1988

The Role of Design in Liberal/Professional Education

Preston K. Covey
Carnegie Mellon University, dtrollcovey@gmail.com

Follow this and additional works at: http://repository.cmu.edu/philosophy
Part of the Philosophy Commons
The Role of Design in Liberal/Professional Education

Preston K. Covey, Editor
Vice Provost for University Studies

[DRAFT - DRAFT - DRAFT]
Contents

Introduction - John P. Crecine [incomplete]

Teaching Design at CMU: From the Carnegie Plan to the Present - W. Andrew Achenbaum 1

The Science of Design - Herbert A. Simon 23

Defining Design Across the Arts & Sciences - Thomas A. Schwartz 47

Academic Writing: Designing for the Community - David Kaufer 65

Problem-Solving in Writing: Can We Teach 'Design' as a Cognitive Process?
   - Linda Flower 89

Design Invisible, Designed Visible - Akram Midani 105

Why Teach Design? - Stefani Ledewitz 113

The Design Process in Music Composition - Marilyn Thomas 137

Reflections on the Experience of Design - J. M. Ballay 161

On Engineering Design - Sarosh Talukdar & Art Westerberg 169

Hierarchical Design Strategies in 3-Dimensional Chemistry - Jonathan S. Lindsey 181

Design and Implementation of Computer Systems - Alfred Z. Spector 219

Logical Structure and Design - Wilfried Sieg & Preston K. Covey [incomplete]
1. TEACHING DESIGN FROM THE CARNEGIE PLAN TO THE UNIVERSITY CORE

W. Andrew Achenbaum

The Foundations of Our Liberal/Professional Tradition

Education, like other growing human institutions, requires timely adjustments...Our problem is to maintain education in an optimum position in the moving procession of civilization such that it will neither find itself overtaken by the rear guard nor so far ahead that it is out of intimate touch and perhaps even off track. That position should indeed always be one of leadership; but in order for it to be this, education must not remain too long in a settled state.

— Robert E. Doherty (1950)

Robert E. Doherty graduated from the University of Illinois in 1909 with the vague notion that his training was inadequate: "I had learned technical book knowledge well enough to reproduce it; I had learned how to solve special types of problems of how to carry out certain important routines. But something was missing. It was a genuine understanding of the significance of fundamental principles and concepts." Through rote learning Doherty had gained a journeyman's appreciation of the art of the possible: he knew what American engineers around the turn of the century thought they could and could not do. Yet nothing in his education had prepared him to think rigorously about how he could use the real tools of his trade— ideas— to design a new world.

Doherty really did not begin to learn how to learn until he joined the General Electric Company. There, he worked closely with Dr. Charles P. Steinmetz, who taught him a new approach to problem solving. Instead of giving him the answer, Steinmetz forced Doherty to design his own intellectual strategy for tackling the unfamiliar. Gradually, Doherty discovered how to apply knowledge he already had in defining and resolving a problem at hand. Invariably this entailed grappling with the basics as he envisioned the big picture. He then had to proceed by trial and error, ever seeking to transfer skills and concepts he knew to new situations. This intuitive approach proved efficacious: Over the next two decades Doherty gained a national reputation as a design engineer, laying the foundations for modern synchronous machine theory.

Based on his own experiences as a consultant and entrepreneur, Doherty believed that all professionals gained expertise through a learning-by-doing approach to problem solving. Thus with Steinmetz's encouragement, he designed an "advanced course in engineering" to persuade young engineers that book learning per se was not the key to success. Rather, a well-ordered
balance between the traditional objectives of a liberal-arts education and the practical
requirements of real-life vocations. The Corporation was authorized to confer degrees "in pure
and applied science and the arts." C.I.T. adopted a four-year curriculum similar to that
offered by other engineering schools at the time. Yet in keeping with the innovative and
unorthodox style of its chief benefactor, the instruction offered here did not slavishly conform
to national patterns.

The Institute's organization and programs were unique. For the first time in the western
world, classes were offered in architecture, art, design, drama, and music under one roof. The
Margaret Morrison Carnegie Division, a technical college for women, offered degrees in
secretarial work and social service as well as home economics. By the end of Arthur
Hamerschlag's directorship, a Division of Science and Engineering granted baccalaureate and
graduate degrees in seven branches of engineering, while a separate school of industrial
engineering gave courses to prospective executives and teachers. There also was the Division of
Applied Psychology, a special institute of educational and industrial research; and a night
school, which taught printing and other technical skills. Under President Thomas S. Baker,
further organizational reshuffling took place; among other things, the College of Industries
merged with the College of Engineering. In addition, the promotion of scholarship became an
explicit priority. More emphasis was placed in the undergraduate curriculum on writing
English, and the range of graduate training extended to coal and metals research.3

Given the rapidity of organizational change and shifts in curricular priorities from 1900 to
1936, it is hardly surprising that no coherent or consistent vision of education dominated
Carnegie Tech's formative years. Paradoxically, however, all of the experimentation (and
confusion) set the stage for a major curricular reform.

General-education objectives, by happenstance more than by intent, commingled with the
predictable commitment to specialization in technical fields. Courses emphasized practical
know-how, but they also were to be "sufficiently broad and flexible to almost encroach upon
the professional."4 Official histories and archival data do not indicate that a rationale was ever
formulated that guided professors and administrators as they put together a curriculum which
approached but did not encroach upon "professional" education. The record nonetheless
reveals an interesting pattern. Beginning in 1908, women in Margaret Morrison were directed to
plan schedules that included instruction in history and geography, mathematics and science, and
English. As early as 1919, Carnegie students were devoting more time to non-professional
subjects than undergraduates in civil, electrical, and mechanical engineering at Cornell,
Pennsylvania, Purdue, Wisconsin, and Boston Tech.
method of using the creative power of ideas would enable them to solve real-life engineering problems in a practical and profitable manner. In essence, Doherty was groping toward a science of design: he tried to teach newly recruited GE employees how to discern truly important problems out of a maze of seemingly trivial bits of information and fit them into an overall scheme. Once they were able to think logically building on knowledge they already possessed, they then could derive well-founded conclusions that in turn would become the basis for tackling ever more challenging problems.

Pleased with the impact of his training program at General Electric, Doherty set out to revolutionize the professional education of engineers. He joined the Yale faculty in 1931 and soon became dean of its newly organized engineering school. Doherty hoped to use this prestigious position as a way to extend and refine ideas he had learned in the "college of hard knocks." But his colleagues resisted many of his radical ideas—especially the notion that students should be socialized to become good citizens as well as professional leaders. Accordingly, Doherty took a big gamble: in 1936, he accepted the presidency of Carnegie Institute of Technology.

From the very beginning "Tech" did not quite fit existing models of higher education. Most of America's best known institutions in 1900 rose from fairly humble origins: their founders wanted to prepare ministers and teachers, or expose the children of provincial elites to a mixture of classics, practical skills, and civics. A few places benefitted from the largesse of tycoons such as William Vanderbilt, Johns Hopkins, John D. Rockefeller, Daniel Drew, and Leland Stanford. Andrew Carnegie ultimately would give $34 million to places like Princeton and generously endow the Carnegie Institution in Washington, a foundation which became one of the country's premier centers of basic scientific research in eighteen different fields—but he did not want to build a liberal arts college or university-level research center in Pittsburgh. Originally, Carnegie's "Technical Schools" were to give youth in Allegheny County training pitched above a grammar-school level but certainly less grand than a baccalaureate-degree curriculum. The Woman's Industrial School taught nutrition and dress-making. Men learned carpentry, plumbing, and electrical repair in the shops of the School for Mechanics and Artisans. Instruction in the School of Fine Arts and Applied Design and in the School of Science and Technology was geared to the marketplace. Within a decade, however, the technical schools were having difficulties recruiting teachers; its students were at a disadvantage in competing for jobs. Thus the Board of Trustees (with Carnegie's approval) petitioned the Commonwealth for a charter to incorporate as the Carnegie Institute of Technology.

Under its 1912 charter, trustees, faculty, and administrators struggled to effect a proper
After World War I, the faculty signalled their willingness to take even bolder steps to ensure that their students were not acquiring skills which were exclusively vocational in nature. A Division of Academic Studies was established to provide courses in general subjects for students in all colleges at Carnegie Tech. The Faculty Senate required in 1933 that at least a sixth of all students' classes be in "cultural courses; those working for a B.A. in architecture, painting, music, decoration or drama spent from 23 to 35% of their time taking classes outside of the College of Fine Arts. In preparation for a site visit in 1935 by a national Committee on Engineering Schools, the Engineering and Science curriculum was evaluated, and efforts made to broaden its purview.  

Thus Tech was to prove to be an ideal laboratory for Doherty's experiment in higher education; he, in turn, would provide the intellectual framework for CMU's inchoate liberal/professional tradition. One can trace the evolution of Doherty's thinking almost from the moment of his arrival on campus. Writing his first official report three months after assuming the presidency, Doherty declared that he wanted to restrict enrollment while upgrading the quality of instruction in the school's 4-year curriculum. He would concentrate first on undergraduate technical branches in the field of engineering; he intended to capitalize on the growing emphasis on science while cultivating a wider intellectual horizon. "All three programs— the undergraduate, graduate, and research— are the requisite elements of a balanced plan of higher learning, and it is our purpose to pursue such a development of these as will maintain a rational relationship of each to the other, and of the whole to the resources of the institution."  

In his second report to the trustees (1936–37), Doherty revealed more of his strategy and operating assumptions. There was to be a reduction in course loads (down to 48 units per semester) and a requirement that students pursue an average of 22% of their time taking courses in the social stem and the rest in a broad area of technological expertise. "Here we would seek first, a broader outlook by the student upon the human, social, and economic sides of life, particularly upon his responsibilities as a citizen and as a professional man; and second, better understanding of all that he studies." Courses in both stems, the president insisted, would have to be upgraded to ensure that students received an intellectual demanding but balanced curriculum which placed a premium on learning by doing.

Doherty did not elaborate further on the ultimate shape of the curriculum, however. His reasons were tactical as well as practical. In part, the president recognized that the reform impulse would be driven by outside forces, including "the extent to which [industry] is willing to accept graduates who have not learned as many routine skills as now, but have minds
fundamentally better trained." In part, the success of the venture depended on the ability of the Institute "to determine the educational outcomes that appear essential, and then rigorously adjust the curriculum procedures to these outcomes." Designing a solid curriculum would take time—at least a decade, the president predicted—and more effort than most professors probably expected.

