

Discrete Representations and Continuous Design: The Normative Coordination of Activity

Hamid R. Ekbia

Computer Science Department
University of Redlands
hamid_ekbia@redlands.edu

A Position Paper
for
Continuous Design Workshop
University of Illinois
Urbana/Champaign, IL.
Oct. 8–9, 2003

In his book *The Sciences of the Artificial* Herbert Simon introduces and advocates a “science of design,” as an empirical science that “uses the artificial to study the natural.” Central to Simon’s notion of a science of design is the idea of bringing together the concepts and theories of science and the tools and models of engineering. A successful example of such a science, in Simon’s mind, is Artificial Intelligence, which has strong elements of both science and engineering (Simon 1995: 100)

Simon’s idea of a science of design expands the traditional boundaries of science beyond the “pristine positivist position,” which limited the realm of science to “natural” phenomena proper. It extends these boundaries to the study of artifacts, despite the fact that, as he notes, “the natural sciences are concerned with how things are” and “design is concerned with the way things ought to be” — the former is descriptive, the latter prescriptive (1980: 132–13). It seems to me that this extension of the proper realm of science is a move in the right direction. We could and we should have a science of design. It also seems to me that a *science of design* is, in fact, what information scientists are striving for.

Ironically, however, when it comes to methodology, Simon’s allegiance to the positivist view limits, in my opinion, the scope of his account. This is best manifested in the dictum “of the irreducibility of “ought” to “is”” (1980: 8). In line with this dictum, Simon seeks to separate the prescriptive from the descriptive, maintaining that a declarative logic such as predicate calculus is sufficient for a science of design as it is, in his opinion, for the sciences of the natural. The inadequacies of the logical framework for understanding the processes of knowledge creation in natural sciences is widely discussed by philosophers of science after Kuhn (e.g., Cartwright 1985, Giere 2000), by sociologists of science after Merton (e.g., Collins 1990, Latour 1987, Pickering

1990), and by the students of science and technology studies (Callon 1986, Latour 1999, Bijker et al 1997). The limits of the logical framework as a foundation for “intentional” sciences (e.g., AI, computer science, information science, psychology) has also been a topic of recent investigation. A common finding of these investigations is the inadequacy of the traditional views of representation (Agre 1997, Hutchins 1995, Smith 1996, 1998, 2002, Woolgar 1995). One of the lessons of these findings, I believe, is that we need a methodology that will not dichotomize between the descriptive and the prescriptive, and that will not marginalize the significance of normative values in our understanding of science and technology. Rather, it would give a central role to normative values, constraints, and expectations.

As was mentioned earlier, a revised notion of representation is central to such a project. What follows is an outline of an attempt in this direction, especially as it relates to the issues of Open Source Software Development (OSSD). Contrary to the Simon’s view, the idea is that designs normatively governed, that representations are normatively constituted, and that there are various physical and cultural norms involved in the creation, use, transformation, and coordination of representations. This view, I believe, is compatible with some of the recent accounts of representation in philosophy (Smith 2002), science studies (Latour 1999), AI (Agre & Chapman 1990), and cognitive science (Hutchins 1995).

Before providing the outline, to emphasize the normative aspect of design, I would like to draw upon an announcement of the last year’s conference of the European Association for the Study of Science and Technology (EASST) in York, UK, under the main theme of *Responsibility under Uncertainty: Science, Technology, & Accountability*:

The move towards a politics of and policy for science, technology, medicine and engineering that recognizes, but is not paralyzed by, uncertainty has marked recent debate...Some have argued that science and technology occupy a position in the public sphere that demands their implications be considered at the outset, and not simply when developments are at a point of application. It could be argued that there is a shift towards what can be called anticipatory science and technology.

I. The Problematic

Current accounts of representation — mainly as a mapping from a given real-world event to an explicit simulacrum in a cognitive system — are based on certain assumptions about the character of our universe, about cognition, and about human society, roughly:

1. The world is a set of objects with fixed properties and relations among them;
2. Cognition is the abstract activity of a mind instantiated in a brain that is housed in a skull, and hence isolated from the world;
3. Human society is a collection of individuals who, largely driven by the dictates of an evolutionary process (natural selection), behave rationally in order to maximize their individual profits.

