

ARTIFACT AS THEORY-NEXUS: HERMENEUTICS MEETS THEORY-BASED DESIGN

John M. Carroll and Wendy A. Kellogg

IBM Watson Research Center
P.O.Box 704
Yorktown Heights, NY 10598

ABSTRACT

We suggest that HCI designs characteristically embody multiple, distinct psychological claims, that virtually every aspect of a system's usability is overdetermined by independent psychological rationales inherent in its design. These myriad claims cohere in being implemented together in a running system. Thus, HCI artifacts themselves are perhaps the most effective medium for theory development in HCI. We advance a framework for articulating the psychological claims embodied by artifacts. This proposal reconciles the contrasting perspectives of theory-based design and hermeneutics, and clarifies the apparent paradox of HCI application leading HCI theory.

KEYWORDS: theory, design, task analysis

As a field of inquiry, the study of human-computer interaction is perplexing. In the midst of enormous activity and considerable technical progress, very fundamental issues remain unresolved. For example, it would seem to be axiomatic that scientific psychology has much to contribute to an understanding of HCI phenomena and to the design of HCI artifacts. However, the role of scientific psychology in HCI is in dispute.

Some theorists argue that only certain fairly narrow conceptions of psychology can successfully be applied. Newell and Card [14] warn that psychology might be driven out of HCI unless it can provide quantitatively predictive cognitive models. This approach focuses of necessity on relatively low-level as-

pects of HCI (e.g., keystroke-level methods for ideal expert performance [2]), or on simplified HCI situations (e.g., rote learning of scaled-down text editors [17]). Its objective is to provide a psychological theory-base suitable for deductive use in HCI design [4, 15].

Other theorists hold that pursuing the goal of developing cognitive science theories of HCI may *impair* progress toward usefully understanding HCI phenomena and effectively contributing to design [20]. This approach stresses the distortion and oversimplification inherent in laboratory-bound psychology and in conventional views of theory-based design. In contrast, this hermeneutic approach recommends treating situations, users and artifacts as unique instances. Understanding such instances is seen as an interactive process of consensual interpretation: the goal is not to identify a theory-base for application to design, but to design through a subjective process of discovery [21].

Both approaches are problematic [3, 5]. The limited scope of quantitative theories precludes adequate grounding for design decisions. Such theory-based design has never occurred on a nontrivial scale. On the other hand, bridges from hermeneutic interpretation into design decision-making are essentially mystical. There is no systematic methodology, no conceptual framework, no explicit way to abstract from particular experiences.

Indeed, interface innovations frequently *lead* HCI research rather than following from it in the conventionally assumed flow of "technology transfer" from theory to application. Often, research returns again and again to the interpretation of particular artifacts and techniques. The evolution of the concept of "direct manipulation" as a psychological analysis of a collection of HCI artifacts provides an excellent example of this [13, 18]. The theoretical work substantially *postdates* the implementation of the concept in running systems [8, 19].

We seek to reconcile the contrasting perspectives of theory-based design and hermeneutics, and to confront the apparent paradox of HCI application lead-

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

ing HCI theory. We conjecture that successful HCI designs embody an assortment of psychological claims, that virtually every aspect of a system's usability is overdetermined by independent psychological rationales inherent in its design. Thus, we are urging a more *systematic* approach to usability interpretation than that countenanced by hermeneutics (namely, one grounded in scientific psychology), a far *richer* utilization of psychology in design than that afforded by current quantitative theories, and a more central role for the design and interpretation of HCI artifacts in HCI research.

In the balance of this brief paper, we illustrate how HCI artifacts embody psychological claims. Our argument, of course, cannot be demonstrative, but we hope to show how taking HCI artifacts more seriously can reconcile theory-based design and hermeneutics by enriching the vision of the former and disciplining that of the latter.