The process of implementation began with certain advantages. Doherty had been thinking about the basic objectives of a professional education for more than twenty years, and he inherited an infrastructure that already allotted slots for general studies. In 1937, he convinced a local foundation to underwrite his educational vision. "This country needs more professional men who can grapple intelligently with intricate social problems," he wrote. "Engineers, who understand technology, must give thought to consequences. They, in common with lawyers, economists, and other responsible professional people, must consider a new kind of educational preparation for their oncoming generations." Armed with a $300,000 endowment to support the Maurice Falk Professorship of Social Relations, Doherty had the wherewithal to institute a "social relations" program to bridge the liberal and technological branches of an intriguing design for a general professional education. Now he could begin the truly difficult tasks of developing new ways of teaching interdisciplinary course materials and of recruiting faculty who were committed to achieving the Institute's newly reformulated objectives.

Institutionalizing the Carnegie Plan

It is tempting to reconstruct the institutionalization of Doherty's educational reforms in a teleological manner. In retrospect, it almost seems foreordained that they would have a salutary impact here and elsewhere. After all, the American Society for Engineering Education adopted in 1939 a set of objectives that were based on Doherty's goals; less than twenty years later, a writer in Technology, declaring that "the Carnegie Plan is Carnegie Tech," would call the venture a "revolutionary success." Yet, particularly in the midst of our current encore for reform, it is important to recognize that Doherty's success were cumulative. No breakthrough, no matter how minor, was either easy or inevitable.

"Learning by doing" proved to be both the means and the end of pedagogical initiatives. Tech's faculty deserves most of the credit for translating Doherty's vision into a viable curriculum, for it is they who enthused the problem-solving orientation in their classes. A "core" faculty spearheaded efforts: Emerson Pugh '18 chaired a Committee on Course Coordination; Willard E. Hotchkiss, as the first incumbent of the Falk professorship, headed the committee on Social Relations. Pilot courses on "Codes and Legal Regulations," "Literature of Social Significance," and the "Social Nature of Art" were developed. Besides encouraging
the fusion of liberal-arts staples into engineering studies, Doherty invited liberal-arts students to campus: roughly 25% of the Class of 1940 were recruited under a 3–2 Plan, wherein students received a B.A. from a place like Allegheny or Washington and Jefferson College and a B.S. in engineering from Carnegie after five years of study.11

Not everyone liked the changes. Doherty eliminated vocational, shop, and other secondary-level courses that had no place in the new order. Instructors were given the opportunity to take on new tasks, but many left. So did members of the "Old Guard" who were distressed by the president's efforts to "eliminate duplication" and promote "simplification" of requirements so that there were would be room in the curriculum for both social and technological courses.12

Perhaps more damaging than the negative fallout were some of the compromises necessary to realize Doherty's interdisciplinary objectives. It proved difficult to recruit faculty who were qualified to teach "social" courses that incorporated an up-to-date understanding of technological advancements (and vice versa). Within a decade, therefore, a decision was made to recruit faculty for the Social Relations Program "who are deeply conscientious in their teaching and are willing to devote a large part of their energy to giving value to their teaching."13 This seemingly innocuous statement ultimately led to developments that Doherty probably did not intend. Increasingly, those among the faculty who envinced a serious commitment to teach well would be encouraged to fulfill their promise— even if it meant that their research would suffer. Over time, fewer regular faculty in the Science and Engineering colleges took part in interdisciplinary curricular ventures.

Fortunately (both for Doherty and for Tech) the appointment of a new general education officer checked a pernicious impulse to make invidious distinctions between research and teaching and between the technological and social stems of the curriculum. Upon Hotchkiss's retirement Elliott Dunlap Smith, a professor of economics at Yale and master of its Saybrook College, was appointed Falk Professor in 1946. He also was recruited to be C.I.T.'s first provost. Smith's educational philosophy complemented the president's. In writings and speeches he stressed the need to provide undergraduates with a relatively small core of fundamental knowledge which should be ready for use as the basis for solving problems as they arise. Like Doherty, the provost believed that Tech was positioned at "a cutting edge in educational progress" because of its size, faculty and endowment: "Above all, its creative conception of professional education— to equip students to continue to learn and to develop throughout their lives as professional men, as citizens, and as individuals— both demands progress and gives direction to progress."14
By dint of his personality and position, Smith was able to lead by example. A master professor himself, the provost was as interested in coaching his instructors so that they would learn how to become better teachers as he was in ensuring quality in undergraduate instruction. The man apparently had an extraordinary capacity to enjoy meeting with his constituencies; he was always talking to students, faculty, and department heads. Smith also chaired a new Educational Policy Committee (EPC). Consisting of the president ex officio, the secretary of the faculty senate, and deans of the colleges and "Division of Humanistic and Social Studies," this body considered general-educational proposals of major importance. Though Smith recognized the liabilities of his position— in his first report, he acknowledged that the EPC was merely an advisory committee and he bemoaned his inability to retain humanistic and social teachers due to the lack of research opportunities and suitable "residential conditions" in Pittsburgh— he nonetheless quickly was perceived as the President's representative in exercising educational leadership.

Together, Doherty and Smith elaborated and formalized the "Carnegie Plan of Professional Education," as it officially became known in 1948. The problem-solving approach to learning they had developed now represented a viable design for teaching analytic and synthetic intellectual skills that were cumulative and transferrable from one situation to the next:

The work of the student from the beginning of the freshman year until graduation consists in considerable measure of his own application of fundamental knowledge in handling specific problems and in learning from observation and experience....The abilities to deal with professional, social, and human problems are thus merged into a common ability applicable to problems of all sorts. Because of this unity, whatever increase in analytical power the student gains in one field strengthens his power to solve problems and to learn in all other fields; and after graduation, growth in professional stature involves growth in stature as a human being and citizen.15

The intellectual underpinnings and course requirements stipulated in the Carnegie Plan were not viewed as simply a rehearsal of earlier objectives. Nor were they to be considered an adjunct to the status quo. The Plan was to form the core of C.I.T.'s liberal/professional tradition of teaching and learning. The Carnegie Corporation of New York gave more than $200,000 to supplement the Falk endowment. This additional support enabled Smith to give selected faculty members release time to plan new courses, experiment with various techniques, and to revamp the undergraduate curricula in engineering and science.

The Carnegie Plan soon became the rationale for graduate training. In 1948, Smith organized a "conference on education for professional responsibility" in which deans and nationally known administrators met to discuss ways to effect innovation in training lawyers,
ministers, doctors, and scientists.\textsuperscript{17} Tech's Graduate School of Industrial Administration, moreover, was conceived as a logical extension of Carnegie Plan principles. Under the terms of William Larimer Mellon’s $6 million gift, President Doherty and Dean G. Leland Bach were to recruit a faculty committed to doing basic research in general problem-solving. The results of this guiding vision are manifest: Present-day luminaries (including William W. Cooper, Richard M. Cyert, James G. March, Allen Newell, and Herbert A. Simon) launched their careers by formulating behavioral theories of the firm and by applying the power of mathematics and the computer to the analysis of human problem-solving. Far from being bound by traditional models, the intellectual climate and \textit{modus operandi} at G.S.I.A. were to foster the design of an educational program that would enable talented students to acquire basic managerial skills and patterns of thought that would enable them to adapt to an ever-changing business environment.\textsuperscript{18}

As men who were trained at Tech rose to prominence in the corporate ranks at Eastman Kodak, National Supply Co., Gulf Oil, and Westinghouse, both the media and captains of industry looked to C.I.T. to play an expanding role in providing leaders for the American Century. Charles E. Wilson, the president of General Motors and Eisenhower’s secretary of Defense was "convinced that time and a fair sampling of graduates will prove that the Carnegie Plan is the type of college program which can produce the professional man industry and America must have in greater numbers than in the past."\textsuperscript{19} Other schools copied Tech’s model. Publicizing the Carnegie Plan was C.I.T.’s most successful marketing and recruiting tool by Doherty’s retirement in 1950.

The Carnegie Plan flourished during the early years of the Warner administration, in part because its principles were endorsed by the central administration and understood by the faculty. Five "Carnegie Educational Papers," which dealt with ways to integrate course aims across the curriculum and improve teaching methods, were published between 1955 and 1957 alone. "I doubt that one could find anywhere a college where so much is being done by faculty to develop the kind of teaching which will accomplish the institution’s objective with the students," the president observed midway through his term.\textsuperscript{20} Smith provided continuity in setting standards for teaching and learning and attained new heights in rhetorical flourishes. Only a balanced approach to education, the provost wrote in 1958, nurtured "profound originality" among Tech’s community of scholars: "If you have depth alone it will tend to detach you from reality. If you have breadth alone it will tend to make you shallow or timeserving, or dissipate your energies. Depth, breadth and integrity must all be present and must interplay if your image of stature is to be truly professional and a fit teacher."\textsuperscript{21}
The Lull

Mindful of its growing reputation as an innovative place, C.I.T. continued to seek out new applications of the Carnegie Plan. With support of the Westinghouse Educational Foundation, C.I.T. faculty offered summer workshops based on the problem-solving approach to upgrade the competence and teaching ability of local high-school science teachers. A Curriculum Study Center was established at Tech. Because of their earlier research on ways to teach effectively, Erwin Steinberg, Edwin Fenton and other curriculum associates were selected by the U.S. Office of Education to launch Project English and Project Social Studies as part of a nationwide effort to improve secondary education. The English department also played an important role in developing the College Board's Advanced Placement tests; the history department won a grant from the Carnegie Corporation to sponsor summer institutes for history teachers in predominantly Black colleges. Such spin-offs of the Carnegie Plan demonstrably strengthened high-school education and addressed some of the needs of minorities, but they did not improve Tech's own undergraduate curriculum.

Indeed, the central vision and centripetal modus operandi of the institution became attenuated after Smith retired. Some faculty openly complained about having to "fit" their courses and "adapt" their teaching styles to the principles of the Carnegie Plan. Research into pedagogical methodologies continued, but it produced less impressive (or at least less newsworthy and lasting) results. A growing dichotomy between research and teaching unfortunately began to distort C.I.T.'s overall educational mission. As Carnegie Tech placed more and more attention on post-baccalaureate initiatives, educational reforms at the undergraduate level inevitably sputtered. In his 1959–60 report, for instance, President Warner continued to extoll the importance of "cross-disciplinary" efforts, but he indicated that he would direct his attention "first toward graduate education and research, then to further interdisciplinary emphasis in our undergraduate educational programs." There was a discernible decline in references to the Carnegie Plan in subsequent annual reports and public-relations releases.

