PRELIMINARY DRAFT
Reality-based Interaction: Understanding the Next Generation of User Interfaces

Robert J.K. Jacob

Department of Computer Science
Tufts University
Medford, Mass. USA

Introduction

Ubiquitous computing, tangible interfaces, and the spread of computers into a wide range of products and objects that surround us are changing interacting with computers from being a specialized activity, segregated from daily life, to becoming increasingly a part of the ‘‘real world.’’ At the same time, as computers are becoming more a part of the real world, user interfaces seem to be evolving to behave more like the real world, as in virtual reality for example. A variety of human-computer interaction styles and developments emerging on several fronts now seem to point in the same direction, toward ‘‘reality-based interaction.’’ We propose using this for leverage to attempt to tie them together and define a new generation of user interfaces.

So far, the next generation of interfaces appears to be more difficult to define than the current GUI generation. Unlike the early days of the GUI or Direct Manipulation interfaces, research in human-computer interaction is currently developing on many fronts, seeming to diverge from the notion of a single new generation of interaction. We find the new generation emerging from a variety of loosely-related research areas being studied today and seek to tie some of them together. These include:

- virtual reality
- augmented reality
- ubiquitous, pervasive, and handheld interaction
- tangible user interfaces
- lightweight, tacit, or passive interaction
- perceptual interfaces
- affective computing
- context-aware interfaces
- speech and multi-modal interfaces.

The leading edge of research in HCI today thus covers a diverse range of new interaction styles without a single unifying thrust. This leading edge is also beginning to generalize HCI into Interaction Design, which includes everyday objects and appliances not previously thought of as computers.

However, there are some threads that could connect many of these developments, and we will try to develop them here and to seek the beginnings of a theory to tie them together. To do this, we will focus on the ways in which interfaces leverage users’ built-in abilities—that is, increasingly natural, realistic, or reality-based interaction. We believe this is both a promising direction in human-computer interaction in general and a notion that can lead to a unifying framework for much current research. We seek here the beginnings of a framework or theory to tie together developments in next generation user interfaces. We propose beginning with our idea of natural or ''reality-based'' interfaces. The key idea is that these interfaces gain their strength because they exploit more knowledge, skills, and abilities that their users already possess than previous interfaces. We will thus formalize this notion of the learned skills vs. the ''reality-based'' skills needed to use a system and attempt to build it into the beginnings of a more precise, formal, and useful theoretical framework.

Can we use the common theme of ''reality-based'' interaction to tie together developments in next generation interaction styles into the beginning of a conceptual framework or a new theory. Recent developments in new interaction styles have tended to proceed independently on many unconnected and unrelated fronts. We attempt to make a connection between them and, ultimately, to provide a more unified framework or basis that can be used to understand, compare, and relate these developments; to find gaps or opportunities for future development; and perhaps to provide some explanatory power for understanding what makes them better or worse.

Reality-based Interaction

The new interfaces increasingly draw their strength from exploiting the user's pre-existing abilities and expectations rather than trained skills. For example, navigating through a conventional computer graphics system requires a set of learned commands, such as keywords to be typed in, or function keys to be pressed. By contrast, navigating through virtual reality exploits the user's existing real-world ''navigational commands'': positioning the head and eyes, turning the body, and walking toward something of interest. Basing the interaction on the real world can reduce the mental effort required to operate the system because the user is already skilled in those aspects of the system. For casual use, this reduction can speed learning; for use in situations involving information overload, time pressure, or stress (e.g., surgery, disaster recovery), this reduction of effort can improve performance. These new ''reality-based'' interaction styles tap into a user's existing knowledge and abilities and leverage them—as direct manipulation or graphical interfaces have done— but to a greater degree. Direct manipulation moved user interfaces toward more realistic interaction with the computer; next generation, reality-based interfaces simply push further in this direction, increasing the realism of the interface objects and allowing the user to interact even more directly with them.

We begin with the notion of ''reality-based'' interaction, which ties together a wide range of new interaction styles. As we move from command line interfaces to direct manipulation, virtual reality, and embedded, ubiquitous, and tangible interfaces, we see interfaces becoming more like the ''real world.'' Such interfaces seek to make the user's input actions as close as possible to the thoughts that motivated those actions, that is, to reduce the ''Gulf of Execution'' described by Hutchins, Hollan, and Norman[17], the gap between the user's intentions and the actions necessary to input them into the computer. The motivation for doing this is that such interfaces build on the equipment and skills humans have acquired through evolution and experience and exploit them for communicating with the computer. We believe that a key benefit of direct manipulation came from its use of metaphors based on the real world for aspects of its interaction. It uses operations that the user already knows—pointing, grabbing, and moving objects from one place to another[33]. The interface syntax leverages facts that the user already knows; it does not require a special language. As noted, virtual reality interfaces, too, gain their strength by exploiting the user's pre-existing perceptual and navigational abilities. Tangible user interfaces also leverage real-world manipulation of physical objects to provide a more realistic interface[18]. The result is to increase the user-to-computer bandwidth of the interface and to make it more realistic, because interacting with it is more like interacting with the rest of the world. The unifying characteristic for much of the next generation of human computer interaction thus appears to be that it increasingly taps into the users' abilities and pre-existing knowledge—providing more natural, reality-based, or realistic user interfaces.

