but interacting realms. These three contexts, which I discuss in turn, are (a) the physical context, (b) the device context, and (c) the information context.

Dey, Abowd, and Salber (2001 [this special issue]) provide a thorough review of previous attempts to define and categorize the notion of context, and offer their own taxonomy based on the top-level categories of "places, people, and things." The authors stipulate that their "things" category may include "either physical objects or software components and artifacts." This analysis takes a somewhat different tack in attempting to capture certain basic distinctions that are either lacking or left implicit by previous analyses. In particular, this model deviates from that of Dey et al. in two fundamental ways: First, in this analysis, "physical context" is limited to references to physical objects and places treated *as such*, and it specifically excludes information systems (including physical devices when their contextual relevance is related to their computational capabilities rather than to their physicality). Second, within the informational realm a further distinction is made between *information devices* (things that compute, whether implemented as hardware or as running computer programs), and *information* per se.

Each of these classes comprises a distinct contextual realm whose pragmatics are sufficiently diverse as to warrant separate analysis. Each realm presents a unique set of technical challenges. But they are not independent, and we treat them as such at our peril. There is a compelling need for the development of a framework within which we can begin to reason about information systems in the large—dealing at once with "context awareness" in each of these three realms.

The first step in developing any such framework is to establish a common system of identity and reference that can become the basis of formalisms and mechanism for dealing with the phenomena in question. This essay is largely limited to considering just this first step. Dey et al. correctly assert that a useful notion of identity "has to be unique in the namespace that is used by the [interoperating] applications." In this essay, I take the radical, but ultimately necessary, position that there is only one space of interoperating applications; therefore, at the end of the day there must be only one identity space. Until we come to terms with the implications of this simple claim, we will remain doomed to islands of context and systems that cannot interoperate. In the realm of contextual computing, the maturity of our thinking about issues of identity varies across the different senses of "context." The more general issue of integrating these systems into a coherent whole is largely unstudied. The following sections provide a brief synopsis of each of the three senses of "context," along with some general observations concerning issues of identity and reference for each.

2.1. Physical Context

The physical context is the first and most obvious of our three contexts of computation. It is perhaps what is most commonly meant when the term *context aware* is encountered in the literature. It is about imbuing our devices with a "sense of place" by the most literal interpretation of that phrase. Even here the notion of context is not simple. We can mean many things by *physical context.* Most obvious, perhaps, is geographic location. Here we are heir to a rich and extremely sophisticated body of work inherited from the disciplines of cartography and navigation. As the etymology of the word *geometry* demonstrates, we have been measuring the earth for a very long time. Consequently, we are not lacking for well-developed standards for denoting locations on the earth's surface. Indeed, there is an embarrassment of riches in this regard: The novice soon discovers that referencing a spot on the earth is no simple matter of latitude and longitude. Various geographic, geodetic, and geocentric coordinate systems are in common use—each optimized for a different purpose. Although practice in this area can be extremely complex, it is for the most part well defined, with precise—if not always simple—mathematics defining the relations involved. The field of geographic information systems (GIS) is quite mature, and there exist comprehensive (if sometimes overly complex) standards for the machine representation and interchange of geographic information of all sorts.

However, as we move beyond mere geolocation to the more interesting problem of denoting geographic *features* (whether natural, man-made, or political), the situation rapidly becomes murky. Suppose, to pick one of the easier cases, you and I wish to refer in an unambiguous way to, say, Hanover, PA. Should we simply use the place name? If so, we will have to devise a way to indicate whether we mean the Hanover near Wilkes-Barre, the one by Allentown, or the one at York (this is assuming we don't mean Hanover Green, Hanover Junction, Hanoverdale, or Hanoverville). We could use latitude/longitude, but what point on the earth, exactly, should we choose to represent Hanover? There is a U.S. Board on Geographic Names that sanctions official U.S. place names, but it treats U.S. and foreign names differently. For example, web-accessible resources from this source present unique numerical identifiers for foreign but not domestic places (USGS, 2001). The Getty Research Institute has compiled an extremely thorough thesaurus of geographic names

(Getty Research Institute, 2000) that is both worldwide and hierarchical, and also includes unique identifiers. However, this work is not in the public domain and is not generally recognized as authoritative. Thus, we see that the ability to unambiguously reference place and feature names—perhaps the most basic of denotational issues concerning geographical context—remains an unsolved problem.

