De*sign* — Indicating Through Signs

M. Cecilia C. Baranauskas and
Rodrigo Bonacin

Acknowledgements
This work was partially supported by grants from the Brazilian Research Council (CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico 476381/2004–05). We also thank Delphi Automotive Systems and Nied for their partnership, and the valuable contribution of the editors and the anonymous referees.

Introduction

Different disciplines have different concepts of "design," so our understanding of design varies according our particular field. The development of the design concept in the Human-Computer Interaction (HCI) field has inherited approaches, methodologies, and theories coming mainly from Information Systems (IS), Software Engineering (SE), Behavioral and Social Sciences and, more recently, from Design Studies. The rationalist tradition has dominated thinking regarding the design of interactive systems in the Information Systems and Software Engineering fields. As discussed by Ehn and Löwgren,[1] the first approaches to IS development can be characterized by a strong belief in systematic design methods founded in mathematical-logical theories. Research interests in accuracy and technical control guided these approaches. The main assumptions behind them, as suggested in some methods of SE, seem to be that the users (end-user, client, customer, stakeholder, or problem owner) are supposed to give complete and explicit descriptions of their demands in terms of the system to be developed.

Within the rationalist view of IS development, reality is objectively ascertained, is the same for everyone and is composed of entities, their properties, and relationships. Data is understood as a means of representing the truth about reality, and truth is the correct correspondence between some real entities. An information system is a kind of "plumbing" system through which data flow and, within this perspective, the role of the designer is to specify the truth data structure and functions of the system needed by users.[2] According to this view, interface design is just a matter of providing access to the underlying system functionality.

In the 1990s, this picture changed and one of the major sources of inspiration was the theoretical discussion on the actual nature of the phenomenon of designing computer artifacts. A reframing of the rationalistic understanding of computer systems started to consider reality as a social construction based on the behavior of its participating agents. Within this view, the role of the designer is to assist users to articulate their problems; discover their information requirements; and evolve a systemic solution. In other words, "design" is understood by Winograd and Flores,[3] and Adler and

1 Pelle Ehn and Jonas Löwgren, "Design for Quality-in-Use: Human-Computer Interaction Meets Information Systems Development" in *Handbook of Human-Computer Interaction,* Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu, eds. (Amsterdam, The Netherlands: Elsevier Science, second completely revised edition 1997), 299–314.

2 Kecheng Liu, *Semiotics in Information Systems Engineering* (Cambridge: Cambridge University Press, 2000).

3 Terry Winograd and Fernando F. Flores, *Understanding Computers and Cognition—A New Foundation for Design* (Reading, MA: Addison Wesley Publishing Company, 1986).

Winograd,[4] as: "... the interaction between understanding and creation." In designing a system, the focus is on concerned involvement rather than on correct descriptions.

In recent years, there also has been a growing interest in the HCI community to think about the development of usable systems as design work.[5] In this work, we will discuss the shift from a rationalistic perspective to the inclusion of interpretative, social, and communicative aspects in designing interactive systems. This position draws upon concepts from Organizational Semiotics (OS) to set up an appropriate foundation for understanding this view of "design" and for it to be reflected in the design of interactive artifacts.

Organizational Semiotics is a discipline that explores the use of signs and their effects on social practices. OS understands that each organized behavior is affected by the communication and interpretation of signs by people, individually or in groups. We base our work on Stamper's school of OS,[6] which proposes a set of methods to deal with information and information systems in a balanced way, taking into account both the technological issues and the human and social aspects of information resources, products, and functions. OS sees informatics from a social angle. We argue that OS provides artifacts that embody knowledge, and support collaboration and reflection among people from the different disciplines involved in interaction design.

This paper facilitates theoretical discussion as well as practical issues on interaction design, proposing a framework in which we have articulated ideas coming from semiotics to conduct work in interactive system design. A brief report on a case study related to the context of Computer-Supported Collaborative Work illustrates our proposal. The paper is organized as follows: we start by conceptualizing design and discussing paradigms that have been the background for interaction design. Then we situate the current understanding of interaction design, as compared to the main approaches of HCI. In the subsequent sections, we present a framework for interaction design inspired by Organizational Semiotics, and illustrate its application in a real design situation. Finally, we discuss the main contributions.

Design: Concept and Paradigms

The word originates from the Latin *designo* meaning to mark out, trace, plan; and also to point out, indicate, signify; to portray; or delineate.[7] In the context of the applied arts, engineering, and architecture; design is both a noun and a verb. Design as a verb is the process of originating and developing a plan for an aesthetic and functional object, which usually requires considerable research, thought, modeling, iterative adjustment, and redesign. The noun is used both for the final plan of action (a drawing, model, or other

4 Paul S. Adler and Terry Winograd, "The Usability Challenge" in *Usability: Turning Technologies into Tools,* Paul S. Adler and Terry A. Winograd, eds. (New York: Oxford University Press, 1992).

5 Phoebe Sengers, Kirsten Boehner, Shay David, and Joseph Jofish Kaye, "Reflective Design," *Proceedings of Critical Computing* (2005): 49–58; Daniel Fallman, "Design-Oriented Human-Computer Interaction," *Proceedings of the Conference on Human Factors in Computing Systems* (2003); and Pelle Ehn and Jonas Löwgren, "Design for Quality-in-Use: Human-Computer Interaction Meets Information Systems Development."