We note a few caveats before proceeding: First, we are trying to characterize and evaluate each interaction style or genre as a whole, rather than each specific user interface. Models such as GOMS[22], the Keystroke-Level Model, EPIC, or Soar can address the latter, but we are seeking broader generalizations about whole interaction styles here. Second, we are characterizing only what the user needs to know to operate the *interface*, that is, syntactic knowledge. We explicitly omit any application or task domain knowledge from analysis, because it would be constant across different interaction styles for the same application. And, third, we have yet to address speech interfaces fully, precisely because they require so little interaction-style knowledge and so much higher-level domain knowledge.

Reality-Plus

The other side of this issue is that reality-based is typically not sufficient. As noted, a useful interface will rarely entirely mimic the real world, but will necessarily include some ‘‘unrealistic’’ or artificial features and commands. In fact much of the power of using computers comes from this ‘‘multiplier’’ effect, the ability to abstract from or go beyond a precise imitation of the real world. For example, in a GUI, one might want to go beyond realistically pointing to and dragging individual files to more abstract commands like *Archive all files older than 180 days* or *Delete all files that contain the text string ‘‘reality-based.’’*[31, 36] We therefore propose a view that identifies some fraction of a user interface as based on realistic knowledge or abilities plus some other part that provides computer-only functionality that is not realistic. As a design approach or metric, the goal might be to make the first category as large as possible and use the second only as necessary[41].

For example, consider the character Superman. He walks around and behaves in many ways like a real man. He has some additional functions for which there is no analogy in real humans, such flying and X-ray vision. For doing realistic things, he uses his real-world commands, walking, moving his head, looking around. But he still needs some additional non-real-world commands for flying and X-ray vision. We can go a step further and ask that these non-real-world commands be analogous to some realistic counterpart—perhaps that they use a similar modality, but somehow augmented, and that the response to the input be similar to the real world (and in a similar modality), but also augmented in some way.

For example one might fly in virtual reality by leaning in the direction you want to go (rather than, say, using an arbitrary hand gesture). Or we might track the user's eye and notice when it is staring intently at an object and use that as the command for X-ray vision—as shown in[40]. We can thus divide the non-realistic part of the interface into degrees of realism (fly by leaning vs. by menu pick). The goal would be to perform realistic tasks realistically, to provide additional non-real-world functionality, and to use analogies for its commands where possible. The designer should seek the right balance of realism and artificiality and assign tasks to each appropriately. We discuss this tradeoff between reality-based interaction and expressive power further below.

Toward a Theory

Next we need to take the vague notion of ‘‘realistic’’ or ‘‘natural’’ and make it more precise. The idea that a reality-based or natural interaction style is somehow better is promising, but vague. We want to see what leverage we can get by taking it further, building upon the notion, making it more precise, and providing operational definitions of some of our concepts.

How can we formalize or even quantify the simple notion that ‘‘realistic is better?’’

We will do so by focusing on the pieces of knowledge or skill that a system requires its user to know. In this view, reality-based interfaces are better because they use things that the user already knows from his or her interaction with the real world—preferably knows so well that they are second-nature. This leads to the idea of a checklist of the knowledge the user needs. It may only be a notional checklist, the complete list may be far too complex to enumerate, but the idea of the list can drive the theory and, especially, will help illuminate relative comparisons between interaction styles, rather than making absolute measurements about them.

Returning to the example of VR navigation, let us compare two interfaces for the same task: viewing and evaluating a 3-D computer model of a building interior. One interface uses a 2-D screen and a mouse and keyboard to input commands that change the point of view. The other uses a virtual reality head-mounted and head-tracked display. What does a user need to know for navigation? To move the point of view in the first system, the user needs to know a set of navigation commands—perhaps mouse or arrow key commands for panning, shift keys or other special commands for zooming, yawing, pitching, or rolling the point of view. In the virtual reality system, the user commands for changing the point of view consist of moving one's head or body in just the same way one would move to change point of view in the real world. We claim, therefore, that the user already knows those commands: they were learned as an infant. To see the view to your right, you turn your head to your right; there is no need to learn a mouse or keyboard command to do this.

If we compare the ‘‘checklist’’ for the two interfaces, the mouse and keyboard interface requires knowing these navigation commands, which must be learned from the instruction manual. The virtual reality interface requires knowing the ‘‘navigation commands’’ that every infant already knows. Therefore, the second interface requires the user to learn one less new piece of knowledge to operate it, because it leverages knowledge the user already has. And by this metric it is therefore ‘‘better.’’ Both systems also require a large amount of task-related knowledge about buildings, architecture, and the architectural problem at hand, but these are the same for both interfaces for the building interior task, so they shed no light on differences between the interaction styles. We propose that this notional checklist provides a way to quantify the idea of a ‘‘reality-based’’ user interface: a more reality-based interface is one in which the user already knows more of the items on the ‘‘checklist’’ from real-world experience.

Gradations of Realism

However, there are many kinds of things that the user already knows. Moving the head to change point of view is one. The user may already know more arcane facts such as pressing the Alt-F4 keys will close a window. It seems intuitively better to exploit the more ‘‘basic,’’ more built-in knowledge that the user learned in infancy (or perhaps was born with) than to exploit more recently learned, less innate knowledge, like the Alt-F4 keys. Our next step, therefore, is to take our checklist and give each piece of knowledge a rating rather than simply a Boolean check mark for reality-based or non-reality-based. We now thus view reality-based as a continuum rather than a dichotomy. We can use our imaginary checklist by summing these ratings to describe an interface by the amount and degree of non-reality-based knowledge required to operate it.