PSYCHOLOGICAL OVERDETERMINATION

HCI artifacts embody psychological claims in contexts of use: aspects of the interface engender psychological consequences and in this sense make claims about the user's behavior and experience. Accordingly, our analysis is couched in a framework for understanding an activity domain; we employ a generic task analysis of user activity into three arenas: identifying and clarifying a goal, planning and acting toward its achievement, and evaluating the results of action (this is a condensed variant of the seven stages of action described by Norman [16]).

Our objective in the analysis is to produce a rich qualitative description in which the claims embodied by various aspects of an interface are grouped under these categories of user activity. We acknowledge that such a description cannot be exhaustive; we seek merely to articulate the *leading claims* of an artifact.

This approach provides a means for exposing the psychological overdetermination of HCI artifacts. A designed artifact exhibits task *coverage* to the extent that the set of psychological claims it embodies addresses every activity category in the task analysis. Thus, an artifact embodying claims about user interaction in formulating appropriate goals, mapping them to system-defined goals, creating plans and executing their actions, and evaluating the results of those actions would have good task coverage.

An artifact has task *depth* to the extent that it embodies multiple claims about a particular category of user activity. A direct manipulation interface, as interpreted by Hutchins et al. [13], for example, makes several claims about planning and acting: smaller semantic distance makes plans simpler, smaller referential distance simplifies plans and their execution, the "point + operation" method facilitates learning by eliminating certain classes of syntactic error.

Artifacts often incorporate a diversity of techniques and component artifacts, each of which exhibits sig-

nificant task coverage and task depth. It is the composition and interaction of these claims that determine the artifact's manifest usability in the various arenas of user activity. Usability is psychologically overdetermined both by the individual claims embodied in specific techniques and component artifacts *and* by the *relations* among these parts and among the claims they embody. This is the sense in which we see HCI artifacts as a nexus of psychological theory: the myriad claims and their interrelations are given coherence through their codification in the artifact.

TRAINING WHEELS

A simple example of an HCI artifact is the Training Wheels interface, a reduced-function training environment for a stand-alone text editor [6, 7]. The key characteristic of this interface for the purpose of articulating its psychological claims is that the training wheels design "blocks" the consequences of problematic user selections. For example, if the first-time user selects Data Merging, a message is returned that the function is not available in the training wheels interface. This simple technique has been found to facilitate initial and continuing learning (i.e., learning advanced functions like Data Merging after the training wheels are removed).

There are a variety of psychological interpretations inherent in the training wheels design, a variety of specific and distinct psychological claims that inhere in the simple error blocking technique, as summarized in Table 1.

Claims Embodied in Training Wheels

Goals. Training Wheels embodies the claim that understanding real-world tasks in terms of system tasks is facilitated by filtering inappropriate goals. In the extreme upperleft of Table 1, this is expressed in the claim "working opportunistically facilitates goal identification." A user who has not yet identified an appropriate initial goal is blocked from prematurely engaging advanced functions like Data Merging. This implicitly guides the user toward identifying more appropriate goals like typing and printing documents. The user who has already articulated an appropriate goal is blocked from mapping it inappropriately to system functions. For example, documents must be Created before they are Printed, so the selection of Print before Create is blocked. This implicitly guides the user toward correct task-device mappings.

A user *can* make errors in the Training Wheels system, for example selecting Print before a Create. Only the *consequence* of errors is blocked, not the opportunity to commit them in the first place. The user goes through realistic decision making to determine a correct goal mapping. Any incorrect mapping is clearly flagged by error blocking but without further consequence. The claim is that this allows the goal mapping process, and particularly the incorrect goal mappings, to remain more salient in memory (for ex-

Table 1. Psychological Claims of Training Wheels Error Blocking

Goals	Planning/Acting	Evaluation
working opportunistically facilitates goal identification	less distraction from error recovery episodes focuses user's attention on correct actions	reduced device space constrains hypothesis generation
working on familiar tasks suggests goal mapping	less nested action sequences are more salient, facilitating plan formation	exposure to full menus supports incidental learning
blocking consequences while allowing commission of errors makes goal mappings more salient	practicing kernel scenarios establishes integrated basic skills	constraining user's intention increases effectiveness of feedback by allowing feedback on goal mapping
accomplishing familiar tasks is intrinsically motivating	blocking requested actions motivates further action	immediate feedback on blocked device states instigates evaluation while relevant intentions, plans and actions are still active in memory

ample, by not getting tangled with lengthy error recoveries) and to accordingly play a more effective role in learning.