Despite the lull, important initiatives took place in the 1960s and 1970s that were to bear fruit later on—and which might become the basis for an even more daring revolution in higher education. Important breakthroughs in research into human problem-solving took place on this campus. At the heart of the program was a concern for information processing. Blazing new frontiers of knowledge in cognition and what would later be called "artificial intelligence" engaged some of Tech's brightest minds and attracted huge sums of outside funding to develop applications for computer technology. More than 300 people attended a 1965 conference on "problem solving," the first of an annual series of symposia in the area of
The acclaim accorded major texts that demonstrated the need for and intellectual excitement of sophisticated interdisciplinary research — notably Cyert and March's *Behavioral Theory of the Firm* (1963) and Newell and Simon's *Human Problem Solving* (1972) — underscored that Carnegie Tech was becoming a major center for research in cybernetics and unorthodox approaches to formal logic.

While some pursued pathbreaking research into theories of cognition and problem-solving, others endeavored to stimulate cross-disciplinary teaching. Among other things, the Curriculum Study Center invited a psychologist and four members of the English department to develop an engineering track suitable for students concentrating increasingly moribund "social relations" track. The National Science Foundation supported educational projects: one venture was intended to give superior undergraduates an opportunity to work closely on faculty mentors' current research; another facilitated the development of programmed instruction in basic courses. Even more boldly, key members of the administration contemplated an approach to problem-solving that went beyond the principles enunciated in the Carnegie Plan. In his 1962–63 report, Richard B. Teare, jr., dean of the Engineering and Science college, wrote:

> It is clear, I think, that what distinguishes the engineer from scientist, apart from education, is that while the scientist's goal is to add knowledge, discover new phenomena, establish new principles, the engineer is concerned with devising and designing systems which not only work but which meet often conflicting criteria of cost, efficiency, life, manufacturability, reliability, safety and so forth....Courses will be needed that are different in content and point of view than those we have now. And just as Carnegie Tech has been a leader in the developments in engineering education that took place in the last two decades, we want to continue to lead in the future.”

In the short-run, the dean's proposal would lead to the creation of a 3-course sequence on analysis, synthesis and evaluation (A/S/E) in the engineering college. From the perspective of the 1980s, it seems likely that Teare's idea foreshadowed a redefinition of Tech's "liberal/professional" tradition of general education for engineers by stretching its parameters under the banner of design. But not just any "design" course that employed problem-solving would do. A prototype developed at U.C.L.A., for instance, was critically evaluated and found wanting. The administration and faculty would have to create a *sui generis* approach that built on Tech's strengths and traditions.

Before these promising research and curricular agendas could be conjoined to create a fresh educational mission, however, the institution itself was to undergo a profound reorganization. Horton Guyford Stever appears to have had remarkably little personal impact on this place, but during his seven-year presidency, Carnegie Tech and Mellon Institute merged...
to form Carnegie-Mellon University, the College of Engineering and Science divided into two separate colleges, and the School of Urban and Public Affairs was created. In addition, eight members of the faculty developed an innovative design for the newly established College of Humanities and Social Sciences at a time in which the free-elective approach was in vogue in higher education: their *Building from Strength* proposed an elaborate set of course requirements during the first two years (and two mandatory upper-level classes in ethics and the twentieth century) as well as "meta-curricular" activities. Like the other undergraduate colleges in the university, the curriculum proposed for H&SS stressed a coherent search for patterns and design. Unlike the rest of CMU, however, teaching and a concern for teaching still took precedence over developing the college into a major research center.

These initiatives are interesting for two reasons. On the one hand, it is clear that major scholars at Tech were trying to formulate theories of cognition, learning and design in disciplinary-specific contexts as well as through cross-disciplinary research. Psychologists, engineers, scientists, music professors, and English teachers were working independently (and often together) to construct a model of a learner: they reconsidered the strategies implicit in teaching problem solving and formal operations and they investigated the cognitive processes presumed to be operating in computer-assisted instruction and in representing music. On the other hand, for all of their indigenous flavor, local efforts to blaze new frontiers in research and teaching were attuned (even if sometimes deliberately in opposition) to major developments in the academy. The Carnegie Plan may have gone underground after its heyday in the late 1940s, but its spirit and dynamic calculus were not obsolescent.

Cyert's Innovative Presidency

Although it is far too early to assess the significance of developments at Carnegie Mellon since 1972, Richard M. Cyert does resemble a latter-day Doherty. Like his predecessor, his vision and style of management build consistently on a few principles that became his trademark early in his professional development. Cyert has long emphasized the need to articulate and implement a strategy for innovation that capitalizes on CMU's size and comparative advantages. As a professor, dean and president, he has challenged this intellectual community to develop and improve upon curricular designs that teach students how to utilize information. Like Doherty, he acknowledged deficiencies in existing programs but he has a clear sense of what is possible:

Traditionally in education we have concentrated on the bodies of knowledge in our professional training...One of the reasons for our concentration on the disciplines that should be taught is that we are knowledgeable about that aspect of education. We impart information quite effectively. We are much less effective in teaching individuals ways of utilizing this information for the solution of professional problems...To improve education we
need a well-developed methodology of problem-solving behavior that would be general—that would be useful for students in all disciplines and professions.30

That Doherty characterized the educational challenge of the 1930s in similar words reminds us that each generation of reformers must confront anew perennial questions of pedagogy.

As a master of organizational behavior, Cyert devoted the first years of his presidency getting his institution into shape. Buildings were repaired. There was a resurgence in athletics. After years of operating in the red, the books were balanced: from Warner Hall, Cyert made sure that the endowment grew; in the labs and faculty lounges, the message was clear—it is fine to think great thoughts; it is even better to get someone else to pay your salary as you pursue them. In his tireless pursuit of excellence, the president has sought to maintain a balance between research and teaching. He tried to encourage an entrepreneurial spirit without eroding collegiality. But the advantages of running a university as a federated academic community, Cyert keenly recognized, had its risks: "Perhaps the most difficult organization to change in society is the university."31 So much energy was spent on nurturing the creative energies of a diverse faculty and on training an increasingly sophisticated student body that it was hard to attend to possible distortions in institutional priorities and to consider alternative approaches to general education.

During the Cyert years, each undergraduate college nurtured its own version of the "liberal/professional" tradition in general education. The stated objectives of CIT's curriculum paraphrase the goals of the original Carnegie Plan: the ability to learn independently with scholarly precision; a philosophic outlook, breadth of knowledge and sense of values; the ability to communicate ideas to others; skill in quantitative analysis; and, of course, a thorough and integrated understanding of the student's major field.32 After their first year, engineering undergraduates are required to take several courses each semester outside of C.I.T. and to follow an in-depth sequence of four courses in either the humanities, social sciences, or fine arts. Currently, the college's Department of Engineering and Public Policy is developing a set of interdisciplinary modules for freshman that will encourage them to hone their problem-solving capabilities by wrestling with real-world policy issues.

The core program in the Mellon College of Science (MCS) closely resembles the CIT curriculum. "To provide science education based on a strong grounding in fundamental knowledge complemented by up-to-date expertise," MCS requires its students to take courses in other colleges.33 Undergraduates are permitted but not required to take an in-depth sequence
of courses. During the 1984–85 year, moreover, arrangements were made so that students in MCS could declare a minor in the College of Humanities and Social Sciences, and vice versa.

The number of classes that undergraduates in the College of Fine Arts, always CMU's most intensely "professional" school, must take beyond their discipline and outside of CFA has increased over time. CFA undergraduates are permitted to minor in seven different programs developed by H&SS departments; it is not all that unusual to find a Drama student specializing in theatre lighting who takes a course in electrical engineering. CFA's curriculum has also become more directly linked to the rest of campus: the college's five departments now offer seven minors for non-CFA students; the Center for Art and Technology is avowedly cross-disciplinary in staffing. The computer, moreover, has become an invaluable educational tool. Not only have its graphics capabilities eliminated much of the drudgery associated with creating visual representations, but faculty and students grappling with fundamental problems of "design" are discovering parallels and differences in the way they go about their business compared to strategies adopted by scientists based in CIT's Design Research Center.

The core curriculum in the College of Humanities and Social Sciences (H&SS) adopted in 1977 built on the legacy of the Carnegie Plan: "While there are alternate strategies for achieving both liberal and professional educations, there are substantial overlaps, too. A sensible and defensible strategy, then, for achieving traditional, liberal educational objectives is to use professional objectives and problems to define the appropriate set of intellectual skills to be developed in a person's education; to use 'professional' problems, broadly defined to answer the question of 'what are the fundamentals?'" Dean John P. Crecine and his faculty created five clusters of courses that were to provide students with a common, integrative educational experience and to show them how to use transferable intellectual and conceptual skills. After a decade of refinement, the H&SS curriculum has become a model worth emulating—according to Edward B. Fiske, educational editor of the New York Times, it represents "perhaps the most original thinking of any American university in pursuing the twin goals of liberal professional education."

Ironically, however, the very fact that each college independently refined its general-education requirements to meet the specific needs of a subset of undergraduates has meant that the sum of the curricular parts did not inevitably add up to a coherent whole at CMU. Grassroots initiatives are indispensable, of course, but in our free-wheeling environment, they rarely have been coordinated by the central administration—or closely monitored by deans. During Cyert's presidency, moreover, the number of departments and research institutes grew sharply as the frontiers of knowledge broadened and the computer revolution created new
opportunities for CMU's faculty.

The resulting fragmentation of student interests and specialization of academic pursuits required a countervailing force that would promote greater awareness of this place as a university. By the early 1980s, the president and key members of his administration and faculty determined that the time had come to revamp the Carnegie Plan to address new challenges and opportunities. "Effective professional education calls for attention to both subject-matter knowledge and general skills." Developing a "core" was thus part of a larger strategy to help various constituencies at CMU see the common conceptual concerns and grounding in methods that link disparate intellectual enterprises and autonomous communities of discourse.

