The Design of Implicit Interactions: Making Interactive Systems Less Obnoxious

Wendy Ju and Larry Leifer

Introduction

Imagine, for a second, a doorman who behaves as automatic doors do. He does not acknowledge you when you approach or pass by. He gives no hint which door can or will open—until you wander within six feet of the door, whereupon he flings the door wide open. If you arrived after hours, you might stand in front of the doors for awhile before you realize that the doors are locked, because the doorman's blank stare gives no clue.

If you met such a doorman, you might suspect psychosis. And yet this behavior is typical of our day-to-day interactions not only with automatic doors, but any number of interactive devices. Our cell phones ring loudly, even though we are clearly in a movie theatre. Our alarm clocks forget to go off if we do not set them to, even if we've been getting up at the same time for years. Our computers interrupt presentations to let everyone know that a software update is available. The infiltration of computer technologies into everyday life has brought these interaction crises to a head. As Neil Gershenfeld observes, "There's a very real sense in which the things around us are infringing a new kind of right that has not needed protection until now. We're spending more and more time responding to the demands of machines."[1]

These problematic interactions are symptoms of our as-yet lack of sophistication in designing interactions that do not constantly demand the input or attention of the user. "Implicit interactions"—those that occur without the explicit behest or awareness of the user—will become increasingly important as human-computer interactions extend beyond the desktop computer into new arenas; arenas such as the automobile, where the driver is physically, socially, or cognitively engaged. Traditional HCI—that involving a command-based or graphical user interface-based paradigm—has focused on the realm of "explicit interactions," where the use of computers and interactive products relies on explicit input and output. The values and principles that govern good desktop computing interactions may not apply when we apply computing to the products that populate the rest of our lives.

1 Neil Gershenfeld, *When Things Start to Think* (New York: Henry Holt, 1999), 102.

We humans have an abundance of experience in implicit interactions. We successfully employ them in our daily interactions without conscious thought: we modulate our speaking volume based on ambient noise level, use smaller words when explaining things to children, and hold the door open for others when we see that their arms are full. These accommodations do much to smooth our day-to-day interactions with one another, and yet are made without an explicit command.[2] By understanding how implicit interactions between humans help to manage attention, govern expectations, and decrease cognitive load; we are able to cross-apply solutions from one domain to another.

In this article, we present a framework for implicit interactions to enable human-computer interaction researchers and designers to understand the ways in which implicit interactions are distinct from explicit interactions, and to provide guidance on when different types of implicit interactions are useful. We also introduce the use of implicit interaction patterns and analogues as a design methodology. This method leverages the domain-independence of the implicit interaction framework to enable interaction designers to draw generalizations about interaction technique across application domains. Together, the implicit interaction framework and its associated methodology lay the groundwork for an emerging area of applied design research[3] focused on improving the interactions between people and computer-based systems embedded in the world.

Approach

By outlining a design method that is useful in creating a broad class of interactions, we seek to complement technology-based approaches (which focus, for instance, on sensors and architectures that enable implicit interaction), or analysis-based approaches (which would investigate implicit interaction through studies and controlled experiments) towards implicit interaction design. This design-based approach has two main objectives: to be "generative"—that is, to guide designers in a constructive fashion in designing implicit interactions—and to be "generalizable"—that is, to suggest techniques and methods that are applicable to interaction designers working on a wide array of ubiquitous computing scenarios. Just as toolkits provide a common architecture and library for software developers working on similar classes of applications,[4] we want the implicit interaction framework and methodology to help designers generate designs for similar types of interactions.

Our approach differs from that taken by many researchers working in the areas of ubiquitous computing. The usual approach is to use ethnography and contextual inquiry techniques to characterize the ways in which the specific domain in question is unique, and then to use some logic or reasoning system to deploy this domain-specific knowledge. Such solutions to knowing when the cell phone should vibrate silently, or when the alarm clock should chime, focus

2 Jakob Nielsen, "Non-Command User Interfaces," *Communications of the ACM* 36 (April 1993): 83–99.

3 Richard Buchanan, "Design Research and the New Learning," *Design Issues* 17:4 (2001): 3–23.

4 Brad Meyers, Scott Hudson, and Randy Pausch, "Past, Present, and Future of User Interface Software Tools," *Transactions on Computer-Human Interactions* 7:1 (2000): 3–28.

on solving these problems by making devices "smarter." While this approach is generative, it is rarely generalizable because the expert knowledge of how to behave in one situation does not translate to any other. But this absorption with modeling human intelligence gives short shrift to the richness of human interactions. It focuses on being "logical" rather than "courteous." What if our true talent as human interactants is less a wealth of situation-specific intelligence and more a measure of situation-independent suave?