To do this we need a way to rate the degree of reality or innate-ness for a piece of knowledge. One way is by when the user learned it; and we conjecture that younger is better. Information that is deeply ingrained in the user seems somehow more robust, perhaps more highly practiced, and should take less effort to use than information learned recently. In particular, if the user is distracted or highly engaged in the domain aspects of the task rather than the interface aspects (as should be the case with a good interface), it should be easier to call up basic knowledge from childhood than information learned from studying the user manual more recently. We may be able to define gradations like the following (and conjecture that the earlier ones are better):

1. Fixed at birth: e.g., size, shape, scale of body, hard-wired mental and motor processes (excluding skills present at birth but later lost, such as some reflexes in newborn babies)
2. Learned early by nearly all humans, regardless of culture: e.g., locomotion, visual perception, grabbing, dragging, dropping, manipulating objects
3. Learned without schooling, but possibly culture-specific: e.g., doorknob opens a door
4. Known by non-literate adults, possibly culture-specific: e.g., how to use shelves for piling things
5. Known by ‘‘gadget-literate’’ adults: e.g., operation of switches, knobs, and dials
6. Known by most computer-literate adults: e.g., use of mouse, icons, windows, pull-down menus
7. More specific computer conventions: e.g., File menu is on left, Exit is last item on menu.

In general, in each case we are interested in the age of when most humans typically learn an item, rather than when a particular individual may have learned it.

Another, more objective approach would be to test a representative population of users on each skill or piece of knowledge. We can certainly imagine administering tests for skills such as grabbing and moving real objects or finding the *Exit* command on a pull-down menu. Some of these tasks are so easy to do in isolation that actually testing them will require adding a distractor task, time pressure, or other ways of raising the workload in the test in order to match a situation in which a user is trying to accomplish a real task and choosing the menu item is a needed step but not the only concern. We could in this way obtain a reasonably objective measure of skill level for each of our pieces of knowledge. Actually performing all these tests is cumbersome, but realizing that it is possible, being able to formulate them in principle, and conducting some representative ones could give confidence that this approach can be grounded in objective measurement.

Formalizing the Checklist

Now, we take our notional checklist and make it formal or operational.

For a given interaction style, enumerate the interface-related (not application domain) knowledge and skills needed to use it. Defining the granularity for these is a key issue. A possible starting point is to use whole interaction techniques. An interaction technique is a way of using a physical input device to perform a generic task in a human-computer dialogue[13]. For example, a pop-up menu is a way of using a screen and mouse to make a choice among a small set of discrete items. Interaction techniques occupy the intermediate area between the device technology and a user interface for a specific application, and so provide a good starting point. An interaction technique like a pop-up menu also implies a combination of inputs and outputs, so the knowledge the user needs includes both motor tasks to operate it and perceptual tasks to interpret the feedback. Our evaluation would thus reward both realistically-based actions and feedback and so include the value of familiar representations and idioms and of stimulus-response compatibility.

Perhaps the set of basic interaction tasks identified by Foley, Wallace, and Chan[12] will serve as the atomic units for this analysis. These were originally conceived for 2-D interactive graphics systems and provide a useful breakdown of the atomic ways a user can interact with such systems. We may be able to divide more complex tasks into sequences of these basic interaction tasks, and then we need only to determine the rating of different ways to perform each of these (relatively few) basic tasks for each interaction style. Or perhaps a new analogue or extension of the basic tasks of Foley et al. beyond the realm of desktop 2-D and 3-D graphics interfaces will be needed, or perhaps an entirely different set of basic units.

Cognitive Task Analysis may help here in identifying basic units of knowledge or skills for a task[16].

Next, for each item, we rate it as ‘‘realistic’’ or not, or, more precisely, as ‘‘known’’ vs. ‘‘needs to be learned.’’ Within the ‘‘known’’ category, we further rate its gradation. We could do this by simply postulating an age at which people learn the skill, or by using published data about learning and development or published experimental results, or by running small experiments of our own on a target population, as appropriate in each case. We could also approach this by directly measuring or estimating users’ current knowledge or skill for each component task as discussed above. That is, simply rate or test potential users on each skill, regardless of how or when they learned or practiced it. In evaluating items the user does not know, we will also discuss below considering the cost of learning the item.

We can then ‘‘score’’ our interaction style by summing these values, as we have shown in[7]. More precisely, we may prefer to sum the ‘‘gap’’ for each knowledge item, that is $(1 - k)$ which gives the difference from a perfect score, if k is the knowledge rated on a scale of 0 to 1, where 0 is unknown and 1 is known innately or instinctively. The result is an overall rating for each interaction style. It is not an absolute performance prediction as GOMS or other models would generate, but a relative rating useful for comparing interaction styles. It also provides a breakdown of each rating into its components, so strengths and weaknesses of proposed modifications can be discussed more objectively.