Efforts to implement a university-wide core began in earnest in 1984. The College Councils endorsed in principle new University Requirements for General Education by legitimating a de facto core that consisted of more than two dozen courses presently offered across campus that were already being taken by a large number of undergraduates. I assumed a part-time position in the central administration to oversee the establishment of a writing- across-the-curriculum program that built on the English department's expertise in teaching people how to think about writing and the introduction of a computing skills workshop to socialize students and staff to our rich environment. The Student Senate canvassed CMU undergraduates for their ideas. With more than a million dollars of support from the Pew and Mellon foundations, faculty in all four colleges and the School of Urban and Public Affairs developed prototypes for new interdisciplinary courses. Edwin Fenton and his associates in the University Teaching Center as well as members of the Center for the Design of Educational Computing (CDEC) assisted faculty in developing instructional materials and software.

The Place of "Design" in the New Core

Even if we implement no other changes in undergraduate education, we will have succeeded in reinforcing some of the interconnections among the liberal arts, science and technology. And yet, I think we can do more. CMU provides an ideal laboratory if we choose to think seriously about the meaning, structure, dynamics, and limits of design. As Herbert Simon notes, the ideas and techniques central to design are a key ingredient in each of our undergraduate liberal/professional curricula:

Engineers are not the only professional designers. Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for the state. Design, so construed, is the core of all
"Design" is the 1980s analog to Doherty's "learning-by-doing" foundation for an education that is represented in productions rather than propositions. If we decide to update the Carnegie Plan, we have intellectual and technological tools unavailable in the 1940s. Our computing environment facilitates broadly interdisciplinary and pedagogically daring ventures. ANDREW can complement faculty input by serving as a "coach": it offers personalized assistance in organizing and utilizing information, provides a programming capacity that eliminates much of the tedium of earlier problem-solving exercises, and creates a cognitive simulation environment that enables students to visualize the big picture. Furthermore, learning about "design" will enable our students to take greater advantage of resources than might otherwise be the case. Members of the Computer Science department, SEI, design centers in CIT and CFA, and faculty who have not spent much time teaching undergraduates are intrigued by the notion of making design a crucial element of general education at CMU.

Despite the appeal of "design" as an integrative concept to undergird the new university core, finding common ground is bound to be easier in principle than in actual practice. In part, the problems are epistemological. It is striking, for instance, that there is no word for "design" in Japanese, Arabic or French; in each instance, these languages have adopted a transliteration of the English word. Nor is there much in the history of American education to sustain the proposition that design might serve as the catalyst for reforming a curriculum. "It is the participation by the practical man in the theory, through the agency of the linking science," John Dewey asserted in his 1899 presidential address to the American Psychological Association, "that determines at once the effectiveness of the work done." It is reasonable to hypothesize that what we are calling a science of design is what Dewey meant by a "linking science." Still, throughout most our past, the term has referred primarily to the interaction between industrial technology and culture that gave form to an American "ethic" in material products, not to a pedagogical principle.

Definitions of design at CMU, moreover, tend to vary from college to college and to depend on a faculty member's professional predilections. In the College of Fine Arts, which at one point was called the school of applied design, artists, architects, musicians and dramatists describe the process of creating sequential patterns of representations as design. What is esteemed most highly is the experiential nature of dialectic process of valued inquiry and esthetic problem-solving. Engineers, in contrast, do not share this concern with visual gratification. They are more preoccupied with the specification of objectives and constraints,
the decomposition of materials, and final results: does a design work?

This manifest divergence of opinion about the meaning of design underscores the fact that necessary theoretical components are missing. Researchers do not understand much about acquiring new perceptual patterns. The most effective ways of representing the process of design still elude us. Gaps in theory, in turn, create problems in the classroom. Jill Larkin has enumerated several difficulties: "(1) problem solving is intrinsically very hard to teach, and this is particularly true in the areas of mathematics and science; (2) educational research has traditionally not used methodologies productive in providing information about problem-solving processes; and (3) although some individuals have produced instruction that seems to be effective in helping students to solve problems, very little is known about how this instruction works."45

In the face of such obstacles, there is bound to be considerable resistance from faculty and students alike to any proposal to introduce "design" into the curriculum. Faculty are most effective as instructors when they are intellectually engaged in the subject matter they are teaching. Some will claim that "design" is not what they do—or at least not what they really want to do. Even those with a stake in the venture will worry about trivializing problems of design in order to produce heuristic models suitable for classroom adoption. Furthermore, an interdisciplinary approach to teaching design invites an educational conundrum. Encouraging students to grapple with problems outside their specialty fosters intellectual breadth, but this tack also will quickly expose their lack of knowledge data in an unfamiliar problem domain. Nor can time constraints be dismissed lightly. In private conversations and surveys, students complain bitterly about curriculum overload; they resent any additional requirements that they perceive might distract them from getting on with the business of completing their major and getting a good job.46

To introduce the principle of "design" into the core curriculum, therefore, requires an appeal to the self interest of all constituencies. Before we press on, the case for design must be made on intellectual grounds. Students and faculty must be persuaded that this venture is consistent with the broad aims of a learning-by-doing approach to general education. As this paper indicates, I believe that the best way to proceed is to demonstrate that the science of design builds on the tradition of the Carnegie Plan: by making its mental processes explicit and by capitalizing on the computer's programming capabilities, we can teach students how to design their own designs for life-long learning.

In practical terms, this entails a two-fold educational strategy. First, I would insinuate an
appreciation for the logic and architecture of "design" into existing courses that already are part of the college-specific and university-wide curricula. Second, I would allocate the resources and marshall the talent necessary to develop discipline-specific, computer-aided instructional modules that might serve as the basis for an upper-level requirement to be satisfied by juniors or seniors.

********************

What is the place of design in CMU's general-education curriculum? To answer the question properly, I submit, is to come face to face with the same issues and challenges that impelled Robert Doherty to come to Tech in order to revolutionize higher education. Paradoxically, at a school that has long been oriented toward the future, it will be institutionally difficult for members of this community to appreciate how deeply the risks and potential success of this venture are rooted in our past.

Teaching design will be a subversive activity. To the extent that pursuing this venture upsets the current equilibrium, commitments within the central administration and at the grassroots level are indispensable. Failure is assured if we lose sight of basic educational principles thrives in our entrepreneurial, federated environment. Success hinges on our ability to remain faithful to our tradition as innovators inspired by the power of fresh ideas and unconventional educational objectives.
NOTES


8 ibid., p. 10


12 Doherty acknowledged as much in his 1936-7 report; see esp. pp. 11-13.
13 Quoted in Cleeton, p. 115.


16 For the complete text of the first official version of the "Carnegie Plan," see the Appendix, *infra*. Professor B. von H. Gilmer of the psychology department is typically given credit for coining the name; Smith used it in his 1948-49 annual report.


31*ibid.*, p. 7.

32Four of these goals alter only slightly the wording of the 1945 Social Relations plan.

33*Undergraduate Catalog, 1984-86*, p. 218. See also, Robert F. Sekerka, "On Liberal/Professional Education of, and by, Engineers and Scientists" prepared for the Board of Trustees "Conference on Interdisciplinary Education and Research at CMU," March 11-13, 1983.

34*ibid.*, p. 64.


38Cyert reconstituted an all-university educational committee in 1980 to lay the groundwork for curricular reform. Important working papers include a series of essays written by Tung Au, who chaired the Faculty Senate, for *Focus* in 1981 and 1982 and John P. Crecine's "Liberal and Professional Education" (1983).


44 That CFA faculty have views about design not shared elsewhere was quite apparent at a “faculty seminar on design” held in January 1986; it also is evident in the essays in this volume.


46 See, for instance, the views expressed in the Tartan, November 20, 1984, p. 6; and "Distribution requirements would make core work," April 8, 1986, p. 2.
2. THE SCIENCE OF DESIGN

THE SCIENCE OF DESIGN

Creating the Artificial

Historically and traditionally, it has been the task of the science disciplines to teach about natural things: how they are and how they work. It has been the task of engineering schools to teach about artificial things: how to make artifacts that have desired properties and how to design.

Engineers are not the only professional designers. Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences. Schools of engineering, as well as schools of architecture, business, education, law, and medicine, are all centrally concerned with the process of design.

In view of the key role of design in professional activity, it is ironic that in this century the natural sciences have almost driven the sciences of the artificial from professional school curricula. Engineering schools have become schools of biological science; business school have become schools of finite mathematics. The use of adjectives like "applied" conceals, but does not change, the fact. It simply means that in the professional schools those topics are selected from mathematics and the natural sciences for emphasis which are thought to be most nearly relevant to professional practice. It does not mean that design is taught, as distinguished from analysis.

The movement toward natural science and away from the sciences of the artificial has proceeded further and faster in engineering, business, and medicine than in the other professional fields I have mentioned, though it has but no means been absent from schools of law, journalism, and library science. The stronger universities are more deeply affected than the weaker, and the graduate programs more than the undergraduate. Few doctoral dissertations in first-rate professional schools today deal with genuine design problems, as distinguished from problems in solid-state physics or stochastic processes. I have to make partial exceptions—for reasons I shall mention—for dissertations in computer science and management science, and there are undoubtedly some others, for example, in chemical engineering.
Such a universal phenomenon must have a basic cause. It does have a very obvious one. As professional schools, including the independent engineering schools, are more and more absorbed into the general culture of the university, they hanker after academic respectability calls for subject matter that is intellectually tough, analytic, formalizable, and teachable. In the past much, if not most, of what we knew about design and about the artificial sciences was intellectually soft, intuitive, informal, and cookbooky. Why would anyone in a university stoop to teach or learn about designing machines or planning market strategies when he could concern himself with solid-state physics? The answer has been clear: he usually wouldn't.

The problem is widely recognized in engineering and medicine today and to a lesser extent in business. Some do not think it a problem, because they regrad schools of applied science as a superior alternative to the trade schools of the past. If that were the choice, we could agree. But neither alternative is satisfactory. The older kind of professional school did not know how to educate for professional design at an intellectual level appropriate to a university; the newer kind of school has nearly abdicated responsibility for training in the core professional skill. Thus we are faced with a problem of devising a professional school that can attain two objectives simultaneously: education in both artificial and natural science at a high intellectual level. This too is a problem of design — organizational design.

The kernel of the problem lies in the phrase "artificial science." In my previous chapters I have shown that a science of artificial phenomena is always in imminent danger of dissolving and vanishing. The peculiar properties of the artifact lie on the thin interface between the natural law within it and the natural laws without. What can we say about it? What is there to study besides the boundary sciences — those that govern the means and the task environment?