At the other end of the spectrum is the surplus of design principles that aim to achieve implicit interaction through platitude. Cooper and Reimann's "About Face 2.0," for example, provides the following guidance for designing considerate software: "Considerate software takes an interest. Considerate software is deferential. Considerate software is forthcoming.... Considerate software doesn't ask a lot of questions. Considerate software takes responsibility. Considerate software knows when to bend the rules."[5] This is not bad advice—it certainly is general enough—but these guidelines do not actually help designers determine when an interactive system should take an interest, and when it should not ask a lot of questions. It is important to provide a vocabulary and an approach that allows designers to more easily reason about what degree of implicitness or explicitness is desired in the situation they are designing, and to hypothesize how they might create the appropriate experience.

A Framework for Characterizing Implicit Interactions

This framework models interactions as the exchange between a person (sometimes called the user or actor) and a machine (sometimes referred to as the computer, robot, or, more generically, the system). This is limited to describing dyadic relations, but provides a useful basis for modeling basic interactions.

Implicit interactions enable communication without using explicit input or output. One way that an interaction can be implicit is if the exchange occurs outside the attentional foreground of the user. This occurs in traditional computing—when the computer auto-saves your files, or filters your spam e-mail, for instance—as well as in ubiquitous computing interaction. The other way that an interaction can be made implicit is if the exchange is initiated by the computer system rather than by the user—if the computer alerts you to new mail, or when it displays a screensaver. (It may seem counter-intuitive that something that grabs your attention could be implicit, but it is important to remember that the interaction is based on an implied demand for information or action, not an explicit one.)

5 Alan Cooper and Robert Reimann, *About Face 2.0: The Essentials of Interaction Design* (Indianapolis, IN: Wiley, 2003), 184.

The implicit interaction framework (Figure 1) divides the space of possible interactions along the axes of attentional demand and initiative. Attentional demand is the attention demanded of the user by the computer system. Interactions that demand the users attention are *foreground interactions*, and interactions that elude the user's attention are *background interactions*. Initiative is an indicator of who—and to what degree—is initiating an interaction. Interactions initiated by the user are *reactive interactions*, and interactions initiated by the system are *proactive interactions*. By characterizing interactions in this way, we are able to generalize about the capabilities and features of whole classes of interactions in a domain-independent fashion.

The following are descriptions of interactions typified by each quadrant:

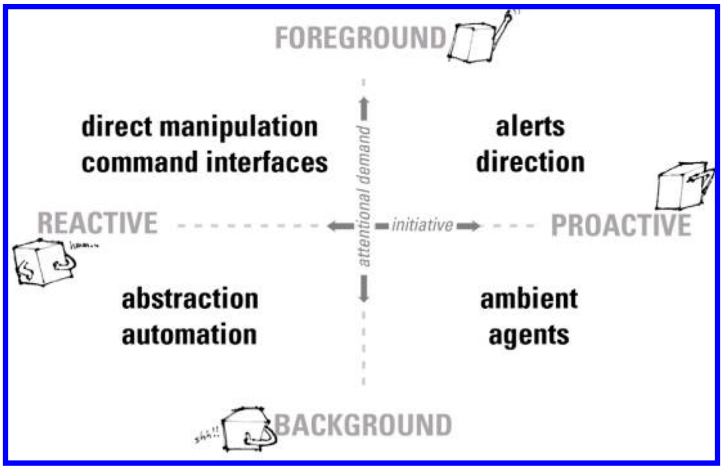

Figure 1
The Implicit Interaction Framework shows the range of interactive system behaviors.

The Framework in Action

To better understand the range of implicit interactions, let us consider this example: Our friend Terry sends us a link to a funny animation that can be found online. To play the animation, we need a Macromedia® Flash plug-in installed on our computer. The following cases show different ways that the plug-in may be installed:

> **CASE 1:** We see that the animation does not work. We deduce that we need the plug-in. We find, download, and install the plug-in.