If we look at other issues, such as denoting street addresses, we discover similar situations. There is no shortage of schemes for encoding these kinds of data, and there exist many high-quality databases of such information. However, each such database is a referential island, defining a local name space adequate to support a particular application, but nearly useless for enabling the kinds of large-scale, open information spaces implied by the context-aware computing agenda. Moreover, the denotation of physical phenomena is only the first, most basic step in developing an identity space to support place-aware computing. We also face the need for standard ontologies of place types, various distance metrics, and many other increasingly semantic matters. Much relevant work exists in the GIS and geoinformatics literature. This work, much of which has been on the periphery of mainstream computing, is poised to take on new significance as computers venture out into the real world.

But there is much more to "place" than mere geography. Human artifice has structured space in many complex ways, often creating new denotational challenges. Consider a large skyscraper. It may have 100 floors, each having 100 rooms. Already today, each room likely contains multiple networked processors—smoke detectors, door locks, building alarms, lighting controllers, thermostats. A single building may easily contain tens of thousands of embedded computers in all. To do its job, each of these processors must in some sense be "aware" of its location. But location means different things to different devices. To some it is room number, to others it is floor, distance from a fire exit, topological location on a LAN, or proximity to a window. The state of the art in such systems is such that each device is imbued with its requisite "address" by an installer on a ladder—often keeping paper records of his or her progress. Standardization efforts are underway within the relevant industry trade groups, but they are proceeding with little thought to how such standards might fit into a larger ecology of context-aware computing.

There is little need to point out that as complex as such systems may be, they are certain to pale in comparison to what is to come. The marginal cost of adding modest computing power to manufactured goods is rapidly approaching zero. It will not be long before the tens of thousands of processors in each office building become tens of millions, and it will not end there. When light switches built into modular room dividers are expected to turn on neighboring lights, how will they determine what lights are neighboring? When sensors in office chairs are able to identify their occupants by weight, how will this information be made available to nearby devices, and how will "nearby" be defined and determined? These questions raise many difficult issues, but any comprehensive answer to them must surely begin with a common architecture for identity and locational reference.

2.2. Device Context

Just as various kinds of sensory apparatus—global positioning satellite receivers, proximity sensors, and so forth—are the means by which mobile devices will become geographically aware, another class of sensors makes it possible for devices to become aware of *each other*. The superficial similarity between such sensors and ordinary communications channels belies their significance. There is a fundamental difference between the mere ability to transfer data between two or more devices along preconfigured channels and the ability of a device to discover the presence of its peers and to autonomously establish such channels without the aid of some external designer.

This, the so-called device discovery problem, forms the basis for a kind of context awareness that is different from that based on physical context. (The "discovery mechanism" of the Context Toolkit as described by Dey et al. should, as they suggest, be thought of as a placeholder for a scalable device discovery architecture.) A cluster of vehicles driving in convoy across the country forms a persistent context that is quite distinct from that of the countryside rolling by. Peer-to-peer communication in such circumstances is largely uncharted territory. If we ignore such rudimentary communications channels as horns, turn signals, and CB radios (which in any event involve humans in the loop), no direct communication occurs among today's vehicles. In fact, one must look to fairly exotic systems (such as military friend-or-foe detection systems) to find *any* interesting instance of such intervehicle communications.

Even if we broaden our search to include all information-processing devices, the pickings are slim. It is surprisingly difficult to find significant instances of peer-to-peer communications that do not involve the mediation of fixed infrastructure. Anyone who has witnessed the often comic antics of two technically adept laptop users attempt to establish direct communications between their machines without the use of external media will appreciate just how primitive is the state of the art. Despite a 7-year effort by an industry consortium of over 150 companies to define the Infrared Data Association (IrDA; 2000) infrared port as a standard for such uses, and despite the fact that essentially all modern laptop computers come equipped with IrDA ports, successful attempts to use this technology in practical situations for peer-to-peer networking are few and far between. (An exception to this is the successful use of "beaming" among Palm Computing devices and other PDAs, but even here most real-life practical use is limited to devices running similar system software.)

