Supporting Situated Interpretation

Gerry Stahl
Department of Computer Science,
School of Environmental Design,
and Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0430
gerry@cs.colorado.edu

Abstract

This paper discusses the role of interpretation in innovative design and proposes an approach to providing computer support for interpretation in design.

According to situated cognition theory, most of a designer's knowledge is normally tacit. Situated interpretation is the process of explicating something that is tacitly understood, within its larger context.

The centrality of interpretation to non-routine design is demonstrated by: a review of the design methodology of Alexander, Rittel, and Schön; a protocol analysis of a lunar habitat design session; and a summary of Heidegger's philosophy of interpretation. These show that the designer's articulation of tacit knowledge takes place on the basis of an understanding of the design situation, a focus from a particular perspective, and a shared language.

As knowledge is made explicit through the interpretive processes of design it can be captured for use in computer-based design support systems. A prototype software system is described for representing design situations, interpretive perspectives, and domain terminology to support interpretation by designers.

The Need for Computer Support

The volume of information available to people is increasing rapidly. For many professionals this means that the execution of their jobs requires taking into account far more information than they can possibly keep in mind. Consider the lunar habitat designers who serve as a key example in this paper. In working on their high-tech design tasks, they must take into account architectural knowledge, ergonomics, space science, NASA regulations, and lessons learned in past missions. Computers seem necessary to store these large amounts of data. However, the problem is how to capture and encode information relevant to novel future tasks and how to present it to designers in formats that support their mode of work.

A framework for clarifying the respective roles for computers and people in tasks like lunar habitat design is suggested by the theory of *situated cognition*. Several influential recent books (e.g., Schön, 1983; Winograd & Flores, 1986; Suchman, 1987; Ehn, 1988; Dreyfus, 1991) argue that human cognition is fundamentally different from computer manipulations of formal symbol systems. These differences imply that people need to retain control of the processes of non-routine design, but that computers can provide valuable computational, visualization, and external memory aids for the designers and support interpretation by them.

From the viewpoint of situated cognition, the greatest impediment to computer support of innovative design is that designers make extensive use of *tacit* knowledge while computers can only use *explicit* representations of information. This paper discusses the role of tacit understanding in designing, in order to motivate an approach to computer support of design. It focuses on three themes: (a) the need to represent novel design *situations*; (b) the importance of viewing designs from multiple *perspectives*; and (c) the utility of formulating tacit knowledge in explicit *language*.

The following sections discuss how these three themes figure prominently in analyses of **interpretation in design methodology** and in a study of **interpretation in lunar habitat design**. Following a discussion of the **tacit basis of understanding**, the **philosophy of interpretation** defines interpretation as the articulation of tacit understanding. Then consequences for **computer support for interpretation** are drawn, and they are illustrated by **the HERMES system**, a prototype for supporting interpretation in the illustrative task of lunar habitat design.

Interpretation in Design Methodology

The centrality of interpretation to design can be seen in seminal writings of design methodologists. The following summaries highlight the roles of appropriate representations of the design situation, alternative perspectives, and linguistic explications of tacit understanding within the processes of interpretation in design.

Alexander (1964) pioneered the use of computers for designing. He used them to compute diagrams or patterns that decomposed the structural dependencies of a given problem into relatively independent substructures. In this way, he developed explicit interpretations for understanding a task based on an analysis of the unique design *situation*.

For Rittel (1973), the heart of design is the deliberation of issues from multiple *perspectives*. Interpretation in design is "an argumentative process in the course of which an image of the problem and of the solution emerges gradually among the participants, as a product of incessant judgment, subjected to critical argument" (p.162). Rittel's idea of using computers to keep track of the various issues at stake and alternative positions on those issues led to the creation of issue-based information systems.