Other Axes (TBD)

Tradeoff Between Reality-based Interaction and Power

(TBD)

Examining New Interaction Styles

Virtual Reality

Virtual reality provides realistic-looking views of the virtual world (although they generally can only be seen, not touched nor smelled). As discussed, the realistic interactions for moving the head, walking short distances, and reaching for objects would be checked off our checklist because the user already knows them. These are often augmented with non-realistic navigation commands for traversing larger distances and additional actions for executing special commands: these would not be checked off.

Tangible User Interfaces

Tangible user interfaces use real physical objects and actions. A user might grab and move a physical model of a building to reposition it, seeing, touching, and feeling it exactly as one would in the real world. The skills required for these tasks would, again, be checked off our checklist. This physical representation may be augmented by a ‘‘digital shadow’’[18] in which the computer presents additional non-realistic information and feedback (such as computing the effect of the proposed building on traffic patterns).

Augmented Reality

Augmented reality allows a user to navigate through the real world and manipulate objects in it directly, using his or her natural skills. These are again checked off our checklist. It then provides additional visual feedback from the computer, not checked off.

Lightweight Interaction

We also consider an emerging genre of ‘‘lightweight’’ interfaces, which have also been variously called passive, tacit, non-command, low-intentional, context-aware, perceptual, or sensing interfaces. These try to obtain useful information from a user, without much effort from him or her. They take a step away from the traditional human-computer dialogue toward a more subtle, implicit interchange based on passive monitoring of the user’s actions rather than explicit commands. The user thus need not always give explicit commands, but the system might observe, guess, infer, and take hints from him or her[27]. Input for this purpose can include eavesdropping on user commands intended for other purposes, physiological sensors (skin resistance, eye movements), or behavioral measures (typing or mousing speed). This represents a change in input from objects for the user to *actuate* by specific commands to passive equipment that simply *senses* parameters of the user’s body. In contrast, previous interaction styles—batch, command line, menu, natural language, and GUI styles—all await, receive, and respond to explicit commands from the user to the computer.

Lightweight or context-aware interfaces can, similarly, be viewed by checking items off our checklist. To the extent that they glean inputs from users without explicit action, users need not learn how to provide those inputs. Or, if they glean inputs from other actions, the user already knows how to perform these. In our framework, the knowledge needed for these would thus be checked off, reducing the knowledge needed to operate the system.

Ubiquitous Computing

(TBD)

Adaptive Interfaces and Affective Computing

Adaptive or affective interfaces appear to fit our approach in a similar way. As with lightweight interfaces, the user need not know any explicit commands for adapting or changing modes because the system takes inputs from its passive sensors and makes inferences from them automatically. To the extent that the system makes good inferences without the user making special actions, this allows a set of user interface knowledge to be checked off our checklist of what the user must know to operate the system. Note that a system that does not adapt at all would also have exactly the same checklist status—it requires the same amount of knowledge—but would presumably be less powerful or useful.

As an analogy, neither a fixed focus camera nor an auto-focus camera requires the user to know about focusing. But the auto-focus camera accomplishes more with the same quantity of user knowledge. And, of course, a poorly designed auto-focus camera, like a poorly designed adaptive system, is the worst of all, because it requires the user to know even more in order to outwit the automatic system.

GUI

(TBD)

Issues and Extensions

All three of these fit our view of ‘‘reality-plus’’ interfaces, with a reality-based part and an augmented or artificial part. They allow the reality-based part to be checked off the checklist and include an artificial part that must be learned.

Reality-based means that the basic interaction style or substrate is *based in* reality. Other features that are not necessarily realistic may be added on top of this base. We do not seek an entire user interface just like the real world, but rather that it be based on reality. In each of the above examples, the point is not that the interface is entirely realistic—but rather that it is more realistic than previous interaction styles. It uses more reality-based knowledge and skills than, for example, current graphical or command-line interfaces and thereby affords better transfer of skills from real world to computer. Our framework attempts to make that transfer more explicit and measurable.

Further, the notion of reality-based should be interpreted as preferring an interaction or display that you already know—where possible. This specifically excludes commands for which there is no ‘‘realistic’’ equivalent. It thus applies to evaluating alternative commands to pick up and move individual objects (for example, in virtual reality), because there is an equivalent physical operation in the real world. But it cannot apply to a command like *Pick up only the red objects*, because this always requires an artificial extension (or perhaps introducing an agent into the interface).

Expert User Interfaces

Our initial approach must be expanded to address user interfaces for experts. Everyone learns to move their head to change their point of view; many learn to operate a mouse and pull-down menu; but few learn to play a violin or operate the cockpit of a commercial airliner. These expert interfaces may be less ‘‘reality-based’’ but we believe they can be placed into our framework straightforwardly. First, we claim that, all other things being equal, an interface that uses what you already know is better. Introducing new, expert-user commands thus has a cost, and it must repay this through better performance. For example, an airplane cockpit is far from a ‘‘realistic’’ interface for human flight (if there is one). But, once learned, it is very effective. The cost of learning this new ‘‘non-realistic’’ interface is repaid by improved performance. Second, the cockpit is indeed based on knowledge of the physical world—reaching for levers, turning knobs, idioms for dial or digital displays as well as, increasingly, the familiar idioms and metaphors of existing computer displays and controls.