More recently, the Bluetooth™ short-range RF initiative represents a far more sophisticated approach to the problem of ad hoc networking. With its so-called "piconet" architecture, Bluetooth promises to address a number of issues in the areas of device discovery and network self-configuration (Miller, 1999)—issues that are in the critical path of practical peer-to-peer networking. However, the complexity and unfamiliarity of the issues involved have no doubt contributed to the many delays and false starts that have plagued this initiative.

Sooner or later, the physical ability to establish practical zero-infrastructure ad hoc networks will be achieved. But this in itself does not get us very far toward a true realm of mobile devices. Let's assume, for instance, that you and I are riding a New York subway along with 80 strangers. Let's further assume (this being the near future) that everyone on board has in his or her pocket a radio-equipped PDA physically able to communicate with each of the others, and each device has succeeded in discovering the presence of the other 81. So, what do we do now? Even ignoring engineering issues, how will my device know to cooperate with yours while remaining wary of the 80 strangers? How will these devices even be identified? More basically, what exactly will constitute identity for purposes of such policy?

Right now, my Palm Pilot has two different identities. The first is a unique hardware serial number that was assigned to it at the factory. The other one is the string "Peter Lucas" that became associated with the device on occasion of its first Hotsync with my desktop computer. Neither of these identifiers seems quite right. The risk of a chance encounter with another Peter Lucas is obviously too high for the latter to be depended on in a scalable way. On the other hand, the device serial number, although perhaps sufficiently unique, doesn't really identify the right thing. I have owned four different Palm devices so far, and the trend is likely to continue. It would be good if the social life of my PDA did not have to begin from scratch on every upgrade or repair. It is not the device that is relevant for this purpose, but rather the *role* of "Peter's PDA," whatever device happens to be playing that role at the moment. On the other hand, suppose I owned a "personal diagnostic assistant"—a device whose job is to invisibly monitor the health of my PDA and all my other electronic "stuff" (a device that is likely to soon become a practical necessity). For this purpose, the hardware ID of the particular PDA device itself would be just the ticket. As this simple example illustrates, establishing a universal device identity space will involve some subtle distinctions that are largely unaddressed in current art. Once again, however, getting such distinctions right in a universally accepted way seems to be a prerequisite to any general architecture for device context.

Before leaving this topic, we should note that device contexts are not necessarily defined by physical proximity. Devices employing a common radio channel may share a context even at great distances. In the case of wired networks, topological distances are typically (but not always) more relevant than topographic proximity. Many other kinds of "distance" (e.g., how far am I from an Internet connection—and at what cost?) will have importance in specific situations. In general, what it means to define a "space" is precisely to define such distance measures. Building a true "Cyberspace" will in large part be a process of coming to agreement on such matters.

Finally, there is the issue of mobile code. The vision—most fully articulated in SUN's® Java™/Jini™ thin client architecture (Arnold, O'Sullivan, Scheifler, Waldo, & Wollrath, 1999)—of small bits of behavior moving freely from device to device raises many issues of device context awareness. Strictly speaking, although code is not a device, but simply data (and therefore the subject of the following section), the running computer programs to which such code gives rise *are* in effect devices. As such, they exist in contexts, raising many of the same issues of context awareness as do physical devices, as well as a few new ones: On what kind of machine am I executing? Who am I sharing it with? What resources are available to me and what restrictions must I observe? These are matters of context awareness no less than the more obvious ones of physical context.

2.3 Information Context

The third—and least discussed—of our three contexts is the information context in which computation takes place. The study of such contexts is the province of the discipline of *Information Architecture,* which we may define as the design of information entities abstracted from the machines that process them. This topic has not received the attention that it deserves. The reason, I think, is rooted in the long habit of thinking of data as residing "in" our computers. In a pervasive computing world, it is useful to take quite the opposite perspective—increasingly, data are not "in" our devices any more than a phone call is "in" a cell phone. In this regard, the nautical analogy with which this essay began is particularly evocative, suggesting a future in which computing devices float freely in a vast sea of data objects—objects whose existence and identity are quite distinct from those of the devices that process them. In this world, the data have been liberated from the devices and have claimed center stage.