The user of the Training Wheels interface accomplishes goals that are already meaningful outside the context of the system; the system embodies the claim that working on such goals is intrinsically motivating.

Planning/Acting. Blocking access to irrelevant functions purports to control the potential distraction of developing and pursuing erroneous plans. Because false starts are curtailed, action sequences are less deeply nested and relatively stereotyped across previous and subsequent attempts. This increases the chance that learners will notice what they are doing as they practice kernel scenarios (typing and printing). It becomes more likely that these action sequences will become established in the plan repertoire.

Training Wheels asserts that practicing kernel scenarios establishes basic skill components as parts of meaningful wholes. Thus, typing and printing scenarios incorporate menu selections, and the learner who practices typing and printing necessarily practices menu selection and experiences that basic skill as meaningful.

Training Wheels error blocking also embodies the claim that thwarting a user action will motivate further action. This claim could be true, but it could also be the case that blocking requested actions is frustrating for users, particularly when they do not fully understand the rationale for an action being blocked or if they experience a series of blocked attempts. Thus, the error blocking intervention might have at least one *negative* psychological consequence.

This example illustrates the contrast between a *claim* embodied by an artifact (about a psychological consequence) and the actual psychological consequence. By definition, a claim inherent in an artifact is an as-

sertion; it is part of the psychological rationale for the design. However, the claims embodied by an artifact are empirical claims, and it is finally an empirical issue whether the claims are true.

Evaluation. The reduction in the number of possible system states, in consequence of error blocking, admits of fewer possible explanations for actions and consequences in an episode, and hence constrains the user's evaluation of an interaction. Though advanced functions are blocked, users see them listed in the context of their menus. This minimal exposure to the system's full functionality seeks to support incidental learning of the scope of the full device space.

Guiding the user's goal identification makes it more likely that the user will have an appropriate goal in mind when error blocking messages are encountered. The claim is that the value of the messages can be enhanced if they allow the user to infer not just that an action is inappropriate, but that it is an inappropriate action *for a particular task*. Finally, presenting messages immediately when a system state transition is blocked seeks to provoke evaluation while the relevant goals, plans, and actions are still in active memory.

Training Wheels as Theory

The Training Wheels interface is psychologically overdetermined in making a variety of claims in the sense of task coverage and task depth. These diverse claims cohere in being implemented together in a running system, in interactive situations of real use. Thus many possible, indeed plausible, psychological claims are *not* embodied in the Training Wheels interface. For example, Training Wheels does not make the goal claim that learning is enhanced by free exploration (error blocking prevents free exploration). It does not make the planning/acting claim that mastering component skills separately before integrating

them in task performance is effective (a claim that is characteristic of "systems approach" instruction).

The coherence of the claims embodied in Training Wheels is highlighted by the Gedanken experiment of trying to excise or correct an empirically false claim. For example, suppose that the claim "blocking requested actions motivates further action" is just false; suppose that the predominant affective consequence of blocking requested actions for planning and acting is frustration. We could address this in design by preempting the selection of blocked functions through removing (or dimming) inappropriate menu items: the user would not have to even consider inappropriate functions, could never select them, and would never be frustrated by having a blocking message returned.