We can thus extend our approach to cover the cost and benefit of knowledge that must be acquired to use an interface. Beyond simply checking items as already known or not known, we would address the value and cost of learning a new item. We would model user state changes (user learns *X*) and then evaluate the benefits of the new state (in which knowledge *X* is now ‘‘checked off’’). One consequence is that this should account for the benefit of consistency in user interfaces as a straightforward consequence.

Finally, the notions of ‘‘floor’’ and ‘‘ceiling’’ introduced by Myers, Hudson, and Pausch[24] may be of value in this framework. ‘‘Floor’’ describes the minimum knowledge needed for a novice user to use a system. For the tools they surveyed, the floor might be what is required to build a ‘‘Hello world’’ program. ‘‘Ceiling’’ describes the maximum power of a tool, the most complex artifact that it can produce. Our initial approach seems to apply to the floor, but it could also apply to the ceiling for expert users. For example, devoting less mental effort to syntax could leave more for virtuosity within the application domain.

Embedded Knowledge

We must also consider the issue of knowledge that is ‘‘embedded’’ in the world or the interface device. Affordances and constraints can directly embody pieces of knowledge about how to operate the device. Then the user need not know such pieces of knowledge nor learn them to use the system. In terms of our notional checklist, such pieces of knowledge could thus be checked off the list. For example, a physical slider embodies the constraint that it can be moved in only one direction; the user does not need to be taught not to slide it the wrong way. Ullmer extends this notion to a tangible interface in which considerably more complex query syntax rules are directly embodied in the physical constraints of the interface[44]. Our framework would credit this knowledge appropriately.

A corresponding notion for output is a self-disclosing interface, a system that informs its user about its commands and capabilities. For example, a physical (or graphical) control panel suggests the available parameters and the kinds of adjustments that can be made to them. A basic command line interface, such as MS-DOS with its blank screen and terse ‘‘C:>’’ prompt, or a speech interface with only a microphone pointing at the user, discloses little about the commands available to the user and thus requires more user interface knowledge.

Other Issues

While the approach we have described here covers a wide swath of current interaction technique research, including virtual reality, augmented reality, tangible, eye movement, light-weight, context-aware, and affective interfaces, some areas will require further consideration.

As noted, we have not yet addressed speech interfaces, because they are difficult to pigeonhole into an interaction style. Using speech defines and constrains an interface so little that it is almost like saying an interface will use the hands (or a keyboard) for input or the eyes for output. The *interaction style* knowledge required is minimal—the ability to utter words, or press keys, or view images. A host of application-specific knowledge about which keys to press or which words to speak may also be required, but these do not characterize the interaction style itself. The interaction style is protean, and thus there seems to be less interaction style-specific knowledge to account for than in most of the styles we have addressed.

We suggest not treating hand-held interaction as a separate category for the present, because most current examples use existing interaction styles, typically miniature versions of conventional graphical user interfaces. They do not yet comprise a distinct interaction style.

A further conjecture is that an interface that uses extra real-world information is better that one that does not. That is, assuming they both require the same amount of non-real-world knowledge, the two would otherwise score identically in terms of new knowledge to be learned. But we conjecture that the former might engage the user more strongly. Our framework allows this conjecture to be tested explicitly, by modifying the way we sum the knowledge components to reach a final score.

Finally, there are other dimensions that may also be useful for rating gradations of user interface knowledge and skills, such as: declarative (words, proposition-like) vs. procedural (catch a fly ball) knowledge; arbitrary vs. non-arbitrary mappings (this includes stimulus-response compatibility); knowledge that is more or less memorable or distinctive; and virtuosity, acquired by extensive practice.

Creativity and Satisfaction

Our framework thus far principally addresses the ability or inability to perform straightforward tasks and measures how this ability comes from exploiting existing skills, patterns, stereotypes, and knowledge. But it measures only whether or not the user can execute a given fixed task; we would also like to address the quality or expressiveness or virtuosity of the task and the user's satisfaction or pleasure. We see an emerging trend in HCI beyond traditional task performance, speed, and error rate to issues of enjoyment, satisfaction, and creativity[4, 35]. This parallels the extension of computer use beyond the workplace into broader aspects of life.

Our framework indirectly addresses the amount of mental effort the user must devote to operating a system. We assume that an interface based more on reality requires the user to concentrate less on the mechanics of the interface. This would allow him or her to concentrate more on the task at hand, and that ought to improve the quality or expressiveness of the final product. Shneiderman's theory of syntactic vs. semantic knowledge argues that the user ought to concentrate his or her mental effort on the *semantics* of the task, and that reducing the amount of *syntactic* knowledge required serves this goal[34]. Hutchins, Hollan, and Norman argue similarly for direct manipulation interfaces. They claim that reducing the ''Gulf of Execution'' and the ''Gulf of Articulation'' allows the user to concentrate more directly on the task rather than on the interface[17]. This may help us include expressiveness or virtuosity in our framework.