This is a classic example of explicit interaction. This is far from a unilateral activity on our part, because the computer is involved throughout this process, but we are actively engaged in diagnosing, deciding, and performing each step along the way.

CASE 2: We see that the animation does not work. We deduce that we need the plug-in, and ask the Web browser to find, download, and install the plug-in.

CASE 3: Our Web browser shows that our animation does not work because we are missing a plug-in. We find, download, and install the plug-in.

The second and third cases highlight the different ways interactions can be implicit. In case 2, we actively perform the task of problem observation and diagnosis, but the individual steps of getting the plug-in installed are abstracted away so we don't have to attend to each step. In case 3, the browser proactively identifies the problem and suggests a solution, although we have to go through the steps to implement it.

Case 2 is an example of *abstraction;* the plug-in installation occurs in the background, so that we don't have to actively and explicitly perform each step. Case 3 is an example of *alert,* where the interaction is implicit in that the system proactively diagnosed and informed me of the need for the plug-in. These cases illustrate how attentional demand and initiative affect the implicitness of the interaction.

CASE 4: Our Web browser shows us that our animation does not work and offers to find, download, and install the plug-in. We accept the offer, and the plug-in is installed.

CASE 5: Our Web browser sees that we are trying to play an animation that we do not have the plug-in for, and lets us know that it is automatically finding, downloading, and installing the plug-in.

CASE 6: Our Web browser sees that we are trying to play an animation that we do not have the plug-in for, and automatically finds, downloads, and installs the plug-in in a background process.

These three cases show increasing degrees of proactivity and *presumption* on the part of the Web browser, and decreasing degrees of attentional demand. In case 4, there is a fair amount of demand on our attention because we need to actively accept an offer. In cases 5 and 6, the plug-in is installed without any activity on our part, but the last case is more implicit because no feedback is offered. Although our actions in both cases are the same, case 6 is more presumptuous because we do not have the opportunity to oversee and possibly cancel the task.

CASE 7: Our Web browser anticipated that we might want to play a Flash animation someday, and already has downloaded and installed the plug-in.

This last case is the most implicit interaction. In fact, with so much presumption and so little visibility, this last interaction may hardly be considered an interaction at all, since there is no activity or awareness on our part.

There is a range of ways to accomplish the task of installing the Flash plug-in with different degrees of attentional demand and proactivity. Which is the best? It depends a lot on the situation: How capable is the user of installing this plug-in? How much control does the user want over disk space or network bandwidth? How concerned is the user about security? Just how funny is the animation Terry sent, anyway? Most plug-ins use a design such as the one in Case 4 because it provides a happy medium.

As this example shows, although we speak of "implicit interactions," it is more accurate to speak of interactions being more and less implicit. Within the course of a task, different aspects of the interaction—the diagnosis, the action, and the feedback—may be more or less implicit. Even though this example reflects a human-computer interaction, the issues that we raised around the implicitness are reflective of the style of the transaction rather than the characteristics of the computer, and thus transcend human-computer interaction to interaction in general.

Now we will examine the two dimensional variables in greater detail:

Attentional Demand

Attentional demand generally is described by the degree of cognitive or perceptual focalization, concentration, and consciousness required of the user. "Foreground interactions" make greater attentional demands on the user, while "background interactions" do not make such demands and, in fact, elude notice.

A more complex definition of attention demand also needs to account for spatiality (as Goffman did in drawing a distinction between "frontstage" and "backstage" interactions), breadth (with many stimuli or just one), or intensity, among other things. This complexity reflects an increasing sophistication in understanding attention itself. Cognitive neuroscientists are starting to believe that attention actually is a catch-all grouping of widely diverse mental functions and phenomena.[6] However, a broad, commonsense understanding of attention allows us to reason sufficiently about our interactions with other humans, and so it is operationally sufficient to design with.

6 Patrick Cavanagh, "Attention Routines and the Architecture of Selection" in *Cognitive Neuroscience of Attention,* Michael I. Posner, ed. (New York: Guilford Press, 2004): 23–24.

Attentional demand can be manipulated by adjusting the perceptual prominence of objects. This may be done through visual organization techniques such as contrast, hierarchy, and weight,[7] as well as more dynamic means such as pointing or placing.[8] Interaction design research on the use of such techniques to present ambient information to users engaged in some other task has been pursued at the MIT Media Lab[9] and Berkeley's Group for User Interface Research,[10] among others.