The artificial world is centered precisely on this interface between the inner and outer environments; it is concerned with attaining goals by adapting the former to the latter. The proper study of those who are concerned with the artificial is the way in which that adaption of means to environments is brought about — and central to that is the process of design itself. The professional schools will reassume their professional responsibilities just to the degree that they can discover a science of design, a body of intellectually tough, analytic, partly formalizable, partly empirical, teachable doctrine about the design process.

It is the thesis of this chapter that such a science of design not only is possible but is actually emerging at the present time. It has already begun to penetrate the engineering schools, particularly through programs in computer science and "systems engineering," and business schools through management science. Perhaps it also has beach-heads in other
professional curricula, but these are the two with which I am most familiar. We can already see enough of its shape to predict some of the important ways in which engineering schools tomorrow will differ from departments of physics, and business schools from departments of economics and psychology. Let me now turn from questions of university organization to the substance of the matter.

The Login of Design: Fixed Alternatives

We must start with some questions of logic. The natural sciences are concerned with how things are. Ordinary systems of logic—the standard propositional and predicate calculi, say—serve these sciences well. Since the concern of standard logic is with declarative statements, it is well suited for assertions about the world and for inferences from those assertions.

Design, on the other hand, is concerned with how things ought to be, with devising artifacts to attain goals. We might question whether the forms of reasoning that are appropriate to natural science are suitable also for design. One might well suppose that introduction of the verb "should" may require additional rules of inference, or modification of the rules already imbedded in declarative logic.

Paradoxes of Imperative Logic

Various "paradoxes" have been constructed to demonstrate the need for a distinct logic of imperatives, or a normative, deontic logic. In ordinary logic from "Dogs are pets" and "Cats are pets," one can infer "Dogs and cats are pets." But from "Dogs are pets," "Cats are pets," and "You should keep pets," can one infer "You should keep cats and dogs"? And from "Give me needle and thread!" can deduce, in analogy with declarative logic, "Give me needle or thread"? Easily frustrated people would perhaps rather have neither needle nor thread than one without the other, and peaceloving people, neither cats nor dogs, rather than both.

As a response to these challenges of apparent paradox, there have been developed a number of constructions of modal logic for handling "shoulds," "shalts," and "oughts" of various kinds. I think it is fair to say that none of these systems has been sufficiently developed or sufficiently widely applied to demonstrate that it is adequate to handle the logical requirements of the process of design.

Fortunately, such a demonstration is really not essential, for it can be shown that the requirements of design can be met fully by a modest adaption of ordinary declarative logic. Thus a special logic of imperatives is unnecessary.
Reduction to Declarative Logic

The easiest way to discover what kinds of logic are needed for design is to examine what kinds of logic designers use when they are being careful about their reasoning. Now there would be no point in doing this if designers were always sloppy fellows who reasoned loosely, vaguely, and intuitively. Then we might say that whatever logic they used was not the logic they should use.

However, there exists a considerable area of design practice where standards of rigor in inference are as high as one could wish. I refer to the domain of so-called "optimization methods," most highly developed in statistical decision theory and management science but acquiring growing importance also in engineering design theory. The theories of probability and utility, and their intersection, have received the painstaking attention not only of practical designers and decision makers but also of a considerable number of the most distinguished logicians and mathematicians of the present and recent past generations. F. P. Ramsey, B. de Finetti, A. Wald, J. von Neumann, J. Neyman, K. Arrow, and L. J. Savage are examples.

The logic of optimization methods can be sketched as follows: The "inner environment" of the design problem is represented by a set of given alternatives of action. The alternatives may be given in entenscr: more commonly they are specified in terms of command variables that have defined domains. The "outer environment" is represented by a set of parameters, which may be known with certainty or only in terms of a probability distribution. The goals for adaption of inner to outer environment are defined by a utility function—a function, usually scalar, of the command variables and environmental parameters—perhaps supplemented by a number of constraints (inequalities, say between functions of the command variables and environmental parameters). The optimization problem is to find an admissible set of values of the command variables, compatible with the constraints, that maximize the utility function for the probabilistic case we might say, "maximize the expected value of the utility function," for instance, instead of "maximize the utility function."

A stock application of this paradigm is the so-called "diet problem" shown in figure 6. A list of foods is provided, the command variables being quantities of the various foods to be included in the diet. The environmental parameters are the prices and nutritional contents (calories, vitamins, minerals, and so on) of each of the foods. The utility function is the cost (with a minus sign attached) of the diet, subject to the constraints, say, that it not contain more than 2,000 calories per day, that it meet specified minimum needs for vitamins and minerals, and that rutabaga not be eaten more than once a week. The constraints may be viewed as characterizing the inner environment. The problem is to select the quantities of
foods that will meet the nutritional requirements and side conditions at the given prices for the lowest cost.

The diet problem is a simple example of a class of problems that are readily handled, even when the number of variables is exceedingly large, by the mathematical formalism known as linear programming. I shall come back to the technique a little later. My present concern with the logic of the matter.

Since the optimization problem, one formalized, is a standard mathematical problem—to maximize a function subject to constraints—it is evident that the logic used to deduce the answer is the standard logic of the predicate calculus on which mathematics rests. How does the formalism avoid making use of a special logic of imperatives? It does so by dealing with set of possible worlds: first consider all the possible worlds that meet the constraints of the outer environment; then find the particular world in the set that meets the remaining constraints of the goal and maximizes the utility function. The logic is exactly the same as if we were to adjoin goal constraints and the maximization requirement, as new "natural laws," to the existing natural laws embodied in the environmental conditions. We simply ask what values the command variables would have in a world meeting all these conditions and conclude that these are the values the command variable should have.

Computing the Optimum

Our discussion thus far has already provided us with two central topics for the curriculum in the science of design:

1. Utility theory and statistical decision theory as a logical framework for rational choice among given alternative.

2. The body of techniques for actually deducing which of the available alternatives is the optimum.

Only in trivial cases is the computation of the optimum alternative an easy matter. If utility theory is to have application to real-life design problems, it must be accompanied by tools for actually making the computations. The dilemma of the rational chess player is familiar to all. The optimal strategy is chess is easily demonstrated: simply assign a value of +1 to a win, 0 to a draw, -1 to a loss; consider all possible courses of play; minimax backward from the outcome of each, assuming each player will take the most favorable move at any given point. This procedure will determine what move to make now. The only trouble is that the computation required are astronomical (the number 10 (120) is often mentioned in this context) and hence cannot be carried out—not by humans, not by existing computers, not by prospective computers.
A theory of design as applied to the game of chess would encompass not only the utopian minimax principle but also some practicable procedures for finding good moves in actual board positions in real time, within the computational capacities of real human beings or real computers. No exceptionally good procedures of this kind exist today, other than those stored in the memories of grandmasters, but there is at least one computer program that plays at the level of an expert of a weak master—that is, better than all save a few hundred human players.

The second topic then for the curriculum in the science of design consists in the efficient computational techniques that are available for actually finding optimum courses of action in real situations, or reasonable approximations to real situations. As I mentioned in chapter 2, that topic has a number of important components today, most of them developed—at least to the level of practical application—within the past 25 years. These include linear programming theory, dynamic programming, geometric programming, queing theory, and control theory.

Finding Satisfactory Actions

The subject of computational techniques need not be limited to optimization. Traditional engineering design methods make much more use of inequalities—specifications of satisfactory performance—than of maxima and minima. So-called "figures of merit" permit comparison between designs in terms of "better" and "worse" but seldom provide a judgement of "best." For example, I may cite the root-locus methods employed in the design of servomechanisms.

Since there did not seem to be any word in English for decision methods that look for good or satisfactory solutions instead of optimal ones, some years ago I introduced the term "satisficing" to refer to such procedures. Now no one in his right mind will satisfice if he can equally well optimize; no one will settle for good or better if he can have best. But that is not the way the problem usually poses itself in actual design situations.

In chapter 2 I argued that in the real world we usually do not have a choice between satisfactory and optimal solutions, for we only rarely have a method of finding the optimum. Consider, for example, the well-known combinatorial problem called the traveling salesman problem: given the geographical locations of a set of cities, find the routing that will take a salesman to all the cities with the shortest mileage. For this problem there is a straightforward optimizing algorithm (analogous to the minimax algorithm for chess): try all possible routings, and pick the shortest. But for any considerable number of cities, the algorithm is computationally infeasible (the number of routes through N cities will be N!). Although some ways have been found for cutting down the length of the search, no algorithm has been discovered sufficiently powerful to solve the traveling salesman problem with a
tolerable amount of computing for a set of, say, fifty cities.

Rather than keep out salesman at home, we shall prefer of course to find a satisfactory, if not optimal, routing for him. Under most circumstances, common sense will probably arrive at a fairly good route, but an even better one can often be found by one or another of several heuristic methods.

An earmark of all these situations where we satisfice for inability to optimize is that, although the set of available alternatives is "given" in a certain abstract sense (we can define a generator guaranteed to generate all of them eventually), it is not "given" in the only sense that is practically relevant. We cannot within practicable computational limits generate all the admissible alternatives and compare their respective merits. Nor can we recognize the best alternative, even if we are fortunate enough to generate it early, until we have seen all of them. We satisfice by looking for alternatives in such a way that we can generally find an acceptable one after only moderate search.

Now in many satisficing situations, the expected length of search for an alternative meeting specified standards of acceptability depends on how high the standards are set, but it depends hardly at all on the total size of the universe to be searched. The time required for a search through a haystack for a needle sharp enough to sew with depends on the density of distribution of sharp needles but not on the total size of the stack.

Hence, when we use satisficing methods, it often does not matter whether or not the total set of admissible alternatives is "given" by a formal but impracticable algorithm. It often does not even matter how big that set is. For this reason satisficing methods may be extendable to design problems in that broad range where the set of alternatives is not "given" even in the quixotic sense that it is "given" for the traveling salesman problem. Our next task is to examine this possibility.

THE LOGIC OF DESIGN: FINDING ALTERNATIVES

When we take up the case where the design alternatives are not given in any constructive sense but must be synthesized, we must ask once more whether any new forms of reasoning are involved in the synthesis, or whether again the standard logic of declarative statements is all we need.

In the case of optimization we asked: "Of all possible worlds (those attainable for some admissible values of the action variables), which is the best (yields the highest value of the criterion function)?" As we saw, this is a purely empirical question, calling only for facts and
In this case, where we are seeking a satisfactory alternative, once we have found a candidate we can ask: "Does this alternative satisfy all the design criteria?" Clearly this is also a factual question and raises no new issues of logic. But how about the process of searching for candidates? What kind of logic is needed for the search?