Accounts of representation based on these assumptions fail in many ways — e.g., (in areas of concern to us) in dealing with an enormous part of human activity that goes beyond pure abstract logical reasoning (“cognition in the wild”; Hutchins 1995), in capturing essential features of the extensive activities performed in the name of computing in contemporary societies (“computation in the wild: Smith 1996), and in explaining how large-scale socio-technical systems work (Gasser and Ripoche 2003; I propose “design in the wild” to refer such systems).

Therefore, the challenge is to develop an account of representation that can explain how cognitive (intelligent) actors make sense of events in the outside world, how meaning and artifacts are co-constituted, and how representations make collective action possible. Simply put, the challenge is for us, as analysts, to be able to provide accounts that would do justice to these widespread activities, without killing their spirit. As far as OSSD is concerned, this amounts to three key questions:

1. A philosophical question about the role of “representations” in OSSD;
 - e.g., bug reports as *representations* of bugs
2. A sociological question about the organization of work, the mechanisms of collective action, and the politics of cooperation and conflict in the development of software;
 - e.g., the collective processes of bug reporting and resolving
3. A technical question about the processes of design and innovation
 - e.g., processes underlying software development, quality assurance, modeling and automation,

II. The Thesis

The main thesis is that representations are *normative vehicles of coordination* among different actors and through different mediums. The actors can be both human and nonhuman, the mediums can be natural and artifactual, and the norms can be physical (“laws of nature”, spatio-temporal constraints, etc.) and cultural (rules, procedures, values, etc.)

III. The Scope

The above questions are interrelated because

1. In order to understand the sociological question (e.g., of the function of bug reports, as mediums of interaction among actors), one needs to understand the question of how these reports act as stand-ins for the bugs (see III.1 below);
2. In order to understand how bug reports represent bugs, one needs to understand the social/organizational processes that are involved in making sense of the reports, in making decisions, in prioritizing, and so on;
3. In order to succeed in design, one needs to accomplish a thorough understanding of both the philosophical and the sociological questions; good design must be grounded in good theory.

There are prior and current studies of the second (e.g., Elliot & Scacchi 2002, Gasser and Ripoche 2003, Shirky 2003) and third questions (e.g., Gasser 2003, Jensen and Scacchi 2003, Scacchi 2003), so I propose to focus on the first question about the character of representations. There are also interestingly related economical and business questions that have been studied and might contribute to the proposed study (see, for instance, Benkler 2002, Kenwood 2001, PITAC 2000, Weber 2000).

IV. The Framework

The framework adopted here is based on certain key assumptions — namely, it is:

1. Realist: There is a world out there.
 - e.g., bugs are, by and large, “real”
 - There is a problem somewhere in the operation of Mozilla, as perceived and judged by the user(s)
 - There is more to a bug than a bug report can represent
2. Constructive: The world and the representations of it mutually constrain each other.
 - e.g., Bugs, bug reports, and the relations among them are not stable entities; they are coconstituted by different mechanisms of *translation* (Callon 1986) and *circulation* (Latour 1999) between different representation media
3. Contextual: there is a strong deictic component in (almost) every representation;
 - The “behavior” of a bug is contingent on place and time; this equally applies to user perception as it does to hardware, platform, etc.
4. Collective: the process of making sense of the world is largely collective, with different human and nonhuman actors taking part in it.
 - e.g., bugs are dealt with in collective activities of sense-making, cooperation, conflict, ...
 - The key process is one of *coordination* among different actors and representations
5. Normative
 - The perception of the user(s) is normative
 - i.e., it is relative to their expectation about how Mozilla should “work”
 - The notion of “working” is both physically and culturally normative
 - It has a causal component
 - It has a semantic component

V. Applications

Possible applications of this revised notion of representation for design in general and for software design in specific become immediately apparent if we notice the ubiquity of representations in the form documents, reports, plans, etc.

1. **Documents:** The character and role of documents, as representations, in the software development process (especially in document-driven systems such as DOD) should be seen in a new light, as guiding activity and as mediums of coordination rather than as faithful replica of external events;
2. **Planning:** The meaning and role of planning in the design process needs to be revised from one of a means for achieving a preset goal to one of intertwined routines and improvisations;
3. **Errors:** The significance of errors as deviations from pre-established standards will be totally revamped to the opening of the space of possibilities, to accommodate the notion of “design for error” (Norman 1983, Hutchins 1995).