This is a provocative image, one that is compatible with the often stated ideal of the computer receding into the background (e.g., Norman, 1998). From this view computing devices are seen as merely transducers—a sort of perceptual prosthesis. We need them to see and manipulate data just as we need special goggles to see infrared light. In both cases, it is the perception that is of interest, not the mediating device. People are beginning to talk about a "cloud of information" following us around as we move from place to place and from device to device. But what, exactly, *is* this cloud? What is it made of? What are its properties? How will it know to follow us? How can we make it usable? These are strange and difficult questions, but beginning to view computing from an information-centric perspective is a first step in dispelling the strangeness.

The World Wide Web comprises a single, vast information space, a space defined not by the computers that implement it, but rather by the network of interlinked “pages” that are only incidentally related to the machines that house them. We could replace every server and router in the world, but this in itself would in no way change the information space that is the Web. High-volume Web sites routinely employ redundant servers in which pages are replicated across physically and even geographically distributed systems. Where are such pages “located”? The only meaningful answer to the question is that they are located in Cyberspace. Similarly, we don’t (in general) store Web pages on our desktop computers. The pages flow “through” the client machine, but they are never meaningfully “in” it.

But what happens when we step beyond the relatively simple client–server model that dominates the Web today? Already, PDA users routinely carry around replicates of Web pages that are automatically cached during periodic desktop synchronizations. More to the point, these pages can be beamed from PDA to PDA, taking on an existence (and a *per*sistence) that is quite independent of the server machines of their origin. Services exist that extend this concept to other kinds of data, allowing appointments, To-do lists, and even editable text documents to exist as coordinated replicates on any number of PDAs and desktops simultaneously. This mode of operation is presently the exception rather than the rule. But the day is not far away when pocket-sized computers will routinely come with gigabytes of storage. As such devices are released by the millions into the wild, it seems inevitable that widespread data replication will rapidly become the norm. Popular Web pages and other data objects will be replicated countless times as they flow from device to device. Explicit backup utilities will disappear—replaced by the inherent redundancy of the infosphere. Corrupted or momentarily unneeded data will be casually deleted from local devices by users confident in their ability to grab fresh copies out of the ether when needed. Damaged devices will be replaced with a shrug: The data stream will soon replenish each emptied vessel. In such a world, no one will be tempted to think of the data as being in the machines.

This is not to suggest that all data will be available everywhere. Contrary to the predictions that all computing devices will someday have continuous Internet access, I assume that connectivity will *always* be intermittent, and there will *always* be devices that are too small and cheap to support direct continuous access. Storage capacities, although huge, will always be finite and unevenly distributed. These and a variety of other factors, ranging from intellectual property issues to bandwidth and absolute connectivity limitations, will guarantee that only a small subset of the aggregate data space will be immediately available to any given device at any given time. The sea of data will be far from homogenous. Natural patterns of human and machine interaction, as well as deliberate data replication, will create currents and eddys of dataflow within the larger sea. Thus, devices will find themselves at any given time in a "context" of the third kind, and awareness of that context will involve a whole new set of questions: What information "objects" are floating around in my immediate neighborhood? Which devices contain them? Are they in a form that I understand? What are the transactions going on among devices for which these information objects serve as currency? In the long run, this kind of "context awareness" may prove to be the most basic of all.

3. THE CASE FOR UNIVERSAL IDENTIFIERS

In even so cursory an examination of our three contexts as this one, we have touched on issues of great diversity. Each of the three topics represents a major research agenda in its own right. What they have in common, I argue, is the need for a uniform basis for dealing with matters of identity and naming. Before I can ask the question "What things am I near?" I must have some clear idea of what constitutes a "thing" and how I might refer to one. To discuss this matter, we must indulge in a bit of ontological philosophizing.

In the physical world, identity comes for free. Because the laws of physics prohibit two things from being in the same place at the same time, every object has a unique coordinate in space– time. Two things can be identical, but there can be no ambiguity that they are distinct. In other words, the identity of each physical object is *intrinsic* to that object. We may find it useful to assign some kind of identifier to a physical object (such as the unique ID in my Palm Pilot), but this is essentially just a name—a denotational device to make it easier for us to refer to specific objects.