Means-Ends Analysis

The condition of any goal-seeking system is that it is connected to the outside environment through two kinds of channels: the afferent, or sensory, channels through which it receives information about the environment and the efferent, or motor, channels through which it acts on the environment\textsuperscript{52} The system must have some means of storing in its memory information about states of the world—afferent, or sensory, information—and information about actions—efferent, or motor, information. Ability to attain goals depends on building up associations, which may be simple or very complex, between particular changes in states of the world and particular actions that will (reliably or not) bring these changes about. In chapter 4 we described these associations as productions.

Except for a few built-in reflexes, as infant has no basis for correlating his sensory information with his actions. A very important part of his early learning is that particular actions or sequences of actions will bring about particular changes in the state of the world as he senses it. Until he builds up this knowledge, the world of sense and the motor world are two entirely separate, entirely unrelated worlds. Only as he begins to acquire experience as to how elements of the one relate to elements of the other can he act purposefully on the world.

The computer problem-solving program called GPS, designed to model some of the main features of human problem solving, exhibits in stark form how goal-directed action depends on building this kind of bridge between the afferent and the efferent worlds. On the afferent, or sensory, side, GPS must be able to represent desired situations or desired objects as well as the present situation. It must be able also to represent differences between the desired and the present. On the efferent side, GPS must be able to represent actions that change objects or situations. To behave purposefully, GPS must be able to select from time to time those particular actions that are likely to remove the particular differences between desired and present states that the system detects. In the machinery of GPS, this selection is achieved through a table of connections, which associates with each kind of detectable difference those actions that are relevant to reducing that difference. These are its associations, in the form of productions, which relate the afferent to the efferent world. Since reaching a goal generally requires a sequence of actions, and since some attempts may be ineffective, GPS must also
have means for detecting the progress it is making (the changes in the differences between the actual and the desired) and for trying alternate paths.

The Logic of Search

GPS then is a system that searches selectively through a (possibly large) environment in order to discover and assemble sequences of actions that will lead it from a given situation to a desired situation. What are the rules of logic that govern such a search? Is anything more than standard logic involved? Do we require a modal logic to rationalize the process?

Standard logic would seem to suffice. To represent the relation between the afferent the efferent worlds, we conceive GPS as moving through a large maze. The nodes of the maze represent situations, described afferently; the paths joining one node to another are the actions, described as motor sequences, that will transform the one situation into the other. At any given moment GPS is always faced with a single question: "What action shall I try next?" Since GPS has some imperfect knowledge about the relations of actions to changes in the situation, this becomes a question of choice under uncertainty of a kind already discussed in a previous section.

It is characteristic of the search for alternatives that the solution, the complete action that constitutes the final design, is built from a sequence of component actions. The enormous size of the space of alternatives arises out of the innumerable ways in which the component actions, which need not be very numerous, can be combined into sequences.

Much is gained by considering the component actions in place of the sequences that constitute complete actions, because the situation when viewed afferently usually factors into components that match at least approximately the component actions derived from an efferent factorization. The reasoning implicit in GPS is that, if a desired situation differs from a present situation by difference $D_1, D_2, \ldots, D_n$, and if action $A_1$, removes differences of type $D_1$, action $A_2$ removes differences of type $D_2$, and so on, then the present situation can be transformed into the desired situation by performing the sequence of actions $A_2, A_2, \ldots, A_n$.

This reasoning is by no means valid in terms of the rules of standard logic in all possible worlds. Its validity requires some rather strong assumptions about the independence of the effects of the several actions on the several differences. One might say that the reasoning is valid in worlds that are "additive" or "factorable" in a certain sense. (The air of paradox about the cat–dog and needle–thread examples cited earlier arises precisely from the nonadditivity of the actions in these two cases. The first is, in economists' language, a case of decreasing returns; the second, a case of increasing returns.)
Now the real worlds to which problem solvers and designers address themselves are seldom completely additive in this sense. Actions have side consequences (may create new differences) and sometimes can only be taken when certain side conditions are satisfied (call for removal of other differences before they become applicable). Under these circumstances one can never be certain that a partial sequence of actions that accomplishes certain goals can be augmented to provide a solution that satisfies all the conditions and attains all the goals (even though they be satisficing goals) of the problem.

For this reason problem-solving systems and design procedures in the real world do not merely assemble problem solutions from components but must search for appropriate assemblies. In carrying out such a search, it is often efficient to divide one's eggs among a number of baskets—that is, not to follow out one line until it succeeds completely or fails definitely but to begin to explore several tentative paths, continuing to pursue a few that look most promising at a given moment. If one of the active paths begins to look less promising, it may be replaced by another that had previously been assigned a lower priority.

Our discussion of design when the alternatives are not given has yielded at least three additional topics for instruction in the science of design:

1. *Adaptation of standard logic to the search for alternatives.* Design solutions are sequences of actions that lead to possible worlds satisfying specified constraints. With satisficing goals the sought—for possible worlds are seldom unique; the search is for sufficient, not necessary, actions for attaining goals.

2. *The exploitation of parallel, or near-parallel, factorizations of differences.* Means-end analysis is an example of a broadly applicable problem-solving technique that exploits this factorization.

3. *The allocation of resources for search to alternative, partly explored action sequences.* I should like to elaborate somewhat on this last-mentioned topic.

**DESIGN AS RESOURCE ALLOCATION**

There are two ways in which design processes are concerned with the allocation of resources. First, conservation of scarce resources may be one of the criteria for a satisfactory design. Second, the design process itself involves management of the resources of the designer, so that his efforts will not be dissipated unnecessarily in following lines of inquiry that prove fruitless.

There is nothing special that needs to be said here about resource conservation—cost minimization, for example, as a design criterion. Cost minimization has always been an implicit consideration in the design of engineering structures, but until a few years ago it generally was only implicit, rather than explicit. More and more cost calculations have been
brought explicitly into the design procedure, and a strong case can be made today for training design engineers in that body of techniques and theory that economists know as "cost–benefit analysis."

An Example from Highway Design

The notion that the costs of designing must themselves be considered in guiding the design process is one that has been introduced much more recently and that has still not had wide application, except at an intuitive level. A good example of what I mean by this is the procedure, developed by Marvin L. Manheim as a doctoral thesis at MIT, for solving highway location problems.33

Manheim's procedure incorporates two main notions first, the idea of specifying a design progressively from the level of very general plans down to determining the actual construction; second, the idea of attaching values to plans at the higher levels as a basis for deciding which plans to pursue to levels of greater specificity.

In the case of highway design the higher-level search is directed toward discovering "bands of interest" within which the prospects of finding a good specific route are promising. Within each band of interest one or more locations is selected for closer examination. Specific designs are then developed for particular locations. The scheme is not limited of course to this specific three-level division, but it can be generalized as appropriate.

Manheim's scheme for deciding which alternatives to pursue from one level to the next is based on assigning costs to each of the design activities and estimating highway costs for each of the higher-level plans. The cost associated with a plan is a prediction of what the cost would be for the actual route if that plan were particularized through subsequent design activity. In other words, it is a measure of how "promising" a plan is. Those plans are then pursued to completion that look most promising after the prospective design costs have been offset against them. In the particular method that Manheim describes, the "promise" of a plan is represented by a probability distribution of outcomes that would ensue if it were pursued to completion. The distribution must be estimated by the engineer—a serious weakness of the method—but, once estimated, it can be used within framework of Bayesian decision theory. The particular probability model used is not the important thing about the method; other methods of valuation without the Bayesian superstructure might be just as satisfactory.

In the highway location procedure the evaluation of higher-level plans performs two functions. First, it answers the question, "Where shall I search next?" Second, it answers the question, "When shall I stop the search and accept a solution as satisfactory?" Thus it is both
a steering mechanism for the search and a satisficing criterion for terminating the search.

Schemes for Guiding Search

Let us generalize the notion of schemes for guiding search activity beyond Manheim's specific application to a highway location problem and beyond his specific guidance scheme based on Bayesian decision theory. Consider the typical structure of a problem-solving program. The program begins to search along possible paths, storing in memory a "tree" of the paths it has explored. Attached to the end of each branch—each partial path—is a number that is supposed to express the "value" of that path.

But the term "value" is really a misnomer. A partial path is not a solution of the problem, and a path has a "true" value of zero unless it leads toward a solution. Hence it is more useful to think of the values as estimates of the gain to be expected from further search along the path than to think of them as "values" in any more direct sense. For example, it may be desirable to attach a relatively high value to a partial exploration that may lead to a very good solution but with a low probability. If the prospect fades on further exploration, only the cost of the search has been lost. The disappointing outcome need not be accepted, but an alternative path may be taken instead. Thus the scheme for attaching values to a partial paths may be quite different from the evaluation function for proposed complete solutions.

When we recognize that the purpose of assigning values to incomplete paths is to guide the choice of the next point for exploration, it is natural to generalize even further. All kinds of information gathered in the course of search may be of value in selecting the next step in search. We need not limit ourselves to valuations of partial search paths.

For example, in a chess-playing program an exploration may generate a continuation move different from any that was proposed by the initial move generator. Whatever the context—the branch of the search—tree on which the move was actually generated, it can now be removed from the context and considered in the context of other move sequences. Such a scheme was added on a limited basis by Baylor to MATER, a program for discovering check-mating combinations in chess, and it proved to enhance the program's power significantly.

Thus search processes may be viewed—as they have been in most discussions of problem solving—as processes for seeking a problem solution. But they can be viewed more generally as processes for gathering information about problem structure that will ultimately be valuable in discovering a problem solution. The latter viewpoint is more general than the former in a significant sense, in that it suggests that information obtained along any particular branch of a
search tree may be used in many contexts besides the one in which it was generated. Only a
few problem-solving programs exist today that can be regarded as moving even a modest
distance from the earlier, more limited viewpoint to the newer one. Here is an important
direction for research in the theory of design.