Another way to change the degree of attention demanded is through "abstraction." By combining elements into a larger whole, the user is presented with less detail. "Chunking" is an example of an abstraction technique through which experts are able to comprehend complex situations (such as the state of a chessboard) with greater ease because they are able to parse the scene into familiar subcomponents.[11] Gestalt psychology has demonstrated that this process of chunking leads an "integrating of awareness," where people are able to identify a whole (say a particular person's face) without being able to identify the details that make up the whole.[12]

This discussion of attentional demand may resonate with those familiar with Bill Buxton's concept of attentional ground:[13] "What we mean by foreground are activities which are in the fore of human consciousness-intentional activities. Speaking on the telephone or typing into a computer are just two examples." Buxton's definition of foreground only overlaps with the left half of the implicit interaction framework, because he only considers the realm of user-initiated interactions—typing on a keyboard or switching on a light—Buxton's definition conflates attention with intention. This definition is inadequate for describing device-initiated interactions—a cell phone ringing or an automatic door opening. These interactions clearly take place in the foreground, but are not at all intentional on the part of the user. Decoupling attention from intention gives us a separate dimension, "initiative."

Initiative

The distinction of who initiates an interaction is critical. If a waiter refills your coffee because you ask him to, that is a *reactive* response to your explicit request. However, if the waiter refills your cup because he sees that it is empty, this interaction becomes implicit. Even if the *proactive* act of pouring the coffee might be in your attentional foreground, the waiter is responding to a projected request for more coffee. (For our purposes, we are only analyzing the interaction on a pragmatic level. Sociologists such as William Foote Whyte[14] have commented on the ways that the server's actual motivations for action are complex and multilayered—the waiter also may be responding to a desire for a tip, for instance, or to make her way around her circuit in an efficient manner. This sophistication of analysis is not needed for the design of implicit interactions.)

7 Luke Wroblewki, "Visible Narratives: Understanding Visual Organization," *Boxes and Arrows* (New York: AIGA, January 20, 2003).

8 Herbert H. Clark, "Pointing and Placing" in *Pointing: Where Language, Culture, and Cognition Meet,* Kita Sotaro, ed. (Mahwah, NJ: Lawrence Erlbaum, 2003), 243–68.

9 Craig Wisneski, Hiroshi Ishii, and Andrew Dahley, "Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information," *International Workshop on Cooperative Buildings* (1998).

10 Tara Matthews et al. "A Toolkit for Managing User Attention in Peripheral Displays," *Proceedings of ACM Symposium on User Interface Software and Technology,* ACM Press 17 (2004): 247–56.

11 William Chase and Herbert Simon, "Perception in Chess," *Cognitive Psychology* 4 (1973): 55–81.

12 Michael Polanyi, *The Tacit Dimension* (London: Cox & Wyman, 1966).

13 William Buxton, "Integrating the Periphery and Context: A New Model of Telematics," *Proceedings of Graphics Interface* (1995): 239–46.

14 William Foote Whyte, *Human Relations in the Restaurant Industry* (New York: McGraw-Hill, 1948).

Initiative is salient in situations in which actors are working together to accomplish a task. From the perspective of those championing direct manipulation or autonomy, joint action is suboptimal because it requires negotiation and coordination. However, it is far easier to think of successful examples of joint actions than terrific tools or perfectly autonomous objects. "Every day, we engage in activities in which we have to coordinate with others to succeed," says Herb Clark. "Face to face, we have systematic, economical, and robust techniques of arranging for joint activities." [15] One can even argue that we can experience readiness-to-hand in interaction with others; certainly we can contrast the ease and transparency with which we can buy a shirt at Macy's with the tortuous process of buying things in a foreign country with a different language and customs. In fact, it is possible to imagine optimal interactions at every point along the initiative continuum. The challenge is in knowing what interaction is appropriate for the situation at hand.

Proactive objects operate in a realm of greater presumption, and so it is common that they need ways of seeing, discerning, and reasoning about the world.[16] This explains why most forays into proactivity, such as the research performed at Microsoft Research,[17] the University of Karlsruhe,[18] and Georgia Tech,[19] have been oriented on the technological issues of sensing, aggregating data, developing user and task models, and performing inference.