Other new measures might also predict user satisfaction within our framework. For example, we could evaluate the degree of control and predictability that a user interface provides, perhaps the degree of directness of its direct manipulation, or a rating for a transparent vs. black-box design. We conjecture that an interface that gives powerful and predictable control would lead to greater user satisfaction. However, this fails to account for designs intended to surprise or delight their users. For example Sengers' Influencing Machine (intended as an artwork rather than an interface for efficient task performance) interests its audience precisely because the user's control over the machine is indirect, subtle, and difficult to predict[32]. Similarly, with Wensveen and Overbeeke's alarm clock design[46], the relationship between user input and system output is complex and indirect for what would otherwise be a straightforward task. A possible conjecture here is that this sort of pleasure or surprise or delight works best when it is partially grounded in reality. Just as a joke or magic trick must begin with a plausible situation and then add an unexpected twist on top of it, perhaps these interfaces succeed by first being based on real world situations and then adding something that violates the user's mindset or assumptions.

Related Work

Related work in developing reality-based interaction techniques includes much research in virtual reality[1], tangible interfaces and related approaches[10, 18, 26, 28], embodied interfaces[8], reality-based eye movement-based interaction[38], and lightweight interfaces, such as tacit interaction[25], affective interfaces[29], sensing interfaces[3], and non-command interfaces[27]. However, to date, work that attempts to explain or organize these new styles of user interfaces has tended to concentrate more on individual classes or groups of new interfaces than on ideas that unify several classes. For example, Ullmer and Ishii provide a framework for tangible interfaces[43]; Nielsen presents the idea of non-command interfaces[27]; Streitz addresses architectural spaces with roomware or cooperative buildings[39]; and Fishkin, Moran, and Harrison propose the concept of embodied interfaces[8].

Other work connects with some of the issues we raise in a variety of ways. Bellotti et al. investigate sensing interfaces, and raise a set of key problems[3]. These were implicitly solved by GUIs but require more careful thought for these next generation interfaces. These questions provide a useful framework for understanding new interfaces.

Beaudouin-Lafon's Instrumental Interaction model is related in the way it breaks an interface into domain objects, which are of interest to the user, and interaction instruments, which are the mediators through which the user interacts with the domain objects[2]. He discusses how the interaction instruments can be based on analogies from physical reality and can exploit the user's existing knowledge. In a similar vein, Rohrer addresses the notion of the reality-based aspects of an interface vs. the artificial or ''magical'' ones[31].

Fishkin presents a new taxonomy for understanding and analyzing tangible user interfaces[9].

Winograd presents a useful general software architecture and design for a broad class of next-generation interfaces[47].

Shneiderman[36] argues for ''enhanced 3-D'' interfaces that extend reality. We further claim that such interfaces ought to be based on 3-D reality rather than on arbitrary artifacts. Gray and Boehm-Davis provide some experimental evidence of a direct impact of the details of low-level, lightweight interaction on user behavior at a much higher level[14].

Finally, the work that helped define the current (GUI) generation may be the most closely related model for this research. Shneiderman took a variety of what at the time seemed disparate new user interface inventions and brought them together by noting their common characteristics and defining a new generation of user interfaces (''Direct Manipulation''), rather than a diverse collection of interesting, powerful, but otherwise unconnected new designs[33]. Hutchins, Hollan, and Norman then explained the power and success of these interfaces with a theoretical framework and provided a basic understanding of that new generation of user interfaces in human terms[17]. Our hope is to take a first step in that direction for the emerging new generation of interaction.

Conclusions

Recent developments in new interaction styles have proceeded independently on several unconnected and unrelated fronts. We seek to advance and unify the area of next generation interaction techniques. We attack the problem by using the notion of reality-based interfaces to provide a unified conceptual framework for understanding, comparing, relating, and evaluating these developments. We provide a basis for understanding and comparison across otherwise disparate subdisciplines of HCI—such as ubiquitous computing, virtual reality, and tangible interfaces—and to provide some explanatory power for understanding what makes each better or worse. We hope to provoke further research on theoretical bases for next generation interfaces and lead to better understanding of these interfaces. And we hope our framework can inform and help guide future research and development agendas by identifying opportunities or ''holes'' in the taxonomy or ''sweet spots'' for new interaction technique designs and future research.

References

1. W. Barfield and T.A. Furness, *Virtual Environments and Advanced Interface Design*, Oxford University Press, New York (1995).

2. M. Beaudouin-Lafon, ''Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces,'' *Proc. ACM CHI 2000 Human Factors in Computing Systems Conference*, pp. 446-453, Addison-Wesley/ACM Press (2000).

3. V. Bellotti, M. Back, W.K. Edwards, R.E. Grinter, A. Henderson, and C. Lopes, ''Making Sense of Sensing Systems: Five Questions for Designers and Researchers,'' *Proc. ACM CHI 2002 Human Factors in Computing Systems Conference*, pp. 415-422, ACM Press (2002).

4. M.A. Blythe, A.F. Monk, K. Overbeeke, and P.C. Wright, *Funology: From Usability to Enjoyment*, Kluwer Academic Publishers, Dordrecht (2003).

5. S. Bodker, *Through the Interface. A Human Activity Approach to User Interface Design*, Lawrence Erlbaum, Hillsdale, N.J. (1991).

6. A. Chang, S. O'Modhrain, R. Jacob, E. Gunther, and H. Ishii, ''ComTouch: Design of a Vibrotactile Communication Device,'' *Proc. ACM DIS 2002 Designing Interactive Systems Conference*, ACM Press (2002). [http://www.cs.tufts.edu/~jacob/papers/dis02.pdf]

7. G. Christou and R.J.K. Jacob, ''Evaluating and Comparing Interaction Styles,'' *DSV-IS'03 Eurographics Workshop on Design, Specification, Verification of Interactive Systems*, Short paper (2003). [http://www.cs.tufts.edu/~jacob/papers/dsvis.03.pdf]

8. K.P. Fishkin, T.P. Moran, and B.L. Harrison, ''Embodied User Interfaces: Toward Invisible User Interfaces,'' *Proc. of EHCI'98 European Human Computer Interaction Conference*, Heraklion, Crete (1998).