Data objects are a different matter. Consisting of patterns instead of atoms, they are not subject to the same rules as physical objects. In particular, their identity is, in a sense, artificial. We are free to define a data "thing" in any way we find useful. We may well decide, as several of the examples discussed earlier illustrate, that the same "thing" can be in more than one place at one time. That is, we may allow *replication*. A replicate is different from a copy in that copies have separate identities, whereas replicates (by definition) have the *same* identity. For example, suppose I move a text file containing the draft of a letter from my office's shared file server onto my laptop and carry it onto an airplane. In doing this, I may have had one of two intentions in mind: Perhaps I planned to edit the draft into its final form. On the other hand, I may have simply wanted a *copy* of the text to use as boilerplate for a completely new letter to a different recipient. In the first case, my intention is for my airborne edits to eventually propagate back to the original version on the file server (and we may have a problem if my assistant edits the same document in my absence). In the second case, however, the copy has assumed a new identity with no continuing tie back to the original. Note, however, that my physical actions are the same in both instances. Only my *intentions* distinguish the two cases. Most contemporary computing systems provide no structural way to denote this essential distinction. In other words, they do not distinguish *copying* from *replication.* As long as this remains the case, the very notion of the identity of data objects will remain poorly defined.

The solution to this problem is well known. It is technically rather trivial, but politically nearly intractable. That solution is the use of Universally Unique Identifiers (UUIDs) as a standard definition of data object identity. A UUID is simply a string of bytes that, within reasonable engineering tolerances, is unique in the universe. This sounds difficult, but it is not. Nor does it require the active assistance of any central authority. For all practical purposes, a sufficiently long (e.g., 16 bytes) random number would suffice. In practice, a number of slightly more systematic schemes for generating unique identifiers are in common use. They essentially fall into two categories: (a) time/network-address schemes and (b) hierarchical authority schemes.

In the former case, identifiers are generated on demand by concatenating the current time (as defined by a local system clock) to the host's IEEE 802 unique network address (typically available in ROM on any machine with a standard Ethernet network adapter). As a result, each identifier is unique in both space and time. This scheme is in wide use in a number of transaction-processing systems, and is also used by Microsoft® to generate unique identities for COM objects within Windows®.

Hierarchical authority schemes depend on the fact that anyone having exclusive control over a single UUID can, in effect, use it as an authority to mint an unbounded number of additional UUIDs simply by extending it with a locally unique identifier. This process may be extended indefinitely. Thus, some central authority need only "seed" the world with unique top-level authority identifiers, and lower tiers of authority can self-define, simply by acquiring a single UUID from a higher level authority. An infrastructure for implementing this scheme is maintained by OSI in support of a communications protocol known as ASN.1 (Larmouth, 1996), although the resulting UUIDs can be used for other purposes. Other examples include the familiar domain name system of the Internet and the Handle System/Digital Object Identifiers (Lannom, 1999) developed by CNRI primarily to support distributed document management.

As the aforementioned incomplete summary indicates, there exist several semicompeting UUID schemes, thus raising questions about the appropriateness of the label "universal." Moreover, there are many important existing "local" identifier schemes in common use that, although they don't pretend to be "universal," are authoritative within some domain. Examples of the latter include the ISBN and ISSN numbers in the publishing industry, and the various place-name schemes reviewed earlier. If we are to achieve a single universal identity space of digital objects, then these conflicts must be somehow reconciled. Further, it is unrealistic to expect that any of the existing schemes will be abandoned in favor of some new, universal scheme, however appealing this might be in the abstract. Finally, different domains have different pragmatic requirements for such secondary characteristics as terseness versus human readability, self-validation, and so forth.

Fortunately, however, even the seemingly daunting task of unifying existing legacy standards is not particularly difficult, at least technically. By "left-extending" existing ID schemes with a short, centrally administered (presumably by some standards body such as ISO) "scheme identifier," any number of local identity schemes could be made universally unique with a minimum of administrative or technical overhead. The details of such a scheme are beyond the scope of this essay, but suffice it to say that the challenges are more political than technical.

The adoption of such a framework for universal identity would be a major milestone in the evolution of ubiquitous computing. Every system architect knows that the first step in ensuring interoperability between two systems is the establishment of a common name space between the systems. If names can collide, interoperability will fail. If we have two namespaces, we have two systems. With one namespace, at least the foundation has been set for a single system. It is quite realistic to envision a world in which every geographic entity, every device, every user, every document, even every assertion is dependably and unambiguously identified using a single universal system of digital identifiers. If we can achieve a universal identity space, we will have set the foundation for universal interoperability. It is difficult to see how we could make much progress toward this goal otherwise.