And yet the solution for proactive interaction cannot lie in technology alone. People, for all their vaunted intelligence, make inference mistakes all the time, and usually are forgiven. Why is it, then, that interactive products such as the Microsoft Office Helper are so roundly criticized for guessing incorrectly what users are trying to do? It is probably because "Clippy" is untrained in the art of what Goffman calls "facework," sometimes called social graces, *savoir-faire,* diplomacy, or social skills.[20] Since the days of expert dialogue systems, human-computer interaction researchers have considered how "mixed-initiative" interplays between proactive and reactive actions, from both users and computers, can contribute to a project or an understanding." [21] Similar negotiations are necessary on an interaction level to help systems communicate intended actions, and enable user override.

When people go out on a limb, taking initiative in the face of uncertainty, they engage in compensating measures; hedging their actions with techniques such as overt subtlety (where actors make a show of how unobtrusive they are trying to be) or preemptive apology (where actors may bow their head, scrunch up their faces, or raise their shoulders as they execute an action to indicate an apology if their initiative is unwelcome). One could easily imagine, for instance, that recent research on interruptions at Carnegie Mellon[22] and Microsoft Research,[23] which have focused primarily on *when* to

15 Herbert H. Clark, "Arranging To Do Things with Others," *Proceedings of ACM Conference on Human Factors in Computing Systems* (1996): 165–67.

16 David Tennenhouse, "Proactive Computing," *Communications of the ACM* 43:5 (May 2000): 43–50.

17 Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel, "Models of Attention in Computing and Communication: From Principles to Applications," *Communications of the ACM* 46 (2003): 52–59.

18 Albrecht Schmidt, "Implicit Human Computer Interaction through Context," *Personal Technologies* 4:2 and 3 (Springer Verlag, June 2000): 191–99.

19 Daniel Salber, Anind K. Dey, and Gregory D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," *Proceedings of the 1999 Conference on Human Factors in Computing Systems* (1999): 434–41.

20 Erving Goffman, *Interaction Ritual* (New York: Pantheon, 1967), 13.

21 Eric Horvitz, el al., "Models of Attention in Computing and Communication: From Principles to Applications": 52–59.

22 James Fogarty, et al., "Examining Task Engagement in Sensor-Based Statistical Models of Human Interruptibility," *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2005): 331–40.

23 Mary Czerwinski, Edward Cutrell, and Eric Horvitz, "Instant Messaging: Effects of Relevance and Time," *Proceedings of HCI 2000,* XIV Vol. 2 (British Computer Society, 2000), 71–76.

interrupt, could be complemented by research on *how* to interrupt. There are conventional ways to act proactively, even in the face of uncertainty, and these are a matter of sociable design rather than technological intelligence.

Implicit Interaction Design Methodology

Because implicit interactions occur outside of the user's notice or initiative, they can be challenging to design: it is insufficient to project what commands we might issue as users and make them possible. Instead, it is important that the designers of implicit interactions pay greater attention to the interplays between interactants. Our design methodology for implicit interactions uses interaction patterns to help designers model interactive object behaviors of know-how about how to engage in everyday interactions with other people.[24]

Interaction Patterns

The patterns of everyday interactions have been studied by those in other disciplines. Sociologists, for instance, represent what Goffman calls the "strips of activity" as detailed narratives, setting the general context and describing specific behaviors.[25] Artificial intelligence researchers, such as Roger Schank and Robert Abelson, choose to use "scripts"—predetermined, stereotyped sequences of actions that define well-known situations.

Like pattern languages, these interaction patterns provide templates for solutions that designers can share with one another. However, while design patterns suggest high-level approaches to specific classes of design problems, based on previous successful designs, our interaction patterns provide detailed instructions for the oft-implicit communications between actors, and are derived from observations in the world.

Here is an example two interaction sequences, one with a doorman, and another, patterned after the first, with an automatic door that mimics the doorman's implicit behaviors analogously:

SETTING: On a sidewalk at the entrance to a building in the middle of the block.