6 Ronald K. Stamper, K. Althaus, and James Backhouse, "MEASUR: Method for Eliciting, Analyzing, and Specifying User Requirements," *Computerized Assistance during the Information Systems Life Cycle,* T. William Olle, Alex A. Verrijn-Stuart, and Love Bhabuta, eds. (Amsterdam, The Netherlands: Elsevier Science, 1988); and Ronald Stamper, "Signs, Information, and Systems" in *Signs of Work Semiotics Information Processing in Organizations,* B. Holmqnist, et al., eds. (New York: Walter de Gruyter, 1996).

7 Latin Dictionary of the University of Notre Dame. Access at: http://archives.nd.edu/latgramm.htm.

description), or the result of following that plan of action (the object produced).[8] Design, ambiguously signifying both process and product, is a term that has been used to include not only the design of physical objects, but the entire range of artifacts made by human beings: from buildings to organizations, behavioral worlds, and theoretical constructs.[9]

Fallman[10] presents an attempt to conceptualize what the discourse of design *is* and what designers *do* when they design. The three accounts for design, which he names the "conservative account," the "romantic account," and the "pragmatic account," are useful to situate our understanding of designing interactive systems, and to frame our approach to *design*.

In the conservative account, "design is thought of as a scientific or engineering endeavor; borrowing methodology and terminology from natural sciences, mathematics, and systems theory, drawing on a philosophical base in rationalism."[11] A widely held model of the design process in the conservative account comes from Simon,[12] and Newell and Simon.[13] Simon has proposed a view of design as a problem-solving process he characterized as a "science of the artificial," distinguishing it from the natural sciences. In his model of designing, he proposes a division of the design process into generation and selection, considering generation either as a random combination of given elements, or as a systematic search of a problem space. This model of designing had profound impact not only on theories specific to the design professions, but on other theories built on the design metaphor, such as those in the fields of psychology, sociology, and economics, and very strongly in computer science.

Within the conservative account, a design methodology movement raised by the works of Alexander[14] and Jones[15] abstracts the design steps into two major processes: an analysis of the problem and a synthesis of a solution. During analysis the ill-defined and unstructured problem, domain is decomposed into more manageable constituents. In the synthesis stage, the designer constructs a solution to the overall design problem by solving each part of the problem constituents found during analysis.

The design of interactive systems, viewed from this account, is supposed to progress gradually from the abstract requirements specification to the resulting artifact: the computer system. The conservative account assumes that there is a problem to be solved, and that the descriptions of this problem can be comprehensively and accurately produced in the form of a requirements specification to be fed into a design process, which culminates with the designed interactive system.

Thus, methodology and structure are the heart of understanding and practicing design, and the designer's role is that of an engineer or a natural scientist.[16] In the conservative account, the designer is seen as a "glass box"[17] in the sense that every step of the process is suggested as rational and possible to describe.

8 www.en.wikipedia.org/wiki/Design
9 Donald Schön, "The Design Process," *Varieties of Thinking*, V. Adrian Howard, ed. (New York: Routledge, 1990).
10 Daniel Fallman, "Design-Oriented Human-Computer Interaction."
11 Ibid., 226.
12 Herbert A. Simon, *The Sciences of Artificial* (Cambridge, MA: MIT Press, 1971).
13 Allen Newell and Herbert A. Simon, *Human Problem Solving* (Englewood Cliffs, NJ: Prentice Hall, 1972).
14 Christopher Alexander, *Notes on the Synthesis of Form* (Cambridge, MA: Harvard University Press, 1964).
15 John Chris Jones, *Design Methods* (New York: Van Nostrand Reinhold, 1970).
16 Daniel Fallman, "Design-Oriented Human-Computer Interaction."
17 John Chris Jones, *Design Methods*.

Schön identifies sources of incompleteness inherent in the conservative model which cannot explain the initial creation of complex design structures; nor can account for the dialectical transformation of structures observed in the ways designers learn through designing. The issue of disagreement is the idea of a design structure: the designer's representation of a problem together with the rule-governed procedures that guide his transformation of it. For Schön, representation and procedures must be compatible. For the model of design as rational decisions, the design structure is assumed to be given with the presented problem. A rational decision process can occur only within such a structure. Hence, the model does not explain how design structures are made and remade in the course of designing. According to Schön, the model coming from the rationalistic tradition is limited either to the special class of artificially constructed problems—where design structure is given from the beginning—or to the later phases of designing—where it takes the form of technical problem-solving within a stabilized structure.

In actual designing, design proposals often are complex, interdependent on each other, and significant in their impact on design structures. This complexity, which Schön calls "figural," is in contrast to the combinatorial or merely additive. Addition or subtraction of one element changes the functional meaning of other elements, with the result that the proposal must be considered different *as a whole.* Examples of figural complexity are found in the drawings of the Gestalt psychologists, computer programs, and human organizations—where a change in one element (position, function, or feature) can produce significant changes in other elements and in the system as a whole. Complexity is closely linked to interdependence.

The "romantic account" of the design process suggests it is not a fully rational and explicable process; it has something "mystical." [18] This account of design can be thought of as "black-boxed" [19] in that the design process is guided by the designer's values and taste, and the product becomes judged according to issues of quality and aesthetics.[20] This view suggests that the arts present better models for design than science.

A design-oriented approach to HCI within this account emphasizes the designer´s individuality, aesthetics, and individual judgment over methodology and control, transparency, and logical reasoning. The product of design and the designer are accentuated, while the process of producing the artifact is opaque.[21]

The "pragmatic account" of design is characterized by its "situatedness": the design process is located in a world populated with people, artifacts, and practices, each with its own history and identity. Rather than science or art, design is understood as a hermeneutic process of interpretation and creation of meaning.[22] Designers iteratively interpret the effects of their designs on the situation at hand. It can be thought of as a reflective conversation with the materials of the design situation. In Schön's perspective:

18 Daniel Fallman, "Design-Oriented Human-Computer Interaction."

19 John Chris Jones, *Design Methods.*

20 Donald Schön, *Educating the Reflective Practitioner* (San Francisco: Jossey-Bass, 1987).

21 Daniel Fallman, "Design-Oriented Human-Computer Interaction."

22 Richard Coyne and S. Adrian, "Is Designing Mysterious? Challenging the Dual Knowledge Thesis," *Design Studies* 12:3 (1991): 124–31.

> Designing is seen as a conversation with the materials of a situation within which new trials are often based on learning from earlier ones. It is seen, for the most part, as a social process in which different designers frame the situation in different ways and learn, when they are successful, to talk across divergent frames.[23]

The pragmatic account focuses on the situatedness of the designer in a real-world situation, and brings to light the combination of roles, practices, and technologies involved in design. [24] The designer has constructive as well as reflective skills.

In the framing of our work, designing is a social process with focus on problem setting as well as on problem solving. It is a dialogue not only with design materials, but mainly among individuals (designers, developers, users, and other stakeholders) in which different views of designing and different ways of framing design situations are contrasted. Design dialogues are dialectical revelations of conflicts among views of design structure held by different parties in the dialogue. Design structures are made and remade during design dialogues. We regard designing mostly within the pragmatic account, as an iterative and interactive process of creating signs, which involves sense production and interpretation by people involved in the design.

Interaction Design and HCI Tradition

Many products that require users to interact with them have not necessarily been designed with the users in mind. Typically, they may have been engineered as systems to perform functions, within the conservative account to design. While they may work effectively from an engineering standpoint, it does not necessarily mean they will be easy, effective, and enjoyable to use from the user's perspective.[25] Because user interfaces are implemented with software, many software engineers believe that the well-established techniques for developing software in general will apply to user interface development. These techniques do apply to user interface software development, but not to designing what that software should implement; namely, the interaction with users.[26] Because of the "human factor," interaction design represents a domain with its own special problems, requiring its own special design techniques. Hartson[27] summarizes this understanding and the interdisciplinary nature of the HCI field as follows:

> Methodology, theory, and practice in the field of human-computer interaction (HCI) all share the goal of producing interactive software that can be used efficiently, effectively, safely, and with satisfaction. ...HCI is cross-disciplinary in its conduct and multidisciplinary in its roots, drawing on—synthesizing and adapting from—several other fields

23 Donald Schön, "The Design Process," 139.

24 Daniel Fallman, "Design-Oriented Human-Computer Interaction."

25 Jennifer J. Preece, J., Yvonne Rogers, and Helen Sharp, *Interaction Design: Beyond Human-Computer Interaction* (New York: John Wiley & Sons, 2002).

26 Deborah Hix and Rex H. Hartson, *Developing User Interfaces: Ensuring Usability through Product and Process* (New York: John Wiley & Sons, 1993).

27 Rex H. Hartson, "Human-Computer Interaction: Interdisciplinary Roots and Trends," *The Journal of Systems and Software* 43 (1998): 103–118.

> including human-factors, ergonomics, cognitive psychology, behavioral psychology, systems engineering and computer science.[28]

In line with this HCI view for producing interactive software, Preece, et al.[29] understand design as a practical and creative activity, whose aim is to develop a product that helps its users achieve their goals. Within this understanding, a goal of interaction design is to develop interactive systems that elicit positive responses from users, such as feeling at ease, being comfortable, and enjoying the experience of using them. Within this understanding, Preece et al.[30] conceptualize design by distinguishing two aspects: one conceptual and the other physical. The former is concerned with developing a conceptual model that captures what the product will do and how it will behave; while the latter is concerned with details of the design such as screen and menu structures, icons, and graphics. Design activities begin once a set of requirements has been established and the design emerges iteratively through repeated design-evaluation-redesign cycles involving users. For users to effectively evaluate the design of an interactive product, designers must produce an interactive version of their ideas. In the early stages of development, these interactive versions may be made of paper and cardboard while, as the design progresses and ideas become more detailed, they may be refined pieces of software or material that resembles the final product. The activity concerned with building this interactive version has been called "prototyping."

Therefore, there is a common understanding that developing a product must begin with constructing some understanding of what is required of it; although various approaches to designing may differ in their search for these requirements. User-centered design and participatory design (i.e., involving users) have been advocated as good practices for interaction design in HCI. User involvement in the design process seems to be generally accepted, although varying levels of participation may impact differently on the design product.

We encourage a broader understanding of the design process in which the software is understood as a medium for the creation of virtualities—the world in which a user of the software perceives, acts, and responds to experiences.[31] Moreover, we believe that technology design practices should support both designers and users in ongoing, critical reflection about technology, and its relationship to human life.[32]

As an alternative to the conservative account in HCI approaches, the design of computer applications that are concerned not only with the quality of the final products but primarily with the quality of system usage and the experience it enables, has been one of the main concerns of Scandinavian Participatory Design (PD). PD

28 Rex H. Hartson, "Human-Computer Interaction: Interdisciplinary Roots and Trends": 103.

29 Jennifer J. Preece, et al., *Interaction Design: Beyond Human-Computer.*

30 Ibid.

31 Terry Winograd, *Bringing Design to Software* (New York: Addison Wesley, 1996).

32 Phoebe Sengers, et al., "Reflective Design."

practitioners have long advocated active cooperation between users and designers, and a great amount of research has been conducted in establishing meaningful practices to provide a common ground for discussion among those directly in charge of technology design and use.[33] Participatory techniques are useful instruments to discuss the social context of the users through their active participation. Nevertheless, PD techniques seldom go beyond the early analysis/design activities of development projects.[34]

Taking the Scandinavian tradition as a starting point[35] proposed a more comprehensive development approach called Cooperative Experimental System Development (CESD). This extended cooperative and experimental techniques throughout the entire life cycle of a computer system, including technical design and implementation. Design was seen as the main concern in system development. The focus is on techniques to facilitate designers' and users' involvement in common creative activities. Experimentation with possible outcomes, based on hands-on experience with mock-ups and prototypes, is a central feature of CESD design. Object-oriented tools, as well as techniques to enable a smooth transformation of design artifacts to application code, also are a concern of CESD design.

Prototyping overcomes some of the problems of requirement specification-oriented methods, which usually assume that system design can be based solely on observation and detached reflection. Nevertheless, prototyping methods usually have a narrow focus and tend to limit discussion within the reality created by the prototype. Moreover, there is very little account of how prototypes are related to the current and future work practices of users. We argue that it is equally important that the people involved (designers, developers, users, and other interested parties) share a representation model of the work domain to be supported by the prospective system. Meaning-making is constructed as a result of cooperation between designers, developers, interested parties, and prospective users of the technology being designed. In the context of our design framework, we argue that Organizational Semiotics provides artifacts which serve the participating disciplines as a means for the people involved in the problem design to express and share their knowledge of the world around them.

A Framework for Interaction Design Inspired by Organizational Semiotics

The conservative approaches to interactive software system design present a strict separation between design, implementation, and the use of computational systems. These approaches assume a preexisting common conceptual model of the domain and their agents that is shared by all practitioners. In this way, the problem is reduced to capturing this model and codifying a solution based on the model.

33 Douglas Schüler and Aki Namioka, *Participatory Design: Principles and Practices* (Florence, KY: Taylor and Frances Group, 1993).

34 Kaj Grønbæk, Morten Kyng, and Preben Mogensen, "Toward a Cooperative Experimental System Development Approach" in *Computers and Design in Context,* Morten Kyng and L. Mathiassen, eds. (Cambridge, MA: MIT Press, 1997), 201–238.

35 Grønbæk, et al., "Toward a Cooperative Experimental System Development Approach."

On the other hand, several authors acknowledge the fact that domain models do not in fact exist as an *a priori* object, but instead are socially and dynamically constructed.[36]

In doing design, we rely on various cues, indicators, and conventional signs. Semiotics, the ancient doctrine of signs, leads us to a more precise understanding of information as various properties of signs. Signs are simple entities easy to deal with within the intersubjective domain. Anything standing for another thing or used to signify something else is an example of a sign: words, sentences, traffic lights, diagrams, a wave of hand, or a facial expression. A language community can cross the bridge between signs and reality (what people are observed doing). Within this understanding, system developers and users coevolve, with the language as the mediator of meaning.[37]

In this work, we take semiotics beyond the study of how we use signs for communication to include the shared knowledge and mutual commitment derived from communication in designing. We share with the pragmatic account to design, the understanding that design is about being engaged directly in a specific design situation. This "situatedness" locates the design process in a nested structure in which the informal, the formal, and the technical layers of information and interaction coexist. The "informal layer" represents the informal interactions in a society, the culture in which meanings are established, intentions understood, beliefs formed, and commitments made, altered, and discharged. In the formal layer, forms and rules of an organized society represent meaning and intention (e.g., laws, formal methods of work organizations, models, etc). The inner layer represents the technical interactive system, derived from part of the formal layer which, in turn, draws on the informal layer. Figure 1 is based on the "organizational onion" from OS, and illustrates our proposed account for the design of interactive systems as *indicating through signs*.

A problem setting is part of the design situation understanding and requires articulation in forms that can be appropriated and assessed by people involved in designing (designers, users, developers, and other stakeholders). The design process involves exploring the reality that constitutes the design situation. An ontology is a crucial aspect of what the involved group understands as constituting reality. The ontology charting allows a discussion of meaning and on what the group considers to be important aspects of reality in a particular design situation. System prototyping refer to the group's idea on how to shape their intervention in the situation, based on their ontology and problem articulation. The work in these three layers is performed in parallel, and coevolves: a problem understanding is revealed as the group works on the semantics and solution ideas.

36 Jonas Löwgren and Erik Stolterman, *Thoughtful Interaction Design: A Design Perspective on Information Technology* (Cambridge, MA: MIT Press, 2004); Gerhard Fischer, Stefanie N. Lindstaedt, Jonathan L. Ostwald, Markus Stolze, Tamara Sumner, and Beatrix Zimmermann, "From Domain Modeling to Collaborative Domain Construction," *Proceedings of the Conference on Designing Interactive Systems* (1995): 75–85; and Morten Kyng, "Designing for Cooperation: Cooperating in Design," *Communications of ACM* 12:34 (1991): 65–73.

37 John Rheinfrank and Shelley Evenson, "Design Languages" in *Bringing Design to Software,* Terry Winograd, ed. (New York: Addison Wesley, 1996).

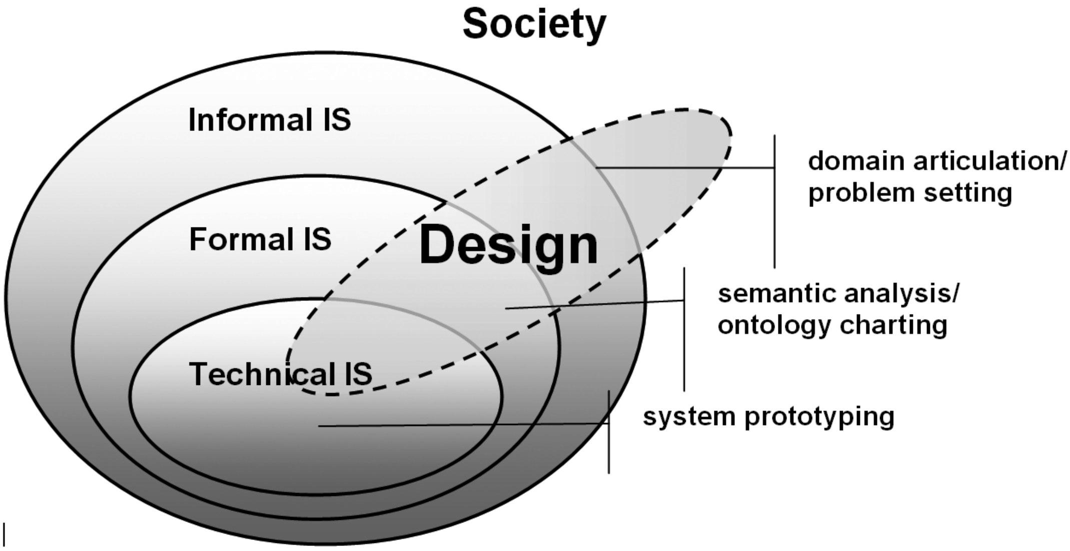


Figure 1
The structure of design as indicated through signs from different layers.

We acknowledge the stages of analysis, synthesis, and evaluation in the design of interactive systems, although not in a linear order. Understanding and describing the problem, finding a solution, and implementing it do not occur straightforwardly as suggested by the conservative account.

Thus, design is conceived as a social process of expressing meaning, communicating intentions, and constructing knowledge, to be carried iteratively and interactively by designers and a group of stakeholders in a participatory style. It is reflective as well as constructive in nature. Several design artifacts, located in the different layers of this structure (from a brainstorming activity or a low-tech artifact situated in the informal layer, to the design models used in the formal layer, to the high-fidelity prototypes) coexist. They aim to encourage and maintain the interaction among users and designers in a social process in which the different views of the design are contrasted and negotiated.

Organizational Semiotics

Organizational semiotics presents theories and methods developed in the course of a research program initiated in the 1960s to allow one to analyze and design information systems in terms of three human information functions: expressing meanings, communicating intentions, and creating knowledge.[38] Studies in OS are not restricted to information expressed in written or graphical discourse, but take into account the semiotic aspects of human interaction in the organization. In the philosophical stance underlying OS, reality is seen as a social construction based on the behavior of agents participating

38 Ronald K. Stamper, "Organizational Semiotics: Informatics without the Computer?" in *Information, Organization and Technology: Studies in Organizational Semiotics*, Kecheng Liu, Rodney J. Clarke, Peter Bogh Andersen, and Ronald K. Stamper, eds. (Norwell, MA: Kluwer Academic Publishers, 2001).

in it: people share a pattern of behavior governed by a system of signs. Since people are constantly communicating and discussing, the world is in constant change.

Semantic analysis is one of the OS methods that focuses on the agents and their pattern of behaviors to describe an organization taken in its broadest sense, including its interactive systems. With the analyst in the role of facilitator, an ontology chart is constructed describing a view of responsible agents in the focal system domain, and their behaviors or action patterns. Some basic concepts of OS adopted in this work are based on Liu[39] and Stamper:[40]

- "The world" is a social construction based on the actions of agents, and on the basis of what is offered by the physical and social worlds: invariant repertoires of behavior constitute the perceivable reality.
- "An agent" is defined as something that has responsible behavior. An agent can be an individual person, a cultural group, a language community, or a society (an employee, a department, an organization, etc).
- "Affordance," the concept originally introduced by Gibson[41] to express the behavior of an organism made available by some combined structure of the organism and its environment, is extended by Stamper[42] to include invariants of the social world: social affordances arise from the norms we share with people around us. Those repertoires of behavior are the ones that make us human rather than animal: "The rich array of affordances available to us we acquire through our engagement in a society able to hand down, through the generations, the useful behavior and perceptions that its members have discovered."[43] Stamper argues that reality, as we know it, was not constructed individually: it was created by cultural development during millenniums. For example, a cup is a human artifact whose use is not only possible because of its physical aspect, but also because of its social affordances (children have learned to use it for drinking, instead of throwing it at someone).
- "An ontological dependency" is formed when an affordance is possible only if certain other affordances are available. An ontological dependency between "A" and "B" means that "A" is only possible when "B" also is possible. The ontological relationship is considered as the most fundamental relationship to model.

The concepts of semantic analysis are represented by means of ontology charts, which have a graphical notation to represent agents (circles), affordances (rectangles), ontological dependencies (lines drawn from left to right), role-names (parentheses), and whole-

39 Kecheng Liu and Alan Dix, "Norm Governed Agents in CSCW," *Proceedings of First International Workshop on Computational Semiotics* (1997).

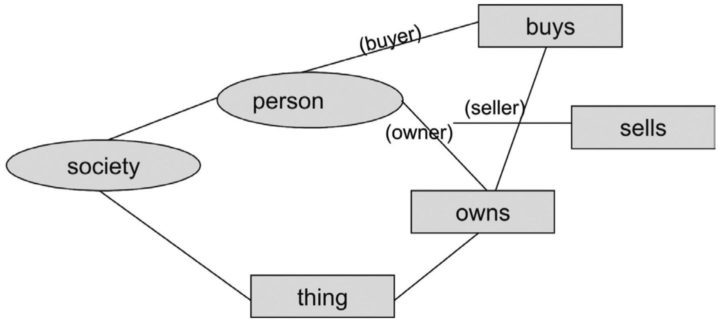
40 Ronald K. Stamper, "Organizational Semiotics: Informatics without the Computer?"

41 James J. Gibson, *The Ecological Approach to Visual Perception* (New York: Houghton Mifflin, 1979).

42 Ronald K. Stamper, "Signs, Information and Systems."

43 Ronald K. Stamper, "Organizational Semiotics: Informatics without the Computer?" 140.

Figure 2
An illustration of an ontology chart.[42]



part relations (dot). The hypothetical example in Figure 2, extracted from Liu,[44] illustrates a fragment of a semantic model represented in an ontology chart. The "society" is the root agent in this model. "Person" and "thing" are both ontologically dependent on "society," which means both and all the other affordances are defined in the context of a certain society. The action's "sells" are ontologically dependent on "owner" (the role of a person who "owns" a "thing"), and the action's "buys" are built upon "person" and "owns." This suggests that, in that particular society, selling is only possible for the person who owns the thing. Selling and buying are referred to the affordance "owns." That means when people are trading, it is the ownership rather than the physical thing itself that is dealt with. In this sense, the representation ontologically reflects the social practice which is dominated by the shared norms in that particular social context.

The meaning of words used in the semantic model is treated as a relationship between the signs and appropriate actions of the agents. We understand the diagram itself as a group of signs. Therefore, the ontology chart is something that is socially constructed in an iterative and interactive process by people involved in designing the organization as well as the interactive system. The design situation is discussed cooperatively in several iterations according to the raised affordances and ontological dependencies: the diagram is not only the object of discussion, but a result of the discussion as well.

In addition to the "semantic analysis," which focuses on the agents' patterns of behavior, "norm analysis" is used to describe the relationships between an intentional use of signs and the resulting behavior of responsible agents in a social context. Considering the example of Figure 2, in a particular society, ownership as well as trading are governed by a set of norms created by the action of agents in that society. At the social level, norms describe beliefs, expectations, commitments, contract, law, and culture, as well as business.[45] Norms can be represented by the use of natural language or "deontic logic"[46] in the late stages of modeling. The norm model

44 Kecheng Liu, *Semiotics in Information Systems Engineering*, 70.
45 Ibid.
46 Kecheng Liu and Alan Dix, "Norm Governed Agents in CSCW."

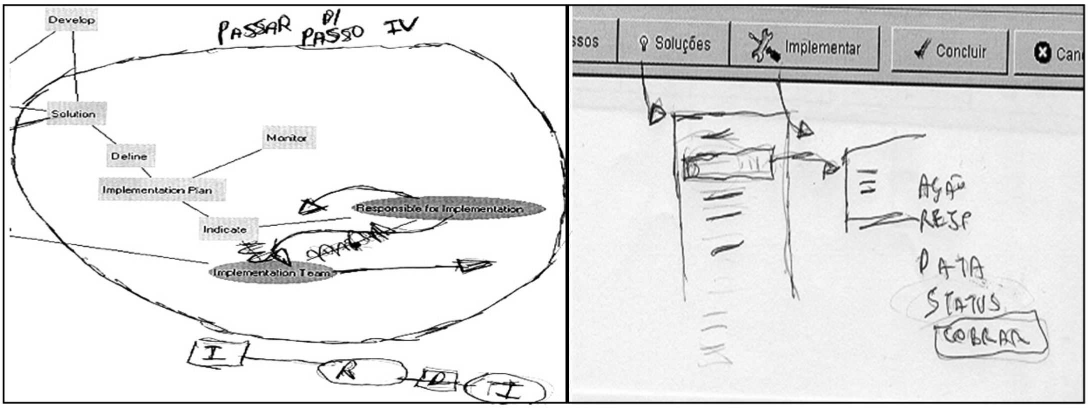


Figure 3
Example of proposed changes during the Pokayoke design.

itself can be understood as a group of signs. In the context of this paper, the norms are modeled as a result of reflection about the organizational context made by the agents in cooperation during the iterations of the design cycle.

Interaction Design in the Proposed Account: Short Report on a Case Study

Pokayoke is a computational system constructed with the aim of exploring the proposed approach to design in practice. The system was designed to support problem solving and decision making in the context of a manufacturing organization that adopts the lean production paradigm. This organization is a unit of a multinational company in Jaguariúna, Brazil which produces automotive parts. Pokayoke is based on one of the factory's procedures to analyze and implement corrective, preventive, security, and health actions, known as "the five steps." The five-step procedure provides a systematic method for dealing with problems in the production routine. Every time an unconformity is identified, an action must be taken to correct it and to prevent its reoccurrence. Also, every time a situation of potential unconformity is indicated, an error-proofing action should be carried out according to the *Poka Yoke*[47] concept of lean production.

The Pokayoke interactive system was developed in fourteen months and distributed in five prototype cycles, with the participation of a diversity of users ranging from shop floor workers to managers. Some participatory techniques (e.g., Starting Conference, Artifact Walkthrough,[48] etc.) were applied in the early iterations, in addition to the Semiotic Conference.[49]

Figure 3 shows a fragment of the combined use of an ontology chart and prototype in discussions during the Pokayoke design. The first versions of the prototype and ontology chart were constructed based on the results of PD techniques in early stages of interaction. The objective of using the ontology chart in this process

47 *Pokayoke* is a hybrid word created by Japanese manufacturing engineer Shigeo Shingo. The word comes from the words *yokeru* (to avoid) and *poka* (inadvertent errors).

48 A summary of an extensive set of PD techniques can be found in Michael J. Müller, J. H. Haslwanter, and John T. Dayton, "Participatory Practices in the Software Lifecycle" in *Handbook of Human-Computer Interaction,* Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu, eds. (Amsterdam, The Netherlands: Elsevier Science, second completely revised, 1997), 255–297.

49 Rodrigo Bonacin and M. Cecilia C. Baranauskas, "Semiotic Conference: Work Signs and Participatory Design," *Proceedings on 10th International Conference on Human-Computer Interaction 1* (2003), 38–42.

is not to construct a precise and formal representation of reality. It is used as a tool through which the practitioners can express their understanding and review their work practices while engaged in a signification process. It can be combined with other representation artifacts such as low-fidelity (and high-fidelity) prototypes.

The iterative approach is aligned with the idea that domain models do not exist *a priori*. The semiotic model and the prototypes of the computational artifacts are continuously (re)designed by designers, developers, and practitioners in a process that combines interaction and iteration.

The use of an ontology chart combined with a low-fidelity prototype in Figure 3 shows the need to expand part of the system (represented in the high-fidelity part of the figure). The main focus is not the models or charts themselves, but the discussion about the concepts behind these artifacts. In the Pokayoke case study, although the workers were not able to build semantic diagrams in the first session, they were able to discuss the modeled concepts and rethink their work practices. As discussion takes place, changes and suggestions are reflected in the models and prototypes. A quick example of the discussion that transpired, corresponding to Figure 3, regarding the workplace is illustrated as follows:

- *"The 4th step should be finished only after conclusion or cancellation of the actions."* Some workers used to finish "Step IV" before the conclusion of some actions. A mechanism to avoid that, which is considered a bad practice by the practitioners, was proposed. Figure 3 shows part of the prototype constructed during the meeting. Motivated by discussions about concepts represented in the ontology chart, they proposed to have the *status* for the problem, and the *control* of the due actions (the Portuguese word *cobrar* in Figure 3) represented in the user interface, and;

- *"The person in charge of the five-step procedure is not responsible for the actions of correction."* The "responsible for actions" concept was clarified through the ontology chart. New practices were adopted in the factory regarding this fact, even before the use of the system. As a result, they proposed the inclusion of a field in the prototype interface that identifies the role of the person in charge of the action (Figure 3).

Figure 4 shows part of the new versions of the ontology chart and prototype that resulted from this particular discussion. The diagram represents the relationship between "Brainstorm," "Solution Ideas," and "Actions" discussed during the previous iteration. The new prototype was implemented based on the suggestions, discussions,

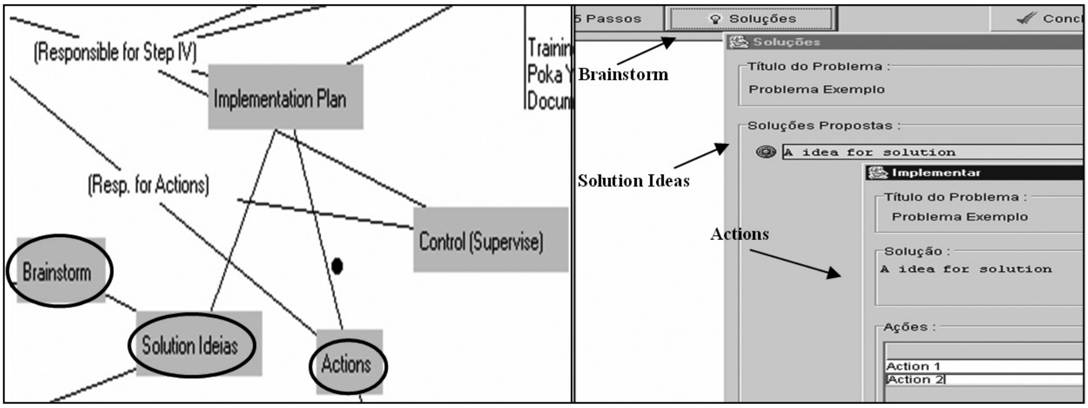


Figure 4
The modified ontology chart and user interface.

and alternatives explored during the previous interaction. The solutions proposed in the "handwrite prototype" were reflected in the new system interface.

After five iterations, Pokayoke substituted the paper-based form of the "five-step" process in the factory. The use of the system in the production line was investigated over the course of one year. The workers' process of making sense of the design elements, both in the abstract level as well as in concrete terms, allied with the feeling of authorship, was fundamental for system acceptance. The workers have expressed these feelings many times during system design and use, saying: "... we have defined this in this way to avoid ...."

The main drawback of the proposed approach encountered during the Pokayoke design was the reading of the ontology chart in the first iterations. This problem was minimized as we focused on the concepts (agents, their patterns of behavior, and ontological dependencies) instead of the notation. After some meetings, the practitioners were able to read the notation and use it to express themselves. The discussion of social norms, for example, resulted in new practices that may have greater value to the organization as a whole than to the computational system itself.

Discussion

> There is no direct path between the designer´s intention and the outcome. As you work a problem, you are continually in the process of developing a path into it, forming new appreciations and understandings as you make new moves.[50]

We share with Schön and Bennett[51] the idea of design having a figural complexity, and demanding a dynamic process of construction, rather than a one-shot approach to it. Moreover, we acknowledge the situatedness character of design, in which the designer is immersed in a world populated with people, artifacts, and practices. Thus, we regard the "designer" not only as the person in charge of

50 Donald Schön and John Bennett, "Reflective Conversation with Materials" in *Bringing Design to Software*, Terry Winograd, ed. (New York: Addison Wesley, 1996), 171.

51 Donald Schön and John Bennett, "Reflective Conversation with Materials."

the process and product of design, but as a group of people (designers, developers, users, and other interested parties) involved in an iterative and interactive process. The design activities are conducted by this group of people in a participatory style.

We share with the Scandinavian design community the understanding that design should be done *with* the users (neither *for* them nor *by* them), and that mutual learning is part of the work of a design group. In the participatory design tradition, prototypes and mock-ups are proposed to allow users to be active in the design process. Nevertheless, prototyping taken in isolation tends to limit discussion within the reality created by the prototype. Our concept of interaction design views it from a social angle, acknowledging from OS the understanding that we construct our social world as layers of affordances that depend on each other for their existence. The semantic analysis enables the group to draw attention to the agents and their pattern of behavior expressing meaning in the ontology charting. In the proposed approach to design, prototypes and ontology charts are artifacts that coevolve, informing each other.

In summary, in the proposed approach, the design process is a social construction of designers, users, and other stakeholders actively engaged in the problem setting as well as in the problem solution. Several artifacts (informal, formal, and technical) are used by the participants during this process as communication and mediation tools in designing the interactive system. The ontology charts provide us with a way to represent the concepts discussed in the design domain. The nodes do not represent concepts in someone's mind but, rather, socially shared, physical, or social affordances (invariant repertoires of behavior). The product of design emerges through several iterations of this process in which analysis, synthesis, and implementation activities are intertwined.

The patterns of agents' behavior in problem setting (social affordances) represented in the ontology chart, reflect the participants' knowledge about the problem domain, exemplifying Schön's: "We could say that our knowing is in our action."[52] During an iteration of the design process, designers and stakeholders are reflecting not only on the phenomena they are representing, and making sense of it through their drawing, but also on their previous understandings of the design problem.

In the context of the proposed approach, conversation does not denote a literal verbal dialog. Rather, it refers to an interactive communication among the participants taking place through changes in the semantic model and prototype drawings, which serve as a representation of shared knowledge. The ontology charts represent their "language of communication" in the sense proposed by Rheinfrank and Evenson,[53] since they have a communicative function as well as the structure of an evolving system of elements and of relationships among those elements.

52 Donald Schön, "The Design Process," 173.

53 John Rheinfrank and Shelley Evenson, "Design Languages."

The ontology chart enables the team to give visible form to the assumptions and the design concepts. At the same time that the language elements are being developed, they are being materialized in the prototype. This simultaneous demonstration of the elements as they are being conceived is crucial because certain concepts may be difficult to imagine without tangible examples. Another purpose for iterative demonstration of the language is to help organizations make development assumptions explicit, and to enable meaning negotiation. In the context of the proposed design framework, meaning is seen not just as the built-in sense of an object, but also as the quality of sense making that objects have and can produce, especially with respect to their social surroundings. The interactive co-construction of the design language is facilitated by the concreteness of the prototypes.

Design as *indicating through signs* allows the participants in designing (designers, users, and stakeholders) to share control of, and responsibility for, the meaning-making process. This requires active participation for co-construction of meaning. This can be accomplished by expressing/communicating signs through the ontology charts and other artifacts, and materializing knowledge in concrete terms through the prototypes.

Conclusion

Methodologies for interactive systems design and development traditionally have drawn upon the conservative paradigm, which considers an objective reality to be discovered, modeled, and represented in the software. If we understand design as communication, and software as a medium for the creation of virtualities, other human communication disciplines can give system engineers a new way to think about interaction design. The designer could be providing the user with tools to create meaning and experience, rather than creating meaning and experience for the user.

Organizational semiotics understands reality as a social construct based on the behavior of its participating agents. OS provides artifacts to represent what we know and share about the world around us. In this paper, we have shown a semiotic-inspired framework that illustrates our understanding of interaction design as communication through signs. This framework has proven its usefulness during the interaction design of Pokayoke, a computer-supported collaborative work system designed for the context of a manufacturer. The approach did not search for an objective truth about the best way to support practice in the factory. Rather, this truth was socially constructed based on meaning negotiations that occurred during the system design. The semiotic models and the use of the prototype screen shots in the design activities have been essential in exploring the connections between the meanings of the design context and the interactions designed to support them in the interactive software system.