This possible alteration, however, has far broader impact on the psychological claims inherent in the Training Wheels interface than merely eliminating the problematic claim about frustration. Blocking deliberately planned actions, albeit actions incorrect with respect to a goal, embodies claims about goal mapping and evaluation (namely, that the process of goal mapping can become more salient to the user and that evaluation will occur when relevant intentions, plans and actions are still active in memory). The potential consequence of frustration cannot be subtracted from these other claims inhering in the blocking technique. The Training Wheels system is the nexus of this whole collection of claims, some of which may be false.

This is why user interface designers must work directly with actual HCI artifacts and why simplified theoretical descriptions are inadequate for this purpose. There is no context-free theory, for example in the sense of Newell and Card [14], that captures all of these claims. Our intention in constructing descriptions like that in Table 1 is to support the use of actual HCI artifacts in the design process by providing analytic guidance for *seeing* example artifacts in psychological terms.

However, training wheels error blocking is a single technique; a more typical case for analysis is that of a complete interface consisting of a variety of techniques, each potentially embodying several psychological claims.

HYPERCARD

Hypercard™ is an authoring tool and information storage and retrieval system running within the Apple® desktop environment. It employs the metaphor of a library card catalog [9], wherein personal database applications are represented as stacks of cards. Some of the key characteristics of Hypercard are its use of progressive disclosure of function by means of user levels, its use of example stacks and other built-in objects, and its use of the card catalog metaphor. In addition, Hypercard employs several techniques common to the desktop environment and

other applications, for example, menu dimming, direct manipulation, and standard interface components.

Our purpose here is not to exhaustively identify the claims inhering in Hypercard's interface, but to analyze some key characteristics of the interface in order to further illustrate overdetermination (Table 2).

User Levels

Hypercard provides five selectable user levels (browsing, typing, painting, authoring, and scripting). Progressing from browsing to scripting, menus are expanded, and new menus become available, disclosing new categories of function.

Labeled user levels embody the claim that the names of the levels can facilitate goal identification by suggesting the kinds of task goals that can be attempted at each level. For example, at the "browsing" level, the user can navigate card stacks, search for information in stacks, and print information, but is unable to alter information. The staging of function seeks to facilitate the mapping of tasks (e.g., finding information in a stack) into device goals (e.g., methods of navigating and searching). Menus and menu functions visible at the browsing level generally support *only* browsing tasks. As in Training Wheels, the claim is that goal mapping is facilitated because inappropriate choices are suppressed.

The staging of function provided by user levels also impacts the planning and acting task component. At each user level, a limited and coherent set of new functions is revealed. *Within* a level, this embodies the claim that the user is led to practice and consolidate kernel scenarios without the distraction of advanced and irrelevant functions. The user can also observe the covariation of available functions with user level labels *across* levels, to better understand how device goals are linked to plans and actions.

The progression of user levels incrementally expands the device space. This embodies the claim that the user's how-it-works evaluation of interactions is initially facilitated because there are fewer entities from which to construct explanations. A further claim is that successive user levels provide a framework for incrementally learning system concepts. For example, a learner is first exposed to buttons through using them to navigate, open stacks, etc. At the authoring level, buttons can be "borrowed" and incorporated into stacks the user is creating or modifying. Ultimately, at the level of scripting, the user learns to modify button actions by writing or rewriting button scripts.

Examples, Built-Ins

Hypercard provides example applications (or stackware™) and built-in objects (e.g., bitmaps, buttons) that users can incorporate into stackware they create. This palette of concrete possibilities establishes a task orientation, cognitively and