ROLES: Doorman, Passerby

SEQUENCE:
1 Doorman: [stands in front of the door, wearing a red uniform]
2 Passerby: [walks down street, on a path that will pass the door]
3 Doorman: [spots person walking down street]
4 Passerby: [notices doorman with red finery in front of the door, but keeps on walking]

24 Goffman, *Interaction Ritual,* 13.
25 Roger Schank and Roert Abelson, *Scripts Plans Goals and Understanding* (Hillsdale, NJ: Lawrence Erlbaum, 1977), 41.

5 Doorman: [puts gloved hand on door handle]
6 Passerby: [slows down a little, and looks into the doorway]
7 Doorman: [opens door slightly]
8 Passerby: [keeps walking past door; turns to look down street]
9 Doorman: [lets door shut, and takes hand away from the door handle]

SETTING: On a sidewalk at the entrance to a building in the middle of the block.

ROLES: Door, Passerby

SEQUENCE:
1 Door: [exists, with sign that says “Automatic Door”]
2 Passerby: [walks down street, on a path that will pass the door]
3 Door: [sensors notice motion down the street]
4 Passerby: [notices door frame, and keeps on walking]
5 Door: [makes a soft motor hum noise, as if preparing to open]
6 Passerby: [slows down a little, and looks into the doorway]
7 Door: [opens a little, jiggling its handle]
8 Passerby: [keeps walking past door; turns to look down street]
9 Door: [lets door shut]

In this scripted example, the doorman employs proactive, low-attention techniques to signal his capability for opening doors. He did this through *overt preparation,* when he put his gloved hand on the door handle, and through an *enactment technique,* by pulling the door open a little as a suggestion. An interaction designer designing an automatic door can use the doorman pattern to motivate questions such as how the door draws attention to itself, how it communicates its role as a portal, and how it introduces its affordance. Such steps sometimes can be accomplished implicitly: the door’s mere physical form serves to draw attention and communicate its “door-ness.” The designer also can look for clever ways to achieve the effects of each step: by opening a little when a person walks by, for example, the automatic door can simultaneously draw attention, define its role as a door, and introduce its ability to open automatically by softly humming in overt preparation or jiggling its handle as enactment. The interaction pattern helps designers to determine the roles, setting, and sequence of the interaction to be designed. The interaction analogues allow the designer to imagine functionally equivalent actions, mapping the capabilities of the automatic door against the techniques employed by the doorman, without slavishly and literally replicating his actions.

Issues for Implicit Interactions

Problem Selection vs. Problem Representation

What types of design problems are implicit interaction problems? We introduced implicit interactions by stating that they may be employed when the user is focused on something other than trying to get an interactive device to do what he or she wants; perhaps because the user is physically, socially, or cognitively engaged, or because he or she is not cognizant of what direction that the interaction should take. These are instances where the design requires some degree of agency on the part of the interactive system.

That said, whether a design requires agency is a matter of the designer's point of view. A car, for example, may be said to be driven through the direct manipulation of the steering wheel, gas, brake, and clutch pedals. However, one also can view the interaction between car and driver as a series of sometimes overlapping transactions—that the driver requests greater speed by pressing on the gas pedal, or a change in direction by turning the steering wheel. This second view grows more apt as steer-by-wire technology for automobile operation becomes prevalent. It may be senseless, from a design standpoint, to speak of which view is right or wrong, but it is evident that the adoption of different points of view suggests very different types of solutions.[26]

For this reason, it is useful to view the implicit interaction framework less as a hammer, and more as a lens. From the design research perspective, the implicit interaction framework is a type of problem representation, a means of representing interaction problems so as to make the solution apparent.[27] The central goal of this paper is not to advocate the design of a class of interactive products ("Make implicit interactions!"), but rather to champion a particular approach to designing interactions ("Consider your design as an implicit interaction!"). As Tom Erickson suggests, "There are multiple perspectives from which interaction designers can analyze the sites or situations with which they are confronted, and that designers will fare best when they are able to pick up one lens, then another, and then a third."[28] It is up to the designer to employ the framework and methodology in a mindful manner.

Interdisciplinarity and Appropriation

Part of the challenge of implicit interaction design is making explicit that which is invisible in day-to-day life. One way to do this is for interaction designers to employ sociological methods to understand human-human interactions, and then translate these interactions to novel human-product interactions.

26 Terry Winograd and Fernando F. Flores, *Understanding Computers and Cognition: A New Foundation for Design* (Boston, Addison Wesley Publishing Company) 1986) 77.