Schön (1983) argues that designers constantly shift perspectives on a problem by bringing various professionally trained tacit skills to bear, such as visual perception, graphical sketching, and vicarious simulation. By experimenting with tentative design moves within the tacitly understood situation, the designer discovers consequences and makes aspects of the structure of the problem explicit. Certain features of the situation come into focus and can be named or characterized in *language*. As focus subsequently shifts, what has been interpreted may slip back into an understanding that is once more tacit, but is now more developed.

Interpretation in Lunar Habitat Design

As part of an effort at developing computer support for lunar habitat designers, thirty hours of design sessions were videotaped and analyzed (see Stahl, forthcoming). The specified task was to accommodate four astronauts for 45 days on the moon in a cylindrical module 23 feet long and 14 feet wide.

A protocol analysis of the designers' activities shows that much of the design time consisted of processes of *interpretation*, i.e., the explication of previously tacit understanding. As part of this interpretation, representations were developed for describing pivotal features of the design situation that had not been included in the original specification; perspectives were evolved for looking at the task; and terminology was defined for explicitly naming, describing, and communicating shared understandings.

The designers felt that a careful balance of public and private space would be essential given the crew's long-term isolation in the habitat. An early design sketch proposed private crew areas consisting of a bunk above a workspace for each astronaut. Space constraints argued against this. The traditional conception of private space as a place for one person to get away was made explicit and criticized as taking up too much room. As part of the interpretive designing process, this concept was revised into a reinterpretation of privacy as a gradient along the habitat from quiet sleep quarters to a public activity area. This notion of degrees of privacy permitted greater flexibility in designing.

In another interchange related to privacy, the conventional American idea of a bathroom was subjected to critical deliberation when it was realized that the placement of the toilet and that of the shower were subject to different sets of constraints based on life in the habitat. The tacit acceptance of the location of the toilet and shower together was made explicit by comparing it to alternative European perspectives. The revised conception permitting a separation of the toilet from the shower facilitated a major design reorganization.

In these and other examples, the designers needed to revise their representations for understanding the design *situation*. They went from looking at privacy as a matter of individual space to reinterpreting the whole interior space as a continuum of private to public areas.

The conventional American notion of a bathroom was compared with other cultural models and broken down into separable functions that could relate differently to habitat usage patterns. Various *perspectives* were applied to the problem, suggesting new possibilities and considerations. Through discussion, the individual perspectives merged and novel solutions emerged.

In this interpretive process, previously tacit features of the design became explicit by being named and described in the *language* that developed. For instance, the fact that quiet activities were being grouped toward one end of the habitat design and interactive ones at the other became a topic of conversation at one point and the terminology of a "privacy gradient" was proposed to clarify this emergent pattern.

The Tacit Basis of Understanding

Situated cognition theory disputes the prevalent view that all human cognition is based on explicit mental representations such as goals and plans. Winograd and Flores (1986) hold that "experts do not need to have formalized representations in order to act" (p.99). Although manipulation of such representations is often useful, there is a background of preunderstanding that cannot be fully formalized as explicit symbolic representations subject to rule-governed manipulation. This tacit preunderstanding underlies people's ability to understand representations when they do make use of them. Suchman (1987) concurs that goals and plans are secondary phenomena in human behavior, usually arising only after action has been initiated: "when situated action becomes in some way problematic, rules and procedures are explicated for purposes of deliberation and the action, which is otherwise neither rule-based nor procedural, is then made accountable to them" (p.54).

Philosophers like Polanyi (1962), Searle (1980), and Dreyfus (1991) suggest a variety of reasons why tacit preunderstanding cannot be fully formalized as data for computation. First, it is too vast: background knowledge includes bodily skills and social practices that result from immense histories of life experience and that are generally transparent to us. Second, it must be tacit to function: we cannot formulate, understand, or use explicit knowledge except on the basis of necessarily tacit preunderstandings.

This is not to denigrate conceptual reasoning and rational planning. Rather, it is to point out that the manipulation of formal representations alone cannot provide a complete model of human understanding. Rational thought is an advanced form of cognition that distinguishes humans. Accordingly, an evolutionary theorist of consciousness such as Donald (1991) traces the development of symbolic thought from earlier developmental stages of tacit knowing, showing how these earlier levels persist in rational human thought as the necessary foundation for advanced developments, including language, writing, and computer usage.