Table 2. Psychological Claims of Some Key Hypercard Techniques

	Goals	Planning/Acting	Evaluation
User Levels	labeled staging of system function facilitates goal identification working on staged tasks facilitates goal mapping	staging kernel scenarios integrates basic skill staged function and label covariation helps link plans to goals	reduced device space constrains hypothesis generation staged function supports incremental learning
Examples, Built-ins	a range of concrete task instances establish a task orientation concrete, meaningful objects suggest goals examples facilitate goal mappings	using built-in parts simplifies plans modifying is easier than creating from scratch	manipulating meaningful, familiar objects suggests hypotheses learning by doing is superior to being told
Card Catalog Metaphor	database-as-stack simplifies goal mapping		prior knowledge framework facilitates explanations
Menu Dimming		preempting syntactic errors reduces the amount of syntactic knowledge to be learned backgrounding syntactic details reduces level of awareness of syntax preempting syntactic errors avoids frustration	visual distinction between active and inactive menu items implicitly poses evaluative questions exposure to full menus supports incidental learning

motivationally, toward the class of applications that Hypercard can support (e.g., address book, to-do list, calendar, labels for VCR tapes and cassettes). It embodies the claim that the user will be cognizant of and motivated toward appropriate goals.

Hypercard also embodies the claims that particular examples and built-ins can suggest specific task goals and goal mappings into system functions. For example, the Address Stack provides a built-in template for address book applications. A user who does not yet understand Hypercard but understands address books can identify the goals of browsing the built-in template or of adding further information slots to it. Working examples can provide an inspectable model of how given goals are achieved in Hypercard (e.g., how to make something shared across all cards in a stack).

Examples and built-ins assert that planning and acting is simplified by being able to directly incorporate appearance and behavior of components (e.g., a button and its function). A corollary claim is that modifying is easier than creating from scratch: users can borrow a button that already behaves in the desired way, but modify its appearance. Alternatively, a button that already has some of the behavior the user desires (e.g., dialing a phone number) can be copied and its script augmented to include additional behavior (e.g., taking the user to a notepad stack to create a record of the conversation).

Examples and built-ins also embody claims about learning by exploration and analysis. Seeing inventories of buttons, fields, and cards suggests how the system taxonomizes the world, what it can do and be used for. For example, the user who browses the backgrounds of cards in various stacks can analyze how different stacks are composed. The system embodies the claim that active evaluation, learning by doing, is superior to being told explicitly.

Card Catalog Metaphor

Hypercard incorporates a "card catalog" interface metaphor. Stacks are likened to a library's card catalog, cards composed of shared information (backgrounds) and card-specific information. Only one stack can be open at a time, as the library user typically uses one catalog at a time. Links between cards can be created as if placing a "magic button" on a card that would put another related card into the hands of the user [9].

The metaphor embodies the claim that the user's prior understanding of the storage and retrieval functions of card catalogs can suggest appropriate goals and goal-mappings. Device goals for various database applications are couched in terms of the entities of the metaphor. For example, the distinction between background (e.g., blocks for call numbers) and card-

unique information suggests a decomposition for the goal of creating a new stack.

The card metaphor claims that evaluation is facilitated by identifying system objects with and encouraging explanations in terms of real-world objects. New functionality (e.g., buttons that link cards) can be more readily understood if it can be grafted onto the metaphor (i.e., a "magic button").

Note that the metaphor seems *not* to make claims vis-a-vis the planning and acting task component. It has been suggested that this is a general property of interface metaphors [10].

Menu Dimming

Hypercard employs several techniques common to the Apple® desktop interface [1], including a form of error blocking in the technique of menu dimming. Menu items that are invalid in the current system state appear in a greyed-out font; valid choices appear in black. In Hypercard this technique incorporates syntactic constraints, for example, object selection prerequisites for requesting an action. As such, most of the claims embodied by menu dimming pertain to planning and acting.

Menu dimming blocks certain syntactic errors by *preempting* them; the user is unable to select a menu function until a syntactically appropriate argument has been selected. For example, a user who wishes to cut text from a card must first select the text block; "Cut Text" is greyed-out in the Edit menu until a selection occurs. This dimming technique makes claims about learning, use and affect.

It asserts that learning can be facilitated if users need not explicitly remember syntactic constraints, reducing the amount of knowledge required for acting [13]. It asserts that fluid use is facilitated when users can remain less attentive to syntactic detail. And it asserts that preempting the possibility of syntactic error avoids the frustration that accompanies having an action thwarted.