27 Herb A. Simon, *The Science of the Artificial* (Cambridge, MA: The MIT Press, 3rd ed., 1996), 132.

28 Tom Erickson, "Five Lenses: Towards a Toolkit for Interaction Design 1," *Foundations of Interaction Design* (New York: Lawrence Erlbaum Associates, 2005).

The application of sociology to human-product interactions is nothing new. Bruno Latour, for instance, enjoyed anthropomorphizing door springs to argue that sociologists need to address the role of nonhumans in their accounts of society:

> For sure, springs do the job of replacing grooms, but they play the role of a very rude, uneducated, and dumb porter who obviously prefers the wall version of the door to its hole version. They simply slam the door shut.[29]

In this paper, we have reversed Latour's approach, objectifying the role of human actors to make products that are less obnoxious, making doors that act not as wall or hole, but as a courteous groom. Designers have broadly employed ethnographically informed practices for decades to inform the user needs or context of the design. This work simply extends the use of ethnography to the generation of positive models for product behavior. We also drew on methods from communications, psychology, and linguistics. For instance, this approach also can be seen as the interactive extension of Reeves and Nass's Media Equation: we expect people to interact most successfully with interactive products in the same manner they interact with other people.[30]

As these techniques are appropriated for design, they are necessarily transformed. The value structures behind the social science methods we use cannot but change when the intended outcome shifts from production of knowledge or performance to production of new interactive systems.[31] We are not claiming that this work is the same as, or a substitute for, the practice of social science by social scientists, or the practice of art by artists within these same domains. At the same time, it is important to recognize that designers *need* to appropriate these techniques and make them their own in order to meet their aims.

In his discussion on studying doormen in New York City, sociologist Peter Bearman notes: "For the founding fathers of sociology,...the city posed special problems for the generation of social order. In contrast to the thick, multivalent, and sustained interactive world of the country, urban interactions were seen as thin, episodic, instrumental, and univalent."[32] (Bearman goes on to argue that urban environments are, in fact, as rich and thick as any other environment.) For designers of interactive systems, however, the desire may very well be to study thin, episodic, instrumental, and univalent interactions, and to ignore layers of motivations and depth of meaning, because the very lack of rich humanity that makes these uninteresting transactions for social scientists makes them the most promising targets for interactive design. Thus, the use of interdisciplinary techniques by designers can offer something original to the world of interaction design.

29 Bruno Latour, "Where Are the Missing Masses? The Sociology of a Few Mundate Artifacts" in *Shaping Technology/Building Society: Studies in Sociotechnical Change*, Wiebe Bijker and John Law, eds. (Cambridge, MA: MIT Press, 1992), 225–58.

30 Byron Reeves and Clifford Nass, *The Media Equation* (New York: Cambridge University Press, 1996).

31 Paul Dourish, "Implications for Design" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York: ACM Press, 2006), 541–50.

32 Peter Bearman, *Doormen* (Chicago: University of Chicago Press, 2005), 17.

Conclusion

As interactive devices continue to permeate our world, it is up to the interaction designers to correct their obnoxious habits in order to make them more usable and useful. Well-designed, implicitly interactive devices can allow us to reap the benefits of computation and communication away from the desktop, assisting us when we are physically, socially, or cognitively engaged, or when we do not know what should happen next. Designed poorly, these same devices can wreck havoc on our productivity and performance, creating irritation and frustration in their wake. By taking stock of what it is we humans do when we work with one another, and using a bit of creativity in applying these lessons to the machine world, we can help make this next generation of interactive devices welcome in our world.

To this end, we have presented a framework for implicit interaction that characterizes interactions based on attentional demand and initiative—factors that are pertinent to any interaction, regardless of domain. We have applied this framework to the use of implicit interaction patterns, which allow designers to apply techniques and solutions from one domain as a template for the analogous solution for another. This framework and methodology can be used by designers as a lens on their interaction design problems, and help them leverage existing linguistic, sociological, or ethnographic techniques to the end of designing better human-computer interactions.

Because implicit interactions have convergent features due to the constraints imposed by the human in the loop, knowledge about the interactions can be generated and generalized—key components in any area of academic research. This transmissibility of solutions from one domain to another also enables design solutions to be passed from one design researcher to another, enabling designers of interactive objects to develop generalized interaction patterns for different classes of interactions.