The most thorough formulation of a philosophical foundation for situated cognition theory is given by Heidegger (1927), the first to point out the role of tacit preunderstanding and to elaborate its implications. For Heidegger, we are always knowledgeably embedded in our world; things of concern in our situations are already meaningful in general before we engage in cognitive activity. We know how to behave without having to think about it. For instance, an architect designing a lunar habitat knows how to lift a pencil and sketch a line, or how to look at a drawing and see the rough relationships of various spaces pictured there. The architect understands what it is to be a designer, to critique a drawing, to imagine being a person walking through the spaces of a floor plan.

Heidegger defines the *situation* as the architect's context—including the physical surroundings, the available tools, the circumstances surrounding the task at hand, and the architect's own personal or professional aims. The situation constitutes a network of significance in terms of which each part of the situation is already meaningful (see Stahl, 1975). That is, the architect has tacit knowledge of the situation as a whole; if something becomes a focus for the architect, it is perceived as already understood and its meaning is defined by its relations to the rest of the situation.

To the architect, a rectangular arrangement of lines on a piece of paper is not perceived as meaningless lines, but, given the design situation, it is already understood as a bunk for astronauts. The bunk is implicitly defined as such by the design task, the shared intentions of the design team, the other elements of the design, the sense of space conveyed by the design, and so on indefinitely. This network of significance is background knowledge that allows the architect to think about features of the design, to make plans for changes, and to discover problems or opportunities in the evolving design. At any given moment, the background is already tacitly understood and does not need to be an object of rational thought manipulating symbolic representations.

At some point the architect might realize that the bunk is too close to a source of potential noise, like the flushing of the toilet. The explicit concern about this physical adjacency arises and becomes something important against the background of relationships of the preunderstood situation. Whereas a commonsensical view might claim that the bunk and toilet were already present and therefore their adjacency was always there by logical implication, Heidegger proposes a more complex reality in which things are ordinarily hidden from explicit concern. In various ways, they can become uncovered and discovered, only to re-submerge soon into the background as our focus moves on.

In this way, our knowledge of the world does not consist primarily in mental models that represent an objective reality. Rather, our understanding of things presupposes a tacit preunderstanding of our situation. Only as situated in our already interpreted world can we discover things and construct meaningful representations of them. Situated cognition is not a simplistic theory that claims our knowledge lies in our physical environment like words on a sign post: it is a sophisticated philosophy of interpretation.

The Philosophy of Interpretation

Human understanding develops through interpretive explication. According to Heidegger, interpretation provides the path from tacit, uncritical preunderstandings to reflection, refinement, and creativity. The structure of this process of interpretation reflects the inextricable coupling of the interpreter with the situation, i.e., of people with their worlds. Our situation is not reducible to our preunderstanding of it; it offers untold surprises, which may call for reflection, but which can only be discovered and comprehended thanks to our preunderstanding. Often, these surprise occasions signal *breakdowns* in our skillful, transparent behavior, although we can also make unexpected discoveries in the situation through conversation, exploration, natural events, and other occurrences.

A discovery breaks out of the preunderstood situation because it violates or goes beyond the network of tacit meanings that make up the preunderstanding of the situation. To understand what we have discovered, we must explicitly *interpret* it *as* something, as having a certain significance, as somehow fitting into the already understood background. Then it can merge into our comprehension of the meaningful situation and become part of the new background. Interpretation of something *as* something is always a reinterpretation of the situated context.