Hypercard's menu dimming makes other claims associated with error management techniques (it controls the distractions of error recovery and affords less nested action sequences). However, these claims seem less central to the dimming technique since its error management addresses only syntactic well-formedness and not, for example, the appropriateness of the user's goals and tasks. Even if it were possible to select "Cut Text" without having first specified the text block to be cut, any failed attempt would necessarily be shallow (in terms of the action sequence) and short-lived. This contrasts with the deeply nested sequences and tangled recovery episodes entrained by inappropriate task-device mappings.

Menu dimming makes the claim that the visual contrast between active (black) and inactive (greyed-out) menu items implicitly poses evaluative questions to the user. The user who expects a menu item to be

available, and finds it is not, may ask why it is not available. More generally, the user may be led to consider what differentiates menu items that are available in a particular system state from those which are not available. Like Training Wheels, the technique also embodies the claim that exposing the user to full menus supports incidental learning of the range of the system's functionality.

Hypercard's menu dimming does not seem to make any goal claims. The technique incorporates only syntactic constraints.

Hypercard as Theory

Hypercard incorporates a variety of interface techniques, each embodying a variety of psychological claims which complement and interact with the other claims embodied in Hypercard. For example, we have noted that the card catalog metaphor seems not to make planning and acting claims, and that menu dimming seems not to make goal claims. However, in both cases the variety of claims embodied in other Hypercard interface techniques compensate for this limitation in task coverage, and produce an overall balance in coverage and depth.

Of course, this is, as in the case of Training Wheels, more than a matter of merely adding and deleting claims and techniques. For example, many of the apparently separate interface techniques in Hypercard work together to project a "model world" to the user [13], a "believable illusion" [12] in which the user thinks in terms of directly interacting with objects that behave according to their natures (this in contrast to a more traditional orientation in which the user is to think in terms of typing commands to a processor). The model world itself can be viewed as an interface technique, perhaps embodying emergent claims not made by any of its constituents.

The gestalt of a model world depends upon the support and interaction of many techniques. In Hypercard, the use of examples and built-ins and the card catalog metaphor are important components of the model world. Other techniques that contribute to this gestalt (ones we have not discussed) are direct manipulation and automatic saving of changes. We do not fully understand how these separate techniques combine to project a model world; we do not know the boundary conditions for achieving this gestalt, but we know that the constituent techniques — and the claims they embody — cannot merely be added, altered or substituted for without considering their systemic interaction.

The specific nature of any given Hypercard technique is constrained by its being implemented in the context of all the other techniques. For example, it is possible to imagine a variation of menu dimming that would support identifying and mapping goals: as in Training Wheels, task order constraints could be used to determine when menu selections are dimmed. However, this seemingly local change would have

pervasive implications. Hypercard is not a training system; incorporating task order constraints would limit the scope of user activity (perhaps entraining new claims about frustration, for example) and undermine existent techniques (e.g., user control).

Moreover, altering menu dimming with respect to user goal activities may not be necessary given other Hypercard techniques that support goal identification and mapping in the system as implemented (e.g., user levels and other techniques we have not discussed). The role of menu dimming as implemented in Hypercard must be evaluated with respect to all the other Hypercard techniques and their interactions.

Menu dimming is also one of several Hypercard techniques included in the Apple® desktop interface. This relationship provides further systemic constraints on menu dimming as a technique in Hypercard, in particular on its claims about consistency.

HCI ARTIFACTS AS PSYCHOLOGICAL THEORIES

In any applied realm there is a tension between science and design. Science and design have different principal objectives. It may be simplistic to imagine deductive relations from science to design, but it would be bizarre if there were no relation at all. We believe that theory-based design is possible in HCI with a sufficiently rich view of artifact as theory-nexus. However, the property of psychological overdetermination discourages hope for simple, deductive bridges from theory into design. Rather, we envision a reciprocal relation between the articulation and rearticulation of a set of psychological claims and the iterations of design.