9. K.P. Fishkin, ''A Taxonomy for and Analysis of Tangible Interfaces,'' Manuscript, Intel Research Seattle, Seattle, Wash. (2003).

10. G.W. Fitzmaurice, H. Ishii, and W. Buxton, ''Bricks: Laying the Foundations for Graspable User Interfaces,'' *Proc. ACM CHI'95 Human Factors in Computing Systems Conference*, pp. 442-449, Addison-Wesley/ACM Press (1995).

11. G.W. Fitzmaurice and W. Buxton, ''An Empirical Evaluation of Graspable User Interfaces: Towards Specialized, Space-multiplexed Input,'' *Proc. ACM CHI'97 Human Factors in Computing Systems Conference*, pp. 43-50, Addison-Wesley/ACM Press (1997).

12. J.D. Foley, V.L. Wallace, and P. Chan, ''The Human Factors of Computer Graphics Interaction Techniques,'' *IEEE Computer Graphics and Applications*, vol. 4, no. 11, pp. 13-48 (1984).

13. J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Mass. (1990).

14. W.D. Gray and D.A. Boehm-Davis, ''Milliseconds Matter: An Introduction to Microstrategies and to Their Use in Describing and Predicting Interactive Behavior,'' *Journal of Experimental Psychology: Applied*, vol. 6, no. 4, pp. 322-335 (2000).

15. D. Hix, J.N. Templeman, and R.J.K. Jacob, ''Pre-Screen Projection: From Concept to Testing of a New Interaction Technique,'' *Proc. ACM CHI'95 Human Factors in Computing Systems Conference*, pp. 226-233, Addison-Wesley/ACM Press (1995). [http://www.cs.tufts.edu/~jacob/papers/chi95.pdf]

16. R.R. Hoffman, B. Crandall, and N.R. Shadbolt, ''Use of the Critical Decision Method to Elicit Expert Knowledge: A Case Study in the Methodology of Cognitive Task Analysis,'' *Human Factors*, vol. 40, no. 2, pp. 254-276 (1998).

17. E.L. Hutchins, J.D. Hollan, and D.A. Norman, ''Direct Manipulation Interfaces,'' in *User Centered System Design: New Perspectives on Human-computer Interaction*, ed. by D.A. Norman and S.W. Draper, pp. 87-124, Lawrence Erlbaum, Hillsdale, N.J. (1986).

18. H. Ishii and B. Ullmer, ''Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms,'' *Proc. ACM CHI'97 Human Factors in Computing Systems Conference*, pp. 234-241, Addison-Wesley/ACM Press (1997).

19. R.J.K. Jacob and L.E. Sibert, ''The Perceptual Structure of Multidimensional Input Device Selection,'' *Proc. ACM CHI'92 Human Factors in Computing Systems Conference*, pp. 211-218, Addison-Wesley/ACM Press (1992).

20. R.J.K. Jacob, L.E. Sibert, D.C. McFarlane, and M.P. Mullen, Jr., ''Integrality and Separability of Input Devices,'' *ACM Transactions on Computer-Human Interaction*, vol. 1, no. 1, pp. 3-26 (March 1994). [http://www.cs.tufts.edu/~jacob/papers/tochi.pdf]

21. R.J.K. Jacob, H. Ishii, G. Pangaro, and J. Patten, ''A Tangible Interface for Organizing Information Using a Grid,'' *Proc. ACM CHI 2002 Human Factors in Computing Systems Conference*, pp. 339-346, ACM Press (2002). [http://www.cs.tufts.edu/˜jacob/papers/chi02.pdf]

22. B.E. John and D.E. Kieras, ''The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast,'' *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 4, pp. 320-351 (December 1996).

23. T.P. Moran, E. Saund, W. Van Melle, A.U. Gujar, K.P. Fishkin, and B.L. Harrison, ''Design and Technology for Collaborage: Collaborative Collages of Information on Physical Walls,'' *Proc. ACM UIST'99 Symposium on User Interface Software and Technology*, pp. 197-206, Addison-Wesley/ACM Press (1999).

24. B. Myers, S.E. Hudson, and R. Pausch, ''Past, Present, and Future of User Interface Software Tools,'' *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 1, pp. 3-28 (March 2000) Also reprinted in Human-Computer Interaction in the New Millenium, ed. J.M. Carroll, Addison-Wesley/ACM Press, Reading, Mass., 2001, pp. 213-233.

25. L. Nelson, S. Ichimura, E.R. Pedersen, and L. Adams, ''Palette: A Paper Interface for Giving Presentations,'' *Proc. ACM CHI'99 Human Factors in Computing Systems Conference*, pp. 354-361, Addison-Wesley/ACM Press (1999).

26. W. Newman and P. Wellner, ''A Desk Supporting Computer-based Interaction with Paper Documents,'' *Proc. ACM CHI'92 Human Factors in Computing Systems Conference*, pp. 587-592, Addison-Wesley/ACM Press (1992).