For instance, the lunar habitat designers discovered problems in their early sketches that they interpreted as issues of privacy. Although they had created the sketches themselves, they were completely surprised to discover certain conflicts among the interactions of adjacent components, like the bunks and the toilet. Of course, the discoveries could only occur because of their understanding of the *situation*, represented in their drawings. The designers paused in their sketching to discuss the new issues. First they debated the matter from various *perspectives*: experiences of previous space missions, cultural variations in bathroom designs, technical acoustical considerations. Then they considered alternative conceptions of privacy, gradually developing a shared *vocabulary* that guided their revisions and became part of their interpretation of their task. They reinterpreted their understanding of privacy and represented their new view as a "privacy gradient."

These themes of representing the situation, changing perspectives, and using explicit language correspond to the three-fold structure of interpretation in Heidegger's philosophy. He articulates the preconditions of interpretation as: (a) *prepossession* of the situation as a network of preunderstood significance; (b) *preview* or expectations of things in the world as being structured in certain ways; and (c) *preconception*, a language for expressing and communicating.

In other words, interpretation never starts from scratch or from an arbitrary assignment of representations, but is an evolving of tentative preunderstandings and anticipations. One necessarily starts with sets of "prejudices" that have been handed down historically; the interpretive process allows one to reflect upon these preunderstandings methodically and to refine new meanings, perspectives, and terminologies for understanding things more appropriately.

Computer Support for Interpretation

The theory of situated cognition and the philosophy of interpretation stress how different human understanding is from computer manipulations of arbitrary symbols. These theories suggest the approach of *augmenting* (rather than automating) human intelligence. According to this approach, software can at best provide computer representations for people to interpret based on their tacit understanding of what is represented.

Representations used in computer programs must be carefully structured by human programmers who understand the task being handled thoroughly, because the computer itself simply follows the rules it has been given for manipulating symbols, with no notion of what these symbols represent. People who understand the domain must codify their background knowledge into software rules sufficiently to make the computer algorithms generate results that will be judged correct when interpreted by people. Only if a domain can be strictly delimited and its associated knowledge exhaustively reduced to rules, can it be completely automated.

Many tasks like lunar habitat design that call for computer support do not have such strictly delimited domains with fully catalogued and formalized knowledge bases. These domains may require exploration of problems never before considered, assumption of creative viewpoints, or formulation of innovative concepts. Software to support designers in such tasks should provide facilities for the creation of new representations and flexible modification of old ones. As the discussion of Alexander emphasized, the ability to develop appropriate representations dynamically is critical. Because they capture understandings of the *situation* that evolve through processes of interpretation, representations need to be modifiable during the design process itself and cannot adequately be anticipated in advance or provided once and for all.

The concept of an objective, coherent body of domain knowledge is misleading. As Rittel said, non-routine design is an argumentative process involving the interplay of unlimited perspectives, reflecting differing and potentially conflicting technical concerns, personal idiosyncrasies, and political interests. Software to support design should capture these alternative deliberations on important issues, as well as document specific solutions. Furthermore, because all design knowledge may be relative to perspectives, the computer should be used to define a network of over-lapping *perspectives* by which to organize issues, rationale, sketches, component parts, and terminology.

As Schön emphasized, interpretive design relies on moving from tacit skills to explicit conceptualizations. Additionally, design work is inherently communicative and increasingly collaborative, with high-tech designs requiring successive teams of designers, implementors, and maintainers. Software to support collaborative design should provide a *language* facility for designers to develop a formal vocabulary for expressing their ideas, for communicating them to future collaborators, and for formally representing them within computer-executable software. An end-user language is needed that provides an extensible domain vocabulary, is usable by non-programmers, and encourages reuse and modification of expressions.

Heidegger's analysis of interpretation suggests that most of the information that would be useful to designers may be made explicit at some moment of interpretation during designing. One strategy for accumulating a useful knowledge base is to have the software capture knowledge that becomes explicit while the software is being used. As successive lunar habitats are designed on a system, issues and alternative deliberations can accumulate in its issue base; new perspectives can be defined containing their own modifications of terminology and critic rules; the language can be expanded to include more domain vocabulary, conditional expressions, and query formulations. In this way, potentially relevant information is captured in formats useful for designers, because it is a product of human interpretation.

