Algebraic Semiotics and User Interface Design

Joseph A. Goguen

Department of Computer Science & Engineering
University of California at San Diego

ABSTRACT

HCI lacks scientific theories for design; so new media, new metaphors (beyond the desktop), new hardware, non-standard users (e.g., with disabilities) can be challenging.

Semiotics seems natural, but (1) lacks mathematical basis, (2) considers single signs (novels, films, etc.), not representations; (3) doesn't address dynamic signs, or (4) social issues, e.g., for cooperative work.

Algebraic semiotics defines sign system & representation, gives calculus of representation & representation quality.

Case studies on browsable proof displays, scientific visualization, natural language metaphor, blending, humor.

Social foundation uses ideas from ethnomethodology.

Outline

1. Motivation: Some Problems
2. Algebraic Semiotics
3. Calculus of Representation
4. Case Studies
5. Summary & Future Research

1. Motivation: Some Problems

Most HCI results are:

- specialized & precise (e.g., Fitt's law), or else
- general but of uncertain reliability & generality (e.g., protocol analysis, questionnaires, case studies, usability studies).

What we need are scientific theories to guide design, e.g., for

- new media,
- new metaphors (beyond the desktop),
- new hardware,
- non-standard users (e.g., with disabilities).

Semiotics, the general theory of signs, seems natural for a general HCI framework. But it

1. does not have mathematical style & so does not support engineering applications;
2. only considers single signs or sign systems (e.g., novel, film), not representing signs in one system by signs in another, as needed for interfaces;
3. has not addressed dynamic signs, as needed for user interaction;
4. has not considered social issues, as arise in cooperative work;
5. ignores the situated, embodied aspect of sign use.

2. Algebraic Semiotics

Algebraic Semiotics provides:

- precise algebraic definitions for sign system & representation;
- calculus of representation, with laws about operations for combining representations;
- precise ways to compare quality of representations.

Have case studies on browsable proof displays, scientific visualization, natural language metaphor, blending, & humor.

Social foundations grounded in ideas from ethnomethodology: semiosis, the creation of meaning, is situated, embodied, etc.

2.1 Signs and Sign Systems

- Signs should not be studied in isolation, but rather
- as elements of systems of related signs, e.g., vowel systems, traffic signs, alphabets, numerals, numbers.
- Signs may have parts, subparts, etc., of different sorts.
- Sign parts may have different saliency, determined by how constructed.

Signs become what they are by having different attributes than other signs – clear from machine learning of patterns.

Same sign in different system has different meaning – e.g., alphabets.

Combines ideas of Peirce (sign), Saussure (structure), Goguen (ADTs).

Formalize sign system as algebraic theory with data, plus 2 specific semiotic items:

- signature for sorts, subsorts & operations (constructors & selectors);
- axioms (e.g. equations) as constraints;
- data sorts & functions;
- levels for sorts;
- priority ordering on constructors.

Sorts classify signs, operations construct signs, data sorts provide values for attributes of signs, levels & priorities indicate saliency.

This is not the formal version; also not necessarily final.

Differs from approaches of Gentner, Carroll, etc. - axiomatic with loose semantics, not set-based; gives a language, not a model; this allows partial models, open structure, etc.

2.2 Representation

User interface design means designing good representations.

E.g., GUIs represent functionality with icons, menus, etc.

Basic insight: representations are maps $M : S_1 \to S_2$ of sign systems, called semiotic morphisms, preserving as much as reasonable:

- sorts & subsorts,
- ops, preserving source & target sorts,
- axioms to consequences of axioms,
- data & functions,
- levels of sorts,
- priority of constructors.

“Reasonable” qualification due to need for tradeoffs.

2.3 Simple Examples

1. S_E – English sentences.
2. S_T – parse trees for English sentences.
3. S_P – printed page format.
4. $P: S_E \to S_T$ – parsing.
5. $H: S_T \to S_P$ – phrase structure representation.

   Time flies like an arrow.

   $[[time]_{\mathrm{N}}[[flies]_{\mathrm{V}}[[like]_{\mathrm{P}}[[an]_{\mathrm{Det}}[arrow]_{\mathrm{N}}]_{\mathrm{NP}}]_{\mathrm{PP}}]_{\mathrm{VP}}]_{\mathrm{S}}$.

Can't always preserve everything - resulting display may be too complex for humans.

And sometimes just want to summarize some data set.

2.4 Quality of Representation

Content means values of selector ops, e.g., size, color.