THE SHAPE OF THE DESIGN: HIERARCHY

In my first chapter I gave some reasons why complex systems might be expected to be
constructed in a hierarchy of levels, or in a boxes-within-boxes form. The basic idea is that
the several components in any complex system will perform particular subfunctions that
contribute to the over-all function. Just as the "inner environment" of the whole system may
be defined by describing its functions, without detailed specification of its mechanisms, so the
"inner environment" of each of the subsystems may be defined by describing the functions of
that subsystem, without detailed specification of its submechanisms.56

To design such a complex structure, one powerful technique is to discover viable ways of
decomposing it into semi-independent components corresponding to its many functional parts.
The design of each component can then be carried out with some degree of independence of
the design of others, since each will affect the others largely through its function and
independently of the details of the mechanisms that accomplish the function.57

There is no reason to expect that the decomposition of the complete design into
functional components will be unique. In important instances there may exist alternative
feasible decompositions of radically different kinds. This possibility is well known to designers
of administrative organizations, where work can be divided up by subfunctions, by subprocesses,
by subareas, and in other ways. Much of classical organization theory in fact was concerned
precisely with this issue of alternative decompositions of a collection of interrelated tasks.

The Generator-Test Cycle

One way of considering the decomposition, but acknowledging that the interrelations
among the components cannot be ignored completely, is to think of the design process as
involving, first, the generation of alternatives and, then, the testing of these alternatives against
a whole array of requirements and constraints. There need not be merely a single generate-
test cycle, but there can be a whole nested series of such cycles. The generators implicitly
define the decomposition of the design problem, and the tests guarantee that important indirect
consequences will be noticed and weighed. Alternative decompositions correspond to different
ways of dividing the responsibilities for the final design between generators and tests.

To take a greatly oversimplified example, a series of generators may generate one or more
possible outlines and schemes of fenestration for a building, while tests may be applied to
determine whether needs for particular kinds of rooms can be met within the outlines
generated. Alternatively the generators may be used to evolve the structure of rooms, while
tests are applied to see whether they are consistent with an acceptable over-all shape and
design. The house can be designed from the outside in or from the inside out.58

Alternatives are also open, in organizing the design process, as to how far development of
possible subsystems will be carried before the over-all coordinating design is developed in
detail, or vice-versa, how far the over-all design should be carried before various components,
or possible components, are developed. These alternatives of design are familiar to architects.
They are familiar also to composers, who must decide how far the architectonics of a musical
structure will be evolved before some of the component musical themes and other elements
have been invented. Computer programmers face the same choices, between working downward
from executive routines to subroutines to a coordinating executive.

A theory of design will include principles—most of which do not yet exist—for deciding
such questions of precedence and sequence in the design process.

Process as a Determinant of Style

When we recall that the process will generally be concerned with finding a satisfactory
design, rather than an optimum design, we see that sequence and the division of labor between
generators and tests can affect not only the efficiency with which resources for designing are
used but also the nature of the final design as well. What we ordinarily call "style" may stem
just as much from these decisions about the design process as from alternative emphases on the
goals to be realized through the final design. An architect who designs buildings from the
outside in will arrive at quite different buildings from one who designs from the inside out,
even though both of them might agree on the characteristics that a satisfactory building should
possess.

When we come to the design of systems as complex as cities, or buildings, or economies,
we must give up the aim of creating systems that will optimize some hypothesized utility
function, and we must consider whether differences in style of the sort I have just been
describing do not represent highly desirable variants in the design process rather than
alternatives to be evaluated as "better" or "worse." Variety, within the limits of satisfactory
constraints, may be a desirable end in itself, among other reasons, because it permits us to
attach value to the search as well as its outcome—if we have wits to organize the process that
way. I shall have more to say on these topics in the next chapter.
However that may be, I hope I have illustrated sufficiently that both the shape of the design and the shape and organization of the design process are essential components of a theory of design. These topics constitute the sixth item in my proposed curriculum in design:

6. The organization of complex structures and its implication for the organization of design processes.

REPRESENTATION OF THE DESIGN

I have by no means surveyed all facets of the emerging science of design. In particular I have said little about the influence of problem representation on design. Although the importance of the question is recognized today, we have little systematic knowledge about it. I shall cite one example, to make clear what I mean by "representation."

Here are the rules of a game, which I shall call number scrabble. The game is played by two people with nine cards—let us say the ace through the nine of hearts. The cards are placed in a row, face up, between the two players. The players draw alternately, one at a time, selecting any one of the cards that remain in the center. The aim of the game is for a player to make up a "book," that is, a set of exactly three cards have been drawn without either player making a book, the game is a draw.

What is a good strategy in this game? How would you go about finding one? If the reader has not already discovered it for himself, let me show how a change in representation will make it easy to play the game well. The magic square here, which I introduced in the third chapter, is made up of the numerals from 1 through 9.

4 9 2
3 5 7
8 1 6

Each row, column, or diagonal adds to 15, and every triple of these numerals that add to 15 is a row, column, or diagonal of the magic square. From this, it is obvious that "making a book" in number scrabble is equivalent to getting "three in a row" in the game of tic-tac-toe. But most people know how to play tic-tac-toe well, hence can simply transfer their usual strategy to number scrabble.

Problem Solving as Change in Representation

That representation makes a difference is a long-familiar point. We all believe that arithmetic has become easier since Arabic numerals and place notation replaced Roman numerals, although I know of no theoretic treatment that explains why.
That representation makes a difference is evident for a different reason. All mathematics exhibits in its conclusions only what is already implicit in its premises, as I mentioned in a previous chapter. Hence all mathematical derivation can be viewed simply as change in representation, making evident what was previously true but obscure.

This view can be extended to all of problem solving—solving a problem simply means representing it so as to make the solution transparent. If the problem solving could actually be organized in these terms, the issue of representation would indeed become central. But even if it cannot—if this is too exaggerated a view—a deeper understanding of how representations are created and how they contribute to the solution of problems will become an essential component in the future theory of design.

Spatial Representation

Since much of design, particularly architectural and engineering design, is concerned with object or arrangements in real Euclidean two-dimensional or three-dimensional space, the representation of space and of things in space will necessarily be a central topic in a science of design. From our previous discussion of visual perception, it should be clear that "space" inside the head of the designer of the memory of a computer may have very different properties from a picture on paper or a three-dimensional model.

These representational issues have already attracted the attention of those concerned with computer-aided design—the cooperation of human and computer in the design process. As a single example, I may mention Ivan Sutherland's SKETCHPAD program, which allows geometric shapes to be represented and conditions to be placed on these shapes in terms of constraints, to which they then conform.

Geometric considerations are also prominent in the attempts to automate completely the design, say of printed or etched circuits, or of buildings. Grason, for example, in a system for designing house floor plans, constructs an internal representation of the layout that helps one decide whether a proposed set of connections among rooms, selected to meet design criteria for communication, and so on, can be realized in a plane.

The Taxonomy of Representation

An early step toward understanding any set of phenomena is to learn what kinds of things there are in the set—to develop a taxonomy. This step has not yet been taken with respect to representations. We have only a sketchy and incomplete knowledge of the different ways in which problems can be represented and much less knowledge of the significance of the differences.
In a completely pragmatic vein we know that problems can be described verbally, in natural language. They often can be described mathematically, using standard formalisms of algebra, geometry, set theory, analysis, or topology. If the problems relate to physical objects, they (or their solutions) can be represented by floor plans, engineering drawings, rendering, or three-dimensional models. Problems that have to do with actions can be attacked with flow charts and programs.

Other items most likely will need to added to the list, and there may exist more fundamental and significant ways of classifying its members. But even though our classification is incomplete and perhaps superficial, we can begin to build a theory of the properties of these representations. A number of topics in the growing theories of machines and of programming languages may give us some notion of the directions that a theory of representations—at least on its more formal side—may take. These topics can also provide, at the beginning, some of the substance for the final subject in our program on the theory of design:

7. Alternative presentations for design problems.

SUMMARY—TOPICS IN THE THEORY OF DESIGN

My main goal in this chapter has been to show that there already exist today a number of components of theory of design and a substantial body of knowledge, theoretical and empirical, relating to each. As we draw up our curriculum in design—in the science of the artificial—to take its place by the side of natural science in the whole engineering curriculum, it includes at least the following topics:

THE EVALUATION PROCESS

1. Theory of evaluation: utility theory, statistical decision theory

2. Computational methods:
   a. Algorithms for choosing *optimal* alternatives such as linear programming computations, control theory, dynamic programming
   b. Algorithms and heuristics for choosing *satisfactory* alternatives

3. THE FORMAL LOGIC OF DESIGN: imperative and declarative logics

4. THE SEARCH FOR ALTERNATIVES

5. Heuristic search: factorization and means–ends analysis

6. Allocation of resources for search
7. THEORY OF STRUCTURE AND DESIGN ORGANIZATION: hierarchic systems

8. REPRESENTATION OF DESIGN PROBLEMS

In small segments of the curriculum—the theory of evaluation, for example, and the formal logic of design—it is already possible to organize the instruction within a framework of systematic, formal theory. In many other segments the treatment would be more pragmatic, more empirical.

But nowhere do we need to return or retreat to the methods of the cookbook that originally put design into disrepute and drove it from the engineering curriculum. For there exist today a considerable number of examples of actual design processes, of many different kinds, that have been defined fully and cast in the metal, so to speak, in the form of running computer programs: optimizing algorithms, search procedures, and special-purpose programs for designing motors, balancing assembly lines, selecting investment portfolios, locating warehouses, designing highways, diagnosing and treating diseases, and so forth.

Because these computer programs describe complex design processes in complete, painstaking detail, they are open to full inspection and analysis, or to trial by simulation. They constitute a body of empirical phenomena to which the student of design can address himself and which he can seek to understand. There is no question, since these programs exist, of the design process hiding behind the cloak of "judgement" or "experience." Whatever judgement or experience was used in creating the programs must now be incorporated in them and hence be observable. The programs are the tangible record of the variety of schemes that man has devised to explore his complex outer environment and to discover in that environment the paths to his goals.

ROLE OF DESIGN IN THE LIFE OF THE MIND

I have called my topic "the theory of design" and my curriculum a "program in design." I have emphasized its role as complement to the natural science curriculum in the total training of a professional engineer—or of any professional whose task is to solve problems, to choose, to synthesize, to decide.

But there is another way in which the theory of design may be viewed in relation to other knowledge. My third and fourth chapters were chapters on psychology—specifically on man's relation to the complex outer environment in which he seeks to survive and achieve.