This is an evolutionary, bootstrap approach, where the software can not only support individual design projects, but simultaneously facilitate the accumulation of expertise and viewpoints in open-ended, exploratory domains. This means that the software should make it easy for designers to formalize their knowledge as it becomes explicit, without requiring excessive additional effort. The software should reward its users for increasing the computer knowledge base by performing useful tasks with the new information, like providing documentation, communicating rationale, and facilitating reuse and modification of relevant knowledge.

The HERMES System

In Greek mythology, Hermes supported human interpretation by providing the gift of spoken and written language and by delivering the messages of the gods. A prototype software system named HERMES has been designed to support the preconditions of interpretation (a) by representing the design construction situation for *prepossession*, (b) by providing alternative perspectives for *preview*, and (c) by including an end-user language for *preconception*.

It supports tacit knowing by encapsulating (a) mechanisms for analyzing design situations using interpretive critics (Fischer, et al., 1993), (b) alternative sets of information organized in named perspectives (Stahl, 1993), and (c) hypermedia computations expressed in language terms (Stahl, et al., 1992). In each of these cases, the hidden complexities can be made explicit upon demand, so the designer can reflect upon the information and modify (reinterpret) it.

HERMES is a knowledge-representation substrate for building computer-based design assistants (like that in Figure 1). It provides various media for designers to build formal representations of design knowledge. The hypermedia network of knowledge corresponds to the

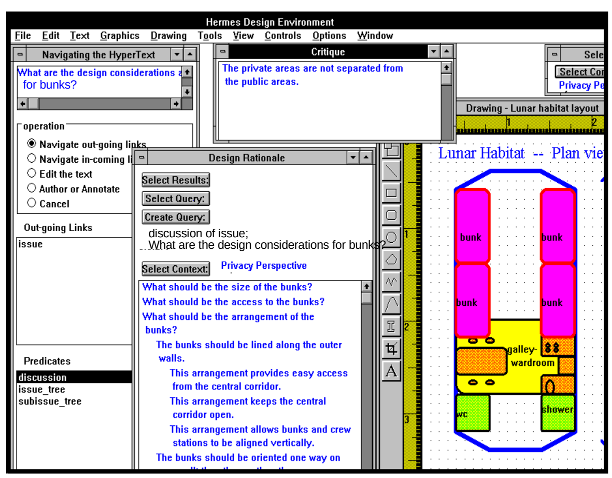

Figure 1. A view of the HERMES design environment, showing (left to right) a dialogue for browsing, a view of the issue base, a critic message, a construction area, and a button for changing interpretive perspectives.

design *situation*. Nodes of the knowledge representation can be textual statements for the issue base, CAD graphics for sketches, bitmap images to illustrate ideas, sound for audio commentary, or language expressions for critics and queries.

HERMES supports the collaborative nature of design by multiple teams through its *perspectives* mechanism. This allows users to organize knowledge in the system into over-lapping collections. Drawings, definitions of domain terms in the language, computations for critic rules, and annotations in the issue base can be grouped together for a project, a technical specialty, an individual, a team, or an historical version. Every action in HERMES takes place within some defined perspective, which determines what versions of information are currently accessible.

The HERMES *language* pervades the system, defining mechanisms for browsing, displaying, and critiquing all information. This means that designers can refine the representations, views, and expressions of all forms of domain knowledge in the system. Vocabulary in the language is modifiable and every expression can be encapsulated by a name. The syntax is English-like, in an effort to make statements in the language easily interpretable. The language is declarative, so users need not be bothered with explicit sequential programming concerns. Combined with the perspectives mechanism, the language permits designers to define and refine their own interpretations. This allows the HERMES substrate to support multiple situated interpretations.