- Easy to define sort preserving, constructor preserving, level preserving, content preserving, etc.
- But not very useful since often are *not* preserved.
- Instead, define more sort preserving, more level preserving, more constructor preserving, more content preserving, etc.
- These comparative notions define orderings on morphisms.
- Can combine orderings to get right one for given application.
- Given S, S', one may preserve more levels, other more content.
- More important to preserve structure than content.
- More important to preserve levels than priority.
- Also it's easier to describe structure.

3. Calculus of Representation

Can compose morphisms & so study composed representations, as arise in iterative design. Have identity & associative laws:

$$A\ ; 1_S = A$$
$$1_S\ ; B = B$$
$$A\ ; (B;C) = (A\ ; B)\ ; C$$

Therefore have a category.

This gives other simple laws, plus notions: isomorphism of sign systems, sum & product of sign systems & representations, plus much more (see following).

3.1 Blending

Fauconnier & Turner studied blending metaphors, using conceptual spaces – sign systems with only constants & relations.

Conceptual blend of maps with same source, the generic space, & targets called input spaces, combining their features in blend space.

$$\begin{tikzcd}
 & B & \\
I_1 \arrow[ur] & & I_2 \arrow[ul] \\
 & G \arrow[ul] \arrow[ur] \arrow[uu] &
\end{tikzcd}$$

We generalize to arbitrary sign systems, morphisms, & diagrams.

Examples: house boat; road kill; computer virus; artificial life; jazz piano; conceptual space; blend diagram; ...

Blend diagram suggests categorical pushout – but doesn't work, since blends not unique.

Example: "house $\diamond$ boat" has 4 different maximal blends:

1. houseboat;
2. boathouse;
3. amphibious RV;
4. boat for moving houses (!).

But since ordered category, use "lax" pushout:

- has non-unique result; and
- can actually calculate the 4 blends above!

Order by $f \leq g$ iff g preserves as much content as f, as many axioms as f, and is as inclusive as f.

3.2 Some Laws

$$
\begin{aligned}
A \times \mathbb{1} &\cong A \\
\mathbb{1} \times A &\cong A \\
A \times B &\cong B \times A \\
A \times (B \times C) &\cong (A \times B) \times C
\end{aligned}
$$

$$
\begin{aligned}
a \diamond b &\cong b \diamond a \\
a \diamond (b \diamond c) &\cong (b; a) \diamond c \\
(a \diamond b) \diamond c &\cong a \diamond (b; c)
\end{aligned}
$$

A, B, C can be either sign systems or semiotic morphisms.

Product is special blend with common space empty; sum of theories gives model product. So product laws are special blends laws.

4. Case Studies

1. Blending (already discussed).
2. Metaphor (similar to Fauconnier & Turner).
3. Scientific visualization.
4. Proof presentation.
5. Humor.

So we will do items 3, 4, 5.

4.1 Scientific Visualization

Visualizations of complex data help scientists discover, verify & predict patterns.

Difficult to construct "appropriate" visualizations.

But visualizations *are* representations & our quality measures apply; best to use in semi-formal style:

1. use ideas & results to guide examination;
2. use formalism only if needed for difficult design decision.

Two examples illustrate techniques:

1. code visualizer.
2. movie visualizer.

Able to suggest improvements in both cases.

4.2 Proof Presentation

- Understanding proofs is notoriously difficult. Why?
- Tatami project views proofs as representing underlying math.
- Then can apply algebraic semiotics, quality measures, etc.
- But what is the underlying math?
- Important ingredients include:
  1. narrative (Labov & Linde).
  2. drama – Aristotle said “drama is conflict.”
  3. image schemas (Lakoff & Nunez).
- Proofweb data structure includes narrative & conflict, as well as formal sentences & inference rules.

See www.cs.ucsd.edu/groups/tatami/kumo/exs/.

4.3 Humor

Studied corpus of over 50 humorous oxymorons —
"military intelligence," "good grief," "almost exactly," ...

"Oxymoron" is phrase with contradictory (or incongruous) terms.

Humorous oxymorons: conventional & contradictory meaning.
i.e., 2 different blends, one with conflicting elements.

Studied over 40 newspaper cartoons – about 3/4 have same pattern.

So this seems a general facet of humor.

Note that humor is used in many interfaces, often badly.

5. Summary & Future Research

Algebraic semiotics seems promising for user interface design & can handle metaphors, blends, humor.

But much more work is needed:

- More case studies, more carefully done.
- Dynamic signs for user interaction – use hidden algebra.
- Combine Gibsonian affordances with algebraic semiotics.
- More on narrative structure.
- More on social foundations, semiosis.
- How to choose orderings on representations?