Hammond et al. [11] illustrate the type of problem that arises when psychological claims are embodied in artifacts on a one-off basis. They describe an instructional exercise designed to teach users about typed input fields, incorporating the claim that feedback can guide the evaluation of input errors. The psychological interpretation of the design was inadequately articulated, however: unavoidable delays in the feedback presentation embodied the further (and perhaps unrecognized) claim that delayed feedback would still be effective. In fact, the tutorial exercise proved quite problematic for users.

We see this kind of example as routine in HCI, and indeed as routine in complex empirical domains. Physicians do not mechanically map isolated, quantitative indicators into states of health and pathology. But the rather complex interpretations and intervention strategies they design are informed by a systematic medical science. Understanding usability is probably no more and no less complex a problem, and we would be foolish to expect any less complex a solution.

The hermeneutic vision is correct in stressing the multiplicity of relevant interpretations of situations, users and artifacts, but too easily conflates multiplicity

and infinity, settling for indeterminate subjectivity. Our view is more disciplined in assuming that there are bounds on interpretations (i.e., they are grounded in psychology and made with respect to a task analysis) and that interpretations are valuable insofar as they produce systematic and falsifiable results.

The set of claims inherent in typical computer equipment may be rich, but it is not unbounded or arbitrary. The psychological interpretation of an artifact need not begin *de novo* for each new artifact. This is manifest in our two examples, the interpretations of which have several points of contact, though they were not preselected for this. Again, this state of affairs seems typical of complex domains: medical diagnosis to a considerable extent proceeds case-by-case, but also improves with every case study. The complexity of HCI may cause us to reject simplistic science, but it need not incline us to reject science *tout court*.

To the contrary, interpretations of the claims embodied by HCI artifacts evince the hallmarks of scientific progress, at least insofar as we can determine this now. The claims embodied in artifacts are absolutely specific; they are always already instantiated in the world. It is incumbent on the analyst to faithfully extract a system of testable empirical claims. For example, some of the claims we discussed as embodied in the Training Wheels interface have been empirically tested [6, 7].

Interpretations of artifacts develop like other theoretical expressions. The interpretation of the direct manipulation technique, the most well-worked example to date, clearly demonstrates theory development as one contrasts, for example, an early interpretation [18], with a subsequent interpretation [13]. This progress has come about through a commitment to science, though not to simplistic and mechanical visions of science.

We have tried to sketch an approach that reconciles and improves upon conventional theory-based design and hermeneutics. The essence of this approach is to view artifacts not through the filter of an isolated theoretical abstraction (e.g., a grammar-in-the-head) nor, without abstraction, as an unbounded collection of idiosyncratically interpreted, specific instances, but to recognize and analyze the multiple, simultaneous psychological claims and theories embodied by the artifact.

We are developing the framework of Tables 1 and 2 as a design tool for systematically articulating usability issues and concerns. It is not a means for simplistically running two techniques or two designs against one another on a single dimension like task time or error rate. It is a framework for carrying out an analytic and empirical investigation, for *seeing* a design in psychological terms. Nevertheless, we must stress that in this first attempt our analyses are bounded not only by psychological theories, but by the limits of our own imaginations and skills.

The remedy for this lies in further developing what we see as an important new role for psychologists working in HCI: interpreting artifacts — both during and after design. Articulating the system of theories inherent in HCI designs is not something that is commonly done, but it could be. It took decades for psychological interpretations of direct manipulation to begin to appear — and the interpretation process is not yet complete. Providing interpretations of artifacts is a new role and an important opportunity for psychology in HCI.