27. J. Nielsen, ''Noncommand User Interfaces,'' *Comm. ACM*, vol. 36, no. 4, pp. 83-99 (April 1993).

28. J. Patten, H. Ishii, J. Hines, and G. Pangaro, ''Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces,'' *Proc. ACM CHI 2001 Human Factors in Computing Systems Conference*, pp. 253-260, ACM Press (2001).

29. R.W. Picard, *Affective Computing*, MIT Press, Cambridge, Mass. (1997).

30. J. Rekimoto, B. Ullmer, and H. Oba, ''DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions,'' *Proc. ACM CHI 2001 Human Factors in Computing Systems Conference*, pp. 269-276, Addison-Wesley/ACM Press (2001).

31. T. Rohrer, ''Metaphors We Compute By: Bringing Magic into Interface Design,'' Technical Report, Center for the Cognitive Science of Metaphor, Philosophy Department, University of Oregon (1995). [http://philosophy.uoregon.edu/metaphor/gui4web.htm]

32. P. Sengers, R. Liesendahi, W. Magar, C. Seibert, B. Muller, T. Joachims, W. Geng, P. Martensson, and K. Hook, ''The Enigmatics of Affect,'' *Proc. ACM DIS 2002 Designing Interactive Systems Conference*, pp. 87-98, ACM Press (2002).

33. B. Shneiderman, ''Direct Manipulation: A Step Beyond Programming Languages,'' *IEEE Computer*, vol. 16, no. 8, pp. 57-69 (1983).

34. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading, Mass. (1987).

35. B. Shneiderman, *Leonardo's Laptop: Human Needs and the New Computing Technologies*, MIT Press, Cambridge, Mass. (2002).

36. B. Shneiderman, ''Why Not Make Interfaces Better Than 3D Reality?,'' *IEEE Computer Graphics and Applications*, pp. 12-15 (November/December 2003).

37. L.E. Sibert and R.J.K. Jacob, ''Evaluation of Eye Gaze Interaction,'' *Proc. ACM CHI 2000 Human Factors in Computing Systems Conference*, pp. 281-288, Addison-

Wesley/ACM Press (2000). [http://www.cs.tufts.edu/~jacob/papers/chi00.sibert.pdf]

38. I. Starker and R.A. Bolt, ‘‘A Gaze-Responsive Self-Disclosing Display,’’ *Proc. ACM CHI’90 Human Factors in Computing Systems Conference*, pp. 3-9, Addison-Wesley/ACM Press (1990).

39. N.A. Streitz, P. Tandler, C. Muller-Tomfelde, and S. Konomi, ‘‘Roomware: Toward the Next Generation of Human-Computer Interaction Based on an Integrated Design of Real and Virtual Worlds,’’ in *Human-Computer Interaction in the New Millenium*, ed. by J.M. Carroll, Addison-Wesley/ACM Press, Reading, Mass. (2001).

40. V. Tanriverdi and R.J.K. Jacob, ‘‘Interacting with Eye Movements in Virtual Environments,’’ *Proc. ACM CHI 2000 Human Factors in Computing Systems Conference*, pp. 265-272, Addison-Wesley/ACM Press (2000). [http://www.cs.tufts.edu/~jacob/papers/chi00.tanriverdi.pdf]

41. V. Tanriverdi, ‘‘A Virtual Reality Interface Design (VRID) Model and Methodology,’’ Doctoral dissertation, Tufts University (2001) ISBN 0-493-14421-8, University Microfilms Inc. Order No. 3004806.

42. B. Ullmer and H. Ishii, ‘‘The metaDESK: Models and Prototypes for Tangible User Interfaces,’’ *Proc. ACM UIST’97 Symposium on User Interface Software and Technology*, pp. 223-232, Addison-Wesley/ACM Press, Banff, Canada (1997).

43. B. Ullmer and H. Ishii, ‘‘Emerging Frameworks for Tangible User Interfaces,’’ in *Human-Computer Interaction in the New Millenium*, ed. by J.M. Carroll, Addison-Wesley/ACM Press, Reading, Mass. (2001).

44. B.A. Ullmer, ‘‘Tangible Interfaces for Manipulating Aggregates of Digital Information,’’ Doctoral Dissertation, Massachusetts Institute of Technology (August 2002).

45. J. Underkoffler and H. Ishii, ‘‘Urp: A Luminous-Tangible Workbench for Urban Planning and Design,’’ *Proc. ACM CHI’99 Human Factors in Computing Systems Conference*, pp. 386-393, Addison-Wesley/ACM Press (1999).

46. S.A.G. Wensveen and C.J. Overbeeke, ‘‘Fun with Your Alarm Clock: Designing for Engaging Experiences through Emotionally Rich Interaction,’’ in *Funology: From Usability to Enjoyment*, ed. by M.A. Blythe, A.F. Monk, K. Overbeeke, and P.C. Wright, pp. 275-281, Kluwer Academic Publishers, Dordrecht (2003).

47. T. Winograd, ‘‘Interaction Spaces for 21st Century Computing,’’ in *Human-Computer Interaction in the New Millenium*, ed. by J.M. Carroll, Addison-Wesley/ACM Press, Reading, Mass. (2001).