Design, Design Representation and Design Education

Clive L. Dym
Department of Engineering
HARVEY MUDD COLLEGE
Claremont, CA 91711
clive_dym@hmc.edu

I. *The workshop should encourage the NSF and other funding agencies to take a broad, expansive and inclusive view of design, incorporating a diversity of ideas, models and representations.*

Drawn too narrowly, the notion of a "Science of Design" may be counterproductive. The engineering design community has recently gone through some major controversy, including the breakup of a new Gordon Conference series on the "Science of Design" because of an attempt to define "good science" as that which could be supported axiomatically and could be reduced to economic decision theory (e.g., von Neumann and Morgenstern, Arrow). Such a reductionist approach is inherently contradictory because design is very much a *synthetic* endeavor. Much of what we do is heuristic (and can be modeled or represented declaratively rather than procedurally). Much of design thinking, work and research is in this mode. Further, as a result of modeling how people think while designing, we have developed teachable models of doing design. It is not that formal mathematics (and decision theory) have no role in the science of design; but they *should not* have an exclusive role because much of what people do is not susceptible to mathematical modeling. Similarly, we should be cautious about correlations since there is a very human tendency to confuse correlations with causality.

II. *More can be done in terms of computational modeling of engineering design tasks through integration at the representational level.*

Programming styles based on symbolic representation allow the representation of significantly different engineering knowledge — especially descriptive, heuristic, and causal knowledge — that cannot be expressed algorithmically. The integration of numerical, graphical and symbolic programming provides the opportunity to incorporate into computer-based tools knowledge that is far deeper and far broader than what was available through traditional numerical tools. These kinds of knowledge include disciplinary fundamentals, specific modeling knowledge, different abstractions of behavior, heuristic knowledge derived from practice and specifications, and meta-knowledge about how and where to invoke the other kinds of knowledge. Thus, there exists the opportunity to identify and integrate the different kinds of knowledge we bring to bear in doing engineering, to identify the appropriate representations for these knowledge types, and to integrate them into a powerful computational engineering modeling environment.

III. *Basic conceptual design principles and methods should be a fundamental element of basic software engineering courses, if they're not already included.*

I've long argued that *every* first-year engineering student should take a course in engineering design, and such courses are becoming more widely taught. I assume that software engineering is about designing computer codes that achieve articulated objectives by performing specified functions, within constraints? If that is the case, then basic conceptual design principles should be a fundamental element of basic software engineering courses, if they're not already included. It is reasonably likely that when Simon so famously said, "Design is the distinguishing feature of engineering," he would have included software engineering!

It is worth reinforcing a point made in this connection by David Parnas. Engineers do not normally start with clear statements of objectives and constraints. Their initial design work aims at producing such a statement because it is *a key step* on the way to a final design. In our first-year design course, we note that clients often pose their "problems" in a ***problem statement*** that may: contain biases and errors, be incomplete, or imply solutions. Thus, designers must first review and revise the client's problem statement, and conduct further interviews and research to elicit a deeper understanding of the problem that is expressed in a ***revised problem statement*** and in subsequent ***lists of objectives and of constraints***.

Name: Clive L. Dym
Fletcher Jones Professor of Engineering Design, Harvey Mudd College, 1991–present
Director, Center for Design Education, Harvey Mudd College, 1995–present
Chair, Department of Engineering, Harvey Mudd College, 1999–2002

Degrees: B.C.E., The Cooper Union, June 1962
M.S. (Applied Mechanics), Polytechnic Institute of Brooklyn, June 1964
Ph.D. (Aeronautics and Astronautics), Stanford University, January 1967

Previous Full-Time Appointments:

1977–91	Professor of Civil Engineering, University of Massachusetts
1977–85	Head, Department of Civil Engineering, University of Massachusetts
1974–77	Senior Scientist, Bolt Beranek and Newman, Inc.
1970–74	Associate Professor of Civil Engineering, Carnegie Mellon University
1969–70	Research Staff Member, Institute for Defense Analyses
1966–69	Assistant Professor of Engineering and Applied Science, SUNY–Buffalo

Principal Relevant Publications:

C. L. Dym, "Expert Systems: New Approaches to Computer-Aided Engineering," *Engineering with Computers*, *1* (1), 9-25, April 1985.

S. Mittal, C. L. Dym and M. Morjaria, "PRIDE: An Expert System for the Design of Paper Handling Systems," *Computer*, *19* (7), 102-114, July 1986.

C. L. Dym, R. P. Henchey, E. A. Delis and S. Gonick, "A Knowledge-Based System for Automated Architectural Code-Checking," *Computer-Aided Design*, *20* (3), 137-145, April 1988.

C. L. Dym and R. E. Levitt, *Knowledge-Based Systems in Engineering,* McGraw-Hill Book Company, New York, 1991.

C. L. Dym, *Engineering Design: A Synthesis of Views*, Cambridge University Press, New York, 1994.

C. L. Dym, "Representing Designed Objects: The Languages of Engineering Design," *Archives for Computational Methods in Engineering*, *1* (1), 75-108, 1994.

C. L. Dym and P. Little, *Engineering Design: A Project-Based Introduction*, John Wiley & Sons, New York, 1999 (1st Edition), 2003 (2nd Edition). Spanish translation published in 2002.

C. L. Dym, "Learning Engineering: Design, Languages, and Experiences," *Journal of Engineering Education*, *88* (2), 145-148, April 1999.

C. L. Dym, W. H. Wood and M. J. Scott, "Rank Ordering Engineering Designs: Pairwise Comparison Charts and Borda Counts," *Research in Engineering Design*, *13*, 236–242, 2002.

Recent Related Professional Service:

National Research Council's USARL Technical Assessment Board, 2001–04.
    Member (2001–) and Chair (2003–), Air, Ground and Vehicle Technology Panel.
ASME Design Theory and Methodology (DTM) Conferences —
    Review Coordinator (1999–2001), Program Chair (2002) and Conference (2003) Chair.
Chair and Organizer, Mudd Design Workshops I–IV (1997, 199, 2001, 2003).