Finally, let me emphasize that we are dealing here with identity in only the most basic of senses. UUIDs are indeed a magic bullet—the problem they solve is crucial, but it is very narrow. Specifically, they establish a single, genuinely universal identity space, and thus form the basis of a single, universal web of reference. Any distinct phenomenon or assertion about a phenomenon—whether physical, conceptual, categorical, or hypothetical—can be unambiguously denoted. What they do *not* do is magically solve any of the intrinsically difficult ontological or epistemological problems that inhere in any rich representational system. Returning to our earlier example, they can indeed permit us to unambiguously denote "Hanover, PA," but only if we have a clear and shared understanding of what exactly we mean by "Hanover, PA." Do we mean the corporate entity? The contiguous plot of land defined by the corporate boundaries? The collection of people whose legal residence are within those boundaries? Or, perhaps we mean the name "Hanover, PA" itself. Each of these may need to be represented and denoted, and each may be assigned a UUID. Doing so will once and for all force clarity on our references to these various concepts. But it will not, in itself, decide for us which one we mean.

4. WHAT ABOUT THE USER?

Conspicuously absent in all of the earlier is a discussion of the role of the user. Surely humans will represent the most important "context" of all for both devices and data. Where do they fit into our "three contexts" scheme? In fact, they form a curious cut through the entire space. On one hand, people are clearly part of the physical context. Awareness of who is present and of their needs, interests, and characteristics will surely be a hallmark of successful "smart spaces." On the other hand, it is often useful to model the user as an "information device"—an architectural peer to the computers, participating in the worldwide dataflow as a source and sink of data like any other system component. Yet again, we must acknowledge the user as a teleological force—the source of purpose and meaning for the system, guiding it from the outside.

All of these points of view have validity. But most important, we must consider the human in the loop from the perspective of system usability. We are on the verge of building systems unprecedented both in their scale and in their very nature. It is one thing to design a usable computer program. It is quite another to design a usable *environment* when that environment comprises innumerable semi-autonomous devices mediating an unbounded swirl of constantly flowing information. Usability, or the lack thereof, will be an emergent property of such a milieu. How does one design an emergent property? The answer is not at all clear. But it is a good guess that the place to start is in defining a basic ontology of existence in this strange new world, with an eye toward imbuing that existence with a consistency of behavior that in some measure approximates the consistency that we get "for free" in the physical world. No amount of clever hacking will achieve this goal unless it is layered on top of a common model of identity and reference.

NOTES

Author's Present Address. Peter Lucas, MAYA Design Group, Inc., 2100 Wharton Street, Suite #702, Pittsburgh, PA 15203. E-mail: lucas@maya.com

HCI Editorial Record. First manuscript received January 3, 2001. Accepted by Tom Moran and Paul Dourish. Final manuscript received March 2, 2001. — *Editor*

REFERENCES

Arnold, K., O'Sullivan, B., Scheifler, R., Waldo, J., & Wollrath, A. (1999). *The Jini™ specification*. Addison-Wesley.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction, 16,* 97–166. [this special issue]

Getty Research Institute. (2000). *Getty thesaurus of geographic names*. Available: http://www.getty.edu/research/tools/vocabulary/tgn/index.html

Infrared Data Association (IrDA). (2000). Available: http://www.irda.org

Lannom, L. (1999, April). *Handle system overview* (ICSTI Forum, No. 30).

Larmouth, J. (1996). *Understanding OSI*. London: Thompson Computer Press.

Miller, B. (1999). *Mapping salutation architecture APIs to Bluetooth service discovery layer* (Bluetooth White Paper No. 1.C.118/1.0). Santa Clara, CA: Bluetooth Special Interest Group.

Norman, D. A. (1998). *Invisible computer: Why good products can fail, the personal computer is so complex and information appliances are the solution*. Cambridge, MA: MIT Press.

USGS. (2001). *U.S. Board on geographic names*. Available: http://mapping.usgs.gov/www/gnis/