Conclusion

The theory of situated cognition argues that only people's tacit preunderstanding can make data meaningful in context. Neither people nor computers alone can take advantage of huge stores of data; such information is valueless unless designers use it in their interpretations of design situations. The data handling capabilities of computers should be used to support the uniquely human ability to understand. The philosophy of interpretation suggests that several aspects of human understanding and collaboration can be supported with mechanisms like those in HERMES for refining representations of the design situation, for creating alternative perspectives on the task, and for sharing linguistic expressions. Together, situated cognition theory and Heidegger's philosophy of interpretation provide a theoretical framework for a principled approach to computer support for designers' situated interpretation in the information age.

Acknowledgments

This paper is indebted to Gerhard Fischer, Ray McCall, Kumiyo Nakakoji, Jonathan Ostwald, and Tamara Sumner, who helped to focus it, and to the HCC research group at the University of Colorado generally, whose ideas and systems it tries to ground theoretically and extend practically.

References

Alexander C (1964) *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Donald M (1991) *Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition*. Cambridge: Harvard University Press.

Dreyfus H (1991) *Being-in-the-World: A Commentary on Heidegger's Being and Time, Division I*. Cambridge: MIT Press.

Ehn P (1988) *Work-Oriented Design of Computer Artifacts*. Stockholm: Arbetslivscentrum.

Fischer G, Nakakoji K, Ostwald J, Stahl G, Sumner T (1993) Embedding Computer-Based Critics in the Contexts of Design. In *Proc. of InterCHI* '93.

Heidegger M (1927) *Sein und Zeit*. Tübingen: Niemeyer. Translation: Heidegger M (1962) *Being and Time*. New York: Harper & Row.

Polanyi M (1962) *Personal Knowledge*. London: Routledge & Kegan Paul.

Rittel H, Webber M (1973) Dilemmas in a General Theory of Planning. *Policy Science*. *4*: 155-169.

Searle J (1980) Minds, Brains, and Programs. *The Behavioral and Brain Sciences*. *3*.

Schön D (1983) *The Reflective Practitioner*. New York: Basic Books.

Stahl G (1975) Hermeneutics and Social Theory: Interpreting and Transforming Our World. Ph.D. Diss., Dept. of Philosophy, Northwestern Univ.

Stahl G (1993) Supporting Interpretation in Design. *Journal of Architecture and Planning Research*.

Stahl G (forthcoming) Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design. Ph.D. Diss., Dept. of Computer Science, Univ. of Colorado.

Stahl G, McCall R, Peper G (1992) Extending Hypermedia with an Inference Language: an Alternative to Rule-Based Expert Systems. In *Proc. of the IBM ITL Conference: Expert Systems (October 19-21, 1992)*.

Suchman L (1987) *Plans and Situated Actions: the Problem of Human Machine Communication*. Cambridge: Cambridge University Press.

Winograd T, Flores F (1986) *Understanding Computers and Cognition: A New Foundation for Design*. New York: Addison-Wesley.

Supporting Situated Interpretation

Gerry Stahl
Department of Computer Science,
School of Environmental Design,
and Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0430
gerry@cs.colorado.edu
(303) 444-2792

Abstract

This paper discusses the role of interpretation in innovative design and proposes an approach to providing computer support for interpretation in design.

According to situated cognition theory, most of a designer's knowledge is normally tacit. Situated interpretation is the process of explicating something that is tacitly understood, within its larger context.

The centrality of interpretation to non-routine design is demonstrated by: a review of the design methodology of Alexander, Rittel, and Schön; a protocol analysis of a lunar habitat design session; and a summary of Heidegger's philosophy of interpretation. These show that the designer's articulation of tacit knowledge takes place on the basis of an understanding of the design situation, a focus from a particular perspective, and a shared language.

As knowledge is made explicit through the interpretive processes of design it can be captured for use in computer-based design support systems. A prototype software system is described for representing design situations, interpretive perspectives, and domain terminology to support interpretation by designers.

For publication in the Proceedings of the:
Fifteenth Annual Meeting of the
COGNITIVE SCIENCE SOCIETY
A Multidisciplinary Conference on Cognition
June 18-21, 1993
University of Colorado at Boulder