Our working hypothesis is that *something* like our framework for interpreting artifacts must be right. We need not choose between no science at all and an ineffective science that postures without offering real sustenance to designers. Rather, we must now develop specific proposals for how a rich and relevant science of HCI can exist. In the end, this endeavor may be as salutary for basic psychology as for its more applied progeny.

ACKNOWLEDGEMENTS

We thank the three anonymous reviewers, members of the UII reading group, and Rachel Bellamy, John Bennett, Robert Campbell, John Richards, Mary Beth Rosson, Linda Tetzlaff and John Thomas for particularly useful comments.

REFERENCES

1. Apple Computer, Inc. (1987). *Human interface guidelines: The Apple desktop interface*. Reading, MA: Addison-Wesley.
2. Card, S.K., Moran, T.P. and Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
3. Carroll, J.M. (1989). Evaluation, description and invention: Paradigms for human-computer interaction. To appear in M.C. Yovits (Ed.), *Advances in computers, Volume 29*. New York: Academic Press.
4. Carroll, J.M. and Campbell, R.L. (1986). Softening up Hard Science: Reply to Newell and Card. *Human-Computer Interaction, 2*, 227-249.
5. Carroll, J.M. and Campbell, R.L. (1988). Artifacts as psychological theories: The case of human-computer interaction. IBM Research Report RC 13454. Yorktown Heights, NY. To appear in *Behaviour and Information Technology*.
6. Carroll, J.M. and Carrithers, C. (1984). Blocking learner errors in a training wheels system. *Human Factors, 26*, 377-389.
7. Catrambone, R. and Carroll, J.M. (1987). Learning a word processing system with training wheels and guided exploration. In J.M. Carroll and P.P. Tanner (Eds.) *Proceedings of CHI + GI'87: Conference on Human Factors in Computing and Graphics Interface*. New York: ACM, pages 169-173.
8. Engelbart, D. and English, W. (1968). A research center for augmenting human intellect. *Proceedings of Fall Joint Computer Conference, 33(1)*. Montvale, NJ: AFIPS Press, pages 395-410.
9. Goodman, D. (1987). *The complete Hypercard handbook*. New York: Bantam Books.
10. Halasz, F., & Moran, T. (1982). Analogy considered harmful. *Proceedings of the Conference on Human Factors in Computer Systems*, Gaithersburg, MD, National Bureau of Standards.
11. Hammond, N., Morton, J., MacLean, A., Barnard, P., and Long, J. (1982). Knowledge fragments and users' models of systems. *Hursley Human Factors Laboratory Report HF071*, IBM Corporation, Hursley Park, U.K.
12. Heckel, P. (1984). *The elements of friendly software design*. New York: Warner Books.
13. Hutchins, E., Hollan, J., & Norman, D.A. (1986). Direct manipulation interfaces. In D.A. Norman and S. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
14. Newell, A. and Card, S.K. (1985). The prospects for psychological science in human-computer interaction. *Human-Computer Interaction, 1*, 209-242.
15. Newell, A. and Card, S.K. (1986). Straightening out softening up: Response to Carroll and Campbell. *Human-Computer Interaction, 2*, 251-267.
16. Norman, D.A. (1986). Cognitive engineering. In D.A. Norman and S. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
17. Polson, P., Kieras, D., and Muncher, E. (1987). Transfer of skill between inconsistent editors. Microelectronics and Computer Technology Corporation Technical Report ACA-HI-395-87, Austin, Texas.
18. Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology, 1*, 237-256.
19. Sutherland, I.E. (1963). Sketchpad: A man-machine graphical communication system. *Proceedings of the Spring Joint Computer Conference, 23*. AFIPS Press: Montvale, NJ, pages 329-346.
20. Whiteside, J. and Wixon, D. (1987). Improving human-computer interaction -- a quest for cognitive science. In J.M. Carroll (Ed.) *Interfacing thought: Cognitive aspects of human-computer interaction*. Cambridge, MA: Bradford/MIT Press, pages 337-352.
21. Winograd, T. and Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex.