Value-Driven Design
with Algebraic Semiotics

Joseph Goguen and Fox Harrell

1 Introduction

The goal of this book is to present a practical engineering approach to design that has a rigorous basis, supports calculation, and is driven by what stakeholders (i.e., those who have a "stake") really want, as expressed in their value systems. These two aspects may seem quite different, in a sense expressed in the 1950s by C.P. Snow [9] as belonging to the "two cultures," of the sciences and the humanities. However, we contend they are highly compatible, and that both are essential for good design. The two are unified in our systematic new vision for *semiotics*, the theory of signs, or better, of signification or significance. Our approach, called *algebraic semiotics*, has a rigorous algebraic foundation that supports precise computations for critical decisions, and that incorporates values in practical methods, which are based on recent sociological research, for determining and using the values of stakeholders. Values, structure and representation are implicitly involved in every design decision, and the major thesis of this book is that design can be improved in quality and efficiency if these factors are made more explicit at appropriate points, and if precise principles for their use guide design decisions, especially tradeoffs.

The practical design principles in this book are general enough to apply to the design of websites, software systems, graphic displays, video games, art objects, magazine layouts, and more. But our illustrations and exercises mainly focus on the design of websites, due to their concreteness, importance, and relative simplicity, although examples are also drawn from from other areas, such as information visualization and multimedia art. Our design principles generally draw on the deep intentions, i.e., the value systems, of stakeholders, to achieve overall consistency and coherence. However, some of the most interesting principles concern optimization when tradeoffs among competing criteria are necessary. Experienced designers will certainly recognize many special cases of our principles, but as far as we know, they are new in the very precise and general formulations given here. We find it both practical and exciting to combine engineering design with sociological methods that take account of the values of stakeholders. Throughout, we seek to use the most rigorous and scientific foundations that are possible.

This book is not primarily concerned with low level programming issues; although these are certainly a necessary part of computer-based system development projects, they are far from sufficient for good design. It is all too common to see systems that fully meet their technical requirements for functionality, efficiency, evolvability, and so on, but that are very difficult, unpleasant, or even impossible to use. This book focuses on designing systems that provide what stakeholders need. Technology is evolving very quickly, but professionals usually have relatively little difficulty learning what they need for any particular purpose, whereas practical help in addressing the challenge of designing systems that users really like is much harder to obtain, as well as (we believe) much more valuable and appropriate in today's professional environment. Nevertheless, Appendix **??** includes some basics of technologies such as HTML, XSL, and CSS that are used in this book.

The remainder of this introductory chapter gives a brief overview of the content of the book. The details of course can be found in the rest of the book, including algebraic semantics, classic and algebraic semiotics, sociology of values, practical value discovery methods, representation theory, including optimality principles, and numerous examples, including some case studies; sidebars expand on certain important topics. Those (perhaps few) who would like to see more theory may consult Appendix **??**. Each chapter has final sections for notes and exercises. The notes provide references, history, and other background information that may be of interest. The exercise sections may also include project suggestions.

1.1 Evolution of Interface Research and Design Methods

The earliest systematic computer system interface research was based on *ergonomics*, a subject concerned with optimizing the usability of tools, machines (e.g., in assembly lines), applicances, furniture, and other physical artifacts. Ergonomic methods work well for keyboards, mice, menu layouts, and the like, but fail to address the major difficulties that arise in helping users accomplish tasks like writing a report, buying a book, or deleting junk email, because such tasks involve cognitive and social issues in a significant way.

The difficulty of applying physical ergonomics to essentially cognitive tasks motivated the later use of *experimental psychology*, based on precise measures of performance, such as task completion time, time between keystrokes, number of keystrokes, and number of errors. These measures are similar to those used in ergonomics, but are now applied in controlled experiments that involve cognitive tasks. Such experiments can be used to compare two designs, but they are expensive, and have failed to deliver general theories that can guide design. As a result, there is an ongoing move towards less rigorous methods, such as case studies, interviews, questionnaires, and stakeholder panels; an excellent survey is given in [8]. In particular, a large community of usability specialists successfully employ informal methods to uncover problems in existing designs. Although new ideas from cognitive science and sociology are gaining ground, vague but useful guidelines and rules remain predominant among professional designers. We argue that deep reasons for this include our current poor understanding of the mind, the mind/brain/body complex, language, and social interaction in general.

The rise of the internet has led to an increasing interest in social aspects of design, because many important applications now have a significant component of communication, and are better viewed as supporting the activities of a group than as supporting the activities of separate individuals. Examples include instant messaging, chat rooms, online forums, distributed games, online auctions, open source software development, wikis, and much more, such as the collective book review process used by Amazon, and the collective link rating system used by Google. The burgeoning field of computer supported cooperative work (CSCW) addresses such issues from a socio-technical point of view, and there has also been significant interest from professional sociologists. But the real world is developing faster than the academic world can create names and organizations, and CSCW has now outgrown its traditional focus on work.

Each of the three broad classes of methods considered above can be valuable in practice, if applied within its proper scope. Physical ergonomics and classical experimental psychology work well for the design and analysis of input/output devices, sensory-motor tasks, and simple displays. But they do not scale up to complex devices (such as aircraft cockpits), complex tasks (such as landing an aircraft), and complex displays (such as the many layers of menus and windows of

Paradigm	Computing	Interface Technology	Interface Research
Social	Local & Area Networks Internet, Email, Actors, Distributed Games, Web, Blogs, Wikis, etc.	Ubiquitous, Pervasive, Ambient, Surveilance, Multi-User, Environnments Spyware, Spam	CSCW Sociological Methods
Individual / Cognitive	Higher Order Languages Interaction Personal Computing	Time Sharing Graphic Interfaces	Ergonomics Cognitive Psychology
Hardware Centered	Machine Language Assembly Language	Direct Machine Access Batch Oriented	Ergonomics

Figure 1: Parallel Evolution of Computer-based Systems, Interface Technology, and Interface Research

a modern operating system interface). Informal methods can be effective for overall evaluation of such interfaces, and for uncovering some design flaws, but are not well suited to developing general design principles, or to discovering difficulties that that involve complex cognition, or social interaction. Unfortunately, methods based on experimental psychology, and on sociology, tend to be labor intensive, and although they can uncover interesting information, it can also be difficult to relate that information to specific design problems. This book attempts to fill the gap, by providing an efficient design method that is thoroughly sensitive to social issues throughout the whole process.

The three rows in Figure 1 are a sequence of relatively mild paradigm shifts (in roughly the sense of Thomas Kuhn [5]), each step upward motivated by weaknesses in the prior paradigm. This sequence neatly reflects the historical development of the three columns, as well as the difficulty of the problems being addressed. For example, ergonomic methods are the earliest and easiest research method[1], while social issues pesent the most recent and complex problems.

In summary, it seems fair to say that there is a *crisis* in interface design: we are called on to design systems of ever broader scope and greater complexity, but we lack precise, systematic methods that can help us do so efficiently and effectively. Simply, we do not know how to design such systems, and we often fail. This book aims to improve that situation, without claiming to have an ultimate solution, but rather, to have taken useful intitial steps down a promising path. We do claim that using the methods presented in this book will result in more efficient and effective design practice than is currently the norm, and we back this claim with a number of case studies of real deployed systems.

1.2 Value-Driven Design

A fundamental thesis of this book is that values can be used throughout the process of design and development to drive that process (we view design and development as inseparable), not just to evaluate designs, or to help determine requirements, but also to drive major and minor design

[1] Recently, ergonomics has tried to evolve into "cognitive ergonomics," which is why it appears twice in the figure.

choices throughout the process. This can not only improve the system, but can also improve the process, by clarifying difficult tradeoffs, and simplifying the way of choosing among alternatives; we will see many examples of this, ranging from low level issues like the ordering of menu items, up to fundamental choices about what to build, or even whether to deploy a built system.

Values are everywhere: all human choices involve values. These range from simple preferencess (such as not liking yellow), up to the kind of large grain issues grouped under names like "family values." It is useful to distinguish between values and ethics. *Ethics* refers to values that are generally proclaimed and promoted in some particular society; these need not be the same as the values that are generally practiced by members of that society, and of course, some individuals may have values that differ greatly from those of their society. Ethical values are usually general principles (such as "thou shalt not kill"), whereas values in the sense that is more useful for design can also be quite specific (such as, "it is more important for this particular screen, to know the date of a transaction than its exact time"). We will see that values having to do with task orientation and social interaction are the most important for design.

We consider that not just users, managers, etc., but also designers and engineers are stakeholders in the systems that they design, build, test and deploy. This is important because it empowers designers to consider and assert (but not to impose) their own values, which may include ethical issues that in some cases require them to refuse to work on a project, or even to resign. Values do not exist in the abstract, but can only be realized in real individuals in actual concrete situations.

Some of our earlier writing used the phrase "value-centered design," but we now think that "value-driven design" better reflects our approach of using values throughout the entire design and development process. The work we know that is most closely related to ours is the "value sensitive design" of Batya Friedman, Peter Kahn, Alan Borning, and others [1]; this approach differs from ours in focusing more on large grain human values, such as privacy, accountability, informed consent, and democracy, and less on small grain values (preferences), such as the relative importance of parts of a website, a webpage, or a menu; our approach is also more mathematical and more semantic.

1.3 Algebraic Semiotics

Semiotics is the study of signs. The American logician Charles Sanders Peirce (pronounced "purse") introduced this term and many of its basic ideas [6]. He emphasized that meanings are not directly attached to tokens; instead, signs are triadic (i.e., 3-part), involving a token, a meaning, and a process of interpretation that includes the context in which interpretation is done. Ferdinand de Saussure also contributed the important ideas [7] that signs are structured, and should be studied as members of systems of signs, rather than as individuals. We consider signs in a very broad sense, as any mediators of meaning: they are not just simple "tokens" or physical marks, but can be complex combinations of lower level signs; e.g., sentences, spoken or written, newspaper advertisements, books, even furniture, buildings, and operating systems. We also consider meaning in a very broad sense, as including the uses and values associated with signs.

We extend classical semiotics to *algebraic semiotics* to model structure and representation of signs, as well as dynamic interaction and values; algebraic semiotics is the basic language of this book, justified because designs consist of structured signs that represent other signs, in a way that reflects values. There are two main concepts in algebraic semiotics. The first is *semiotic space*, also called *semiotic theory*, which is a model for a family of dynamic structured signs with values. The second is *semiotic morphism*, which is a model for a representation of one semiotic space into of the signs of another semiotic space (which might for example represent the family of possible signs for some output device). The formal basis for algebraic semiotics is *algebraic semantics*, a rigorous mathematical theory of data representation based on equational logic; however, we introduce and use it only in a very informal way. Those who wish to see a more formal exposition may consult Appendix **??**.

1.4 Cognitive Science

Cognitive science studies the human mind, including perception and cognition. While not ignoring the brain, that is not its focus. On the other hand, it has become increasingly clear that the mind should not be studied independently of action, the body, and the environment. It seems obvious that good design should take account of the actual cognitive capabilities and limitations of human beings. Cognitive science topics of particular interest for this book include cognitive linguistic approaches to concepts, metaphors, and related developments, such as blending theory; these are discussed in Chapter **??**. Significant precursors of these modern developments can be found in gestalt psychology, especially in relation to perception; for example, the notion of "affordance" from the ecological psychology of James Gibson [2] is related. We also explore the limitations of focusing on individual cognition, and develop alternatives for situations where social interaction is essential.

1.5 Empirical Methods for Social Issues

Few computer scientists are aware of the vast amounts of empirical work in the social sciences that can help them deal with the social issues that so often plague their projects. Some of these, though empirically based, are mainly useful in providing a perspective on the nature of socio-technical systems, but others can be applied directly by computer scientists to understand particular situations and discover the values of participants. Among the former is cultural historical activity theory in the tradition of Lev Vygotsky [10], as well as the distributed cognition of Ed Hutchins [3], and the actor network theory of Latour, Callon and others [4]. Among the many useful concepts that come out of these different schools, that of *mediation* is of particular importance for interface design. The main method for discovering values in this book is based on William Labov's research on narrative; it provides a way to uncover the values implicit in the stories that members of social groups tell one another.

1.6 Notes

The subsections below discuss the history and use of this book, some early sources of its approach, a note for instructors, and acknowledgements to those who helped.

1.6.1 Origin and Use of this Book

This book grew out of the first author's several years experience teaching undergraduate and graduate courses on user interface design at UCSD. This experience ensures that the material can be understood by ordinary students, and can be applied by them in practical projects. The second author has been involved in the course as a student, teaching assistant, and author or co-author of research papers that develop and use its content. Much of the content of the book comes from papers and course notes by one or both of these authors.

The book is intended for use in both undergraduate and graduate courses; material that is more suitable for the later is marked with a star, and a few items marked with a double star are intended to challenge even graduate students.

1.6.2 Historical Notes

For us, the three great precursors of our approach to design are Charles Sanders Peirce, Ferdinand de Saussure, and Douglas Engelbart. Pierce (1839–1914) was a Boston philosopher and logician, whose early research laid the foundations for much of what became modern first order logic and modal logic; he regarded semiotics as a necessary foundation for this work. He also founded pragmatism, a philosophy of practical action in the world, which has been influential in Anglo-American philosophy, and which is very close to the worldview of engineering. Ferdinand de Saussure (1857–1913) was a Swiss linguist who taught in Geneva. Whereas Peirce focused more on the philosophy of meaning, Saussure was empirically motivated, and noticed in particular that signs always come in families that are structured in particular ways (e.g., verb tense system of a given language). An important contrast between their approaches is that Saussure's signs have a direct dyadic (2-ary) connection beteen token and meaning, whereas Peirce's signs have a more subtle triadic (3-ary) relation that explicitly includes interpretation. Peirce also studied metaphor, and developed a detailed classification of the different kinds of sign. Saussure had a great influence of the French structuralist and post-structuralist movements in anthropology, literature, popular culture, and other areas. Algebraic semiotics tries to synthesize the best of each approach with some recent advances in computational semantics.

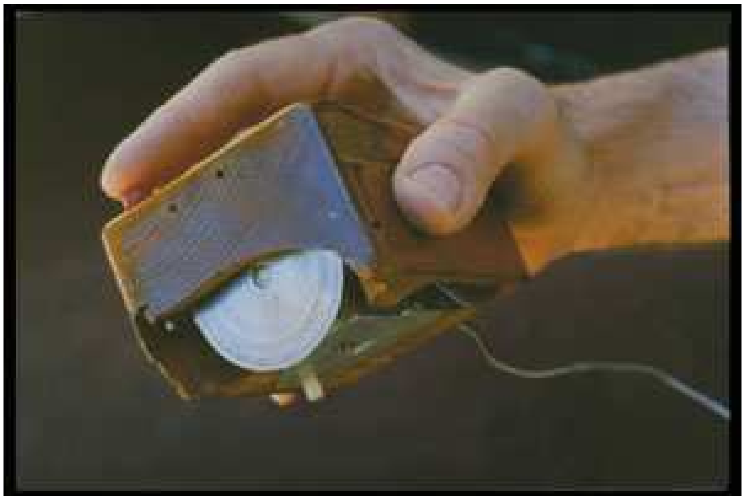

Figure 2: The First Mouse

Engelbart (1925–) is the great pioneer of computer interface design. Unfortunatly, he is far less known than he should be, despite holding many awards, including the Turing Award. Although

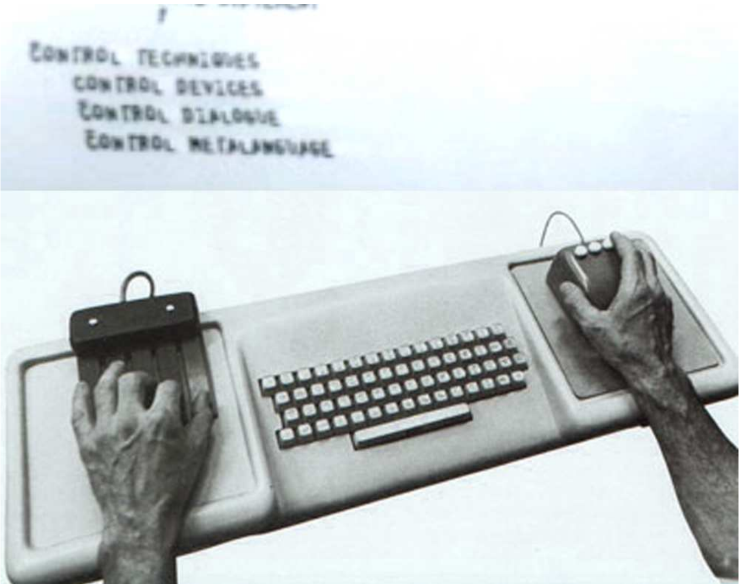

Figure 3: Menu Selector, Keyboard, Mouse, and Video Display

best known for inventing the mouse (Figure 2 shows the first mouse), he did far more than that. A legendary 90 minute show in San Fransicso on 9 December 1968 demonstrated hyperlinks, remote video conferencing and collaboration, windows, menus, and remote search and retrieval, in a system called NLS that had been under development since 1962 (Figure 3 shows NLS in use at the San Francisco demo). On 29 October 1969, his lab at Stanford Research Institute (now called SRI) became one end (with UCLA) of the first link of the Arpanet, which of course evolved into today's internet. Later, he founded and ran Arpa's Network Information Center, using NLS. His approach, called "Human Intelligence Augmentation," is a human centered approach that differs greatly from the "Artificial Intelligence" goals of replacing humans by computers for specific tasks (often military oriented tasks). Around 1950, Engelbart began developing the strong value system that motivated all his subsequent research. He first asked, "How can my career maximize my contribution to mankind?" and decided that developing and using computers and networks to augment human intelligence for solving the world's increasingly urgent and complex problems was a timely and realistic opportunity. Engelbart is therefore also an important precursor to what we call value-driven design.

1.6.3 Special Note for Instructors

In teaching the material in this book, we found that many computer science students expect a greater ratio of programming over reading and theory than we provide: they don't really want to design, they want to start writing code. We therefore found it helpful to repeatedly emphasize the importance of good design for good systems, in a variety of ways, e.g., the introduction to Chapter **??**, which discusses analogies to (and differences from) mature engineering disciplines, such as electrical and mechanical engineering.

1.6.4 Acknowledgements

We are very grateful to the students and TAs in UCSD Computer Science and Engineering classes 171 and 271 for their enthusiasm and feedback on this material. We thank Ben Shneiderman for some valuable suggestions. We have also drawn inspiration and some content from the online semiotics tutorial prepared for CSE 171 by Dana Dahlstrom and Vinu Somayaji.

1.7 Other Resources

There is an enormous literature on interface design, including very many books and journals, and very many websites. Here we list just a small number that seem especially relevant to this book.

- **User-Centered Website Development**, by Daniel McCracken and Rosalee Wolfe, Prentice-Hall, 2003. ISBN 0-13-041161-2. This is close to the spirit of our book, in its elegant combination of pragatic orientation, sound relevant theory, and good examples; we have used it in the past as our textbook, and we highly recommend it for supplemental reading.
- **Designing the User Interface**, by Ben Shneiderman and Catherine Plaisant, Addison Wesley, 2004 (fourth edition). ISBN 0-321-19786-0. This is the most comprehensive general reference in the field, a widely used standard, that we have also used as a text in the past; it is highly recommended for supplemental readeing.
- **Human Values and the Design of Computer Technology**, by Batya Friedman, editor. Chicago/CSLI, 2004. An excellent resource also close in spirit to this course; highly recommended for browsing.
- **Interaction Design**, by Jennifer Preece, Yovonne Rogers, and Helen Sharp, Wiley, 2002. ISBN 0-471-49278-7. A classical psychology oriented approach to interface design; recommended for diversity.
- **Metaphor and Iconicity: A cognitive approach to analyzing texts**, by Masako Hiraga, Palgrave Macmillan, 2005. ISBN 1-4039-3345-6. This is the most sophistiated and inspiring text we know on Peircean semiotics applied to literature.
- **The Way We Think**, by Gilles Fauconnier and Mark Turner, Basic Books, 2002. ISBN 0-465-8785-X. A basic popular text on cognitive linguistics and blending theory; highly recommended.
- **The Design of Everyday Things**, by Donald A. Norman, Doubleday, 1988. A charming and well illustrated short volume on user interface design for everyday objects; fun to read, and informative.
- **Information Visualization**, by Robert Spence, Addison Wesley, 2001. ISBN 0-201-59626-1. A good survey but a bit out of date now.
- **Understanding Comics**, by Scott McCloud, Harper Collins, 1993. ISBN 0-06-097625-X. The best book we know on semiotics that does not mention that word at all! Highly readable and a lot of fun, especially on the use of space and time.
- **Introduction to Barthes**, Mireille Ribiere (Hodder and Stoughton, 2002). ISBN 0-340-84499-X. An excellent introduction to post-structuralist French semiotics; short, cheap and clear; highly recommended for diversity.

- **Aramis, or the Love of Technology**, by Bruno Latour, Harvard University Press, 1996. Perhaps the best introduction to classical actor-network theory, written in the form of a novel in which the "hero" is a failed Parisian suburban transport system. Enjoyable as well as informative and unusual.
- **Requirements Engineering: Social and Technical Issues**, ed. Marina Jirotka and Joseph Goguen, Academic Press, 1994. A sophisticated survey of ways to obtaining requirements and values for computer-based systems, with some novel ideas on socially situated abstract data types; recommended for browsing.
- **Algebraic Semantics of Imperative Programs**, by Joseph Goguen and Grant Malcolm, MIT Press, 1996. Mathematical background for the algebraic semiotics that informally underlies this book; recommended for those with some background in abstract algebra.
- **Life Stories: The Creation of Coherence**, by Charlotte Linde, Oxford, 1993. Deep background with many examples on narrative theory; enjoyable and informative to read.
- **The Yale Style Manuual**, perhaps the best general style manual currently on the web, at `info.med.yale.edu/caim/manual/`.
- **The UCSD Semiotic Zoo**, at `www.cs.ucsd.edu/ goguen/zoo`; an exotic collection of poor designs captured in the wilds of academic discourse.
- **The interface Hall of Shame**, some bad examples of GUI design, mostly OS's and browsers; at `www.pixelcentric.net/x-shame`.
- **Website on blending** and metaphor, by Gilles Fauconnier, Mark Turner and many others, on cognitive linguistics; at `www.wam.umd.edu/ mturn/WWW/blending.html`.
- Homepage of **Interactions** magazine, published online by ACM.

There are many many specialized online resources, of highly variable quality, as well as many specialized books and journals; it is well worth spending some time browsing. The websites of Ben Shneiderman and Don Norman are especially recommended for their further links and their interesting content.

1.8 Exercises

The purpose of these exercises is to orient you towards some issues that will be important for this course, and that will be explored in much more detail later.

1. Search the internet (excluding other material by the authors of this book) for uses of *values* in design, and write a short essay (about one page) comparing what you find with the approach of this book.
2. Search the internet (excluding other material by the authors of this book) for uses of *semiotics* in design, and write a short essay (about one page) comparing what you find with the approach of this book.
3. Describe an example of a complex sign and indicate some of its parts and subparts. Pick a *specific* real sign (e.g., "this book" rather than "a book" – but do not use this particular book!). Also describe a *family* of real signs of which your sign is a member in a some significant way.

4. Observe a computer user at a typical task, such as reading email, debugging a program, or playing a game; after the task is completed, interview the user, and try to discover some values that are significant for that user at that task. Obtain written permission from the user. Ideally, you should record the session and analyze the recording. (Detailed guidelines for conducting and analyzing such sessions are given in Section **??**.)

References

[1] Batya Friedman, editor. *Human Values and the Design of Computer Technology.* Chicago/CSLI, 2004.

[2] James Gibson. *An Ecological Approach to Visual Perception.* Houghton Mifflin, 1979.

[3] Edwin Hutchins. *Cognition in the Wild.* MIT, 1995.

[4] Bruno Latour. *The Pasteurization of France.* Harvard, 1988.

[5] Thomas Kuhn. *The Structure of Scientific Revolutions.* Chicago, 1962.

[6] Charles Saunders Peirce. *Collected Papers.* Harvard, 1965. In 6 volumes; see especially Volume 2: Elements of Logic.

[7] Ferdinand de Saussure. *Course in General Linguistics.* Duckworth, 1976. Translated by Roy Harris.

[8] Ben Shneiderman and Catherine Plaisant. *Designing the User Interface.* Addison Wesley, 2005. Fourth edition.

[9] C.P. Snow. *The Two Cultures and the Scientific Revolution.* Cambridge, 1959.

[10] Lev Vygotsky. *Mind in Society.* Harvard, 1985.