All three chapters, so construed, have import that goes beyond the professional work of the man we have called the "designer." Many of us have been unhappy about the
fragmentation of our society into two cultures. Some of us think there are not just two cultures but a large number of cultures. If we regret that fragmentation, then we must look for a common core of knowledge that can be shared by the members of all cultures—a core that includes more significant topics than the weather, sports, automobiles, the care and feeding of children, or perhaps even politics. A common understanding of our relation to the inner and outer environments that define the space in which we live and choose can provide at least part of that significant core.

This may seem an extravagant claim. Let me use the realm of music to illustrate what I mean. Music is one of the most ancient of the sciences of the artificial, and was so recognized by the Greeks. Anything I have said about the artificial would apply as well to music, its composition or its enjoyment, as to the engineering topics I have used for most of my illustrations.

Music involves a formal pattern. It has few (but important) contacts with the inner environment; that is, it is capable of evoking strong emotions, its patterns are detectable by human listeners, and some of its harmonic relations can be given physical and physiological interpretations (though the aesthetic import of these is debatable.) As for the outer environment, when we view composition as a problem in design, we encounter just the same tasks of evaluation, of search for alternatives, and of representation that we do in any other design problem. If it pleases us, we can even apply to music some of the same techniques of automatic design by computer that have been used in other fields of design. If computer-composed music has not yet reached notable heights of aesthetic excellence, it deserves, and has already received, serious attention from professional composers analysts, who do not find it written in tongues alien to them.65

Undoubtedly there are tone-deaf engineers, just as there are mathematically ignorant composers. Few engineers and composers, whether deaf, ignorant, or not, can carry on a mutually rewarding conversation about the content of each other's professional work. What I am suggesting is that they can carry on such a conversation about design, can begin to perceive the common creative activity in which they are both engaged, can begin to share their experiences of the creative, professional design process.

Those of us who have lived close to the development of the modern computer through gestation and infancy have been drawn from a wide variety of professional fields, music being one of them. We have noticed the growing communication among intellectual disciplines that takes place around the computer. We have welcomed it, because it has brought us into contact
with new worlds of knowledge—has helped us combat our own multiple-cultures isolation. This breakdown of old disciplinary boundaries has been much commented upon, and its connection with computers and the information sciences often noted.

But surely the computer, as a piece of hardware, or even as a piece of programmed software, has nothing to do directly with the matter. I have already suggested a different explanation. The ability to communicate across field—the common ground—comes from the fact that all who use computers in complex ways are using computers to design or to participate in the process of design. Consequently we as designers, or as designers of design processes, have had to be explicit as never before about what is involved in creating a design and what take place while the creation is going on.

The real subjects of the new intellectual free trade among the many cultures are our own thought processes, our processes of judging, deciding, choosing, and creating. We are importing and exporting from one intellectual discipline to another ideas about how a serially organized information-processing system like a human being—or a computer, or a complex of men and women and computers in organized cooperation—solves problems and achieves goals in outer environments of great complexity.

The proper study of mankind has been said to be man. But I have argued that man—or at least the intellective component of man—may be relatively simple, that most of the complexity of his behavior may be drawn from man's environment, from man's search for good designs. If I have made my case, then we can conclude that, in large part, the proper study of mankind is the science of design, not only as the professional component of a technical education but as a core discipline for every liberally educated person.
That was in fact the choice in our engineering schools a generation ago. The schools needed to purged of vocationalism; and a genuine science of design did not exist even in a rudimentary form as an alternative. Hence the road forward was the road toward introducing more fundamental science. Karl Taylor Compton was one of the prominent leaders in this reform, which was a main theme in his presidential inaugural address at MIT in 1930:

I hope...that increasing attention in the Institute may be given to the fundamental sciences; that they may achieve as never before the spirit and results of research; that all courses of instruction may be examined carefully to see where training in details has been unduly emphasized at the expense of the more powerful training in all-embracing fundamental principles.

Notice that President Compton's emphasis was on "fundamental," an emphasis as sound today as it was in 1930. What I am urging in this essay is not a departure from the fundamental but an inclusion in the curriculum of the fundamental in engineering along with the fundamental in natural science. That was not possible in 1930; but it is possible today.


I should like to underline the word "unnecessary." When I said something like this in another place (the second paper mentioned in the previous footnote), an able logician, who had specialized in modal logics, accused me of asserting that modal logics were "impossible." Now this is patently false: modal logics can be shown to exist in the same way that giraffes can—namely, by exhibiting some of them. The question is not whether they exist but what they are good for. A modal logician should have no difficulty in distinguishing "non-necessity" from "impossibility."

The use of the notion of "possible worlds" to embed the logic of imperatives in declarative logic goes back at least to Jorgen Jorgensen, "Imperatives and Logic," *Erkenntnis*, (1937–1938):288–296. See also my *Administrative Behavior* (New York:
Macmillan, 1947), chapter 3. More recently this same idea has been used by several logicians
to construct a formal bridge between the predicate calculus and modal logic by means of so-
called semantic or model-theoretic methods. See, for example, Richard Montague, "Logical
references are also given to work of Saul Kipke; and Jaakko Hintikka, "Modality and
Quantification," Theoria, 27(1961):119–128. While these model-theoretic proposals are basically
sound, none of them seems yet to have given adequate attention to the special role played in
the theory by command variables and criterial constraints.

51 "The traveling salesman problem" and a number of closely analogous combinatorial
problems—such as the "warehouse location problem"—have considerable practical importance,
for instance, in siting central power stations for an interconnected grid.

52 Notice that we are not saying that the two kinds of channels operate independently of each
other, since they surely do not in living organisms, but that we can distinguish conceptually,
and to some extent neurlogically, between the incoming and outgoing flows.


54 That this point is not obvious can be seen from the fact that most chess-playing programs
have used similar or identical evaluation procedures both to guide search and to evaluate the
positions reached at the ends of paths.

55 George W. Baylor and Hervert A. Simon, "A Chess Mating Combinations Program,"

56 I have developed this argument at greater length in my essay "The Architecture of
Complexity," chapter 7.

57 This approach to the design of complex structures has been explored by Christopher
He also has presented in his book some automated procedures for finding plausible
decompositions once the matrix of interconnections of component functions has been specified.

58 I am indebted to John Grason for many ideas on the topic of this section. J. Grason,
"Fundamental Description of a Floor Plan Design Program," EDRA1, Proceedings of the First
Number scrabble is not the only isomorph of tic-tac-toe. John A. Michon has described another, JAM, which is the dual of tic-tac-toe in the sense of projective geometry. That is, the rows, columns, and diagonals of tic-tac-toe become points in JAM, and the squares of the former become line segments joining the points. The game is won by "jamming" all the segments through a point—a move consists of seizing or jamming a single segment. Other isomorphs of tic-tac-toe are known as well.

My colleague, Allen Newell, has been investigating this question. I shall not try to anticipate his answer.


3. DESIGN ACROSS THE ARTS AND SCIENCES

Thomas Schwartz

The term "design," along with activities conventionally described thereby, figure less in philosophy than in many other disciplines. Yet the concept bears philosophic scrutiny. As elusive as Socrates' sophist, it escapes initially plausible attempts at analysis. Apparently essential characteristics of design, taken for granted in recent writings, may not be such; at least they are problematic. Activities that are hard to distinguish from design—including some that have lately been regarded as exemplars of design—resist the label in ordinary usage. The disciplines of theoretical science, often thought to be most dissimilar from the more professional, design-based disciplines, can be shown to involve quite as much design activity as the paradigm design disciplines.

The Essence of Design

As we look for features common to different design tasks and for features that distinguish design from other human endeavors, several generalizations about design readily come to mind, all of them mentioned, in one form or another, by previous contributors to this topic. These generalizations are problematic. Even those that are not demonstrably false nevertheless give rise to perplexities, thereby affording philosophers the opportunity to make a contribution. Here I discuss four such generalizations.

I. COMPOSITION. To design is (among other things) to compose a complex whole from components, or at least to specify in detail how to do so. Merely choose one color of paint over another is not to design anything, but to choose a combination of paint, carpet, chairs, lamps, etc., along with their placement in a room, is to design an interior.

The problem is that sometimes the most clever and meritorious solution to a design problem involves no composition. I once had to mount a papier-mâché parrot over a flower pot as part of a buffet-table display. The parrot had to look suspended over the plant, not resting on the dirt. My time constraint was ten minutes; my production-possibility frontier ended at the kitchen door. Noticing a whole in the parrot's bottom, I chose a shish-kabob skewer. I once needed a pedal mount for a piece of machinery. Before designing and making one, I looked around a hardware store and noticed that a ceiling-joist bracket, turned upside-down and backward, would do the job perfectly. In each case I solved a design problem by choosing a single ready-made object instead of composing one.

True, that object was in turn put to work in a complex whole. But consider an engineer asked to design a wrench to fit an oddly shaped nut in a hard-to-reach location. Before even
sketching such a tool, he looks through his company's hardware inventory and notices a piece of flat steel, oddly shaped, designed as a connecting link in a piece of machinery. Examining it, he sees that a cut-out in one end fits the nut, that it is thin enough to enter the hard-to-reach location, and that it is long and strong enough for the proper torque. He has solved the design problem without composition, and his solution is all the better on that account. The chosen object will figure in a complex process of a sort. But if that is taken to mean that he has contributed to a "composition," then all problem solving is similarly compositional, and our requirement of compositionality becomes vacuous.

Some examples of found art and conceptual art arguably have the purpose of gainsaying composition as an essential feature of art. Witness Duchamp's celebrated urinal.

How different, in terms of intellectual tasks, is the use of ready-mades from compositional design? The compositional designer searches his imagination for a complex of components. The artist or engineer who solves a problem with a ready-made searches a shelf or junk yard or catalogue for a complex of components. How different is such a process from the heuristic use by compositional designers of rough sketches, mock-ups, prototypes, and movable computer-screen displays? Not much. Not enough to justify the requirement of a compositionality. Let's drop it — appreciating, of course, that the problem of distinguishing "design" from other species of problem-solving is made no easier thereby.

By the way, federal courts (unwisely) recognize compositionality as essential to design. Clever, innovative, and useful though they be, the pencil-mounted eraser and the Weed Eater were denied patents on account of extreme compositional simplicity.

II. CREATIVITY AND INNOVATION. To design is (among other things) to create, to innovate. Someone who puts together a model airplane from a kit does not design anything. Someone who inadvertently reinvents the widget may have faced and solved the same problems and gone through the same intellectual tasks as the originator, but we do not attribute the design to him: he did not invent the widget.

This is true only if carefully qualified to avoid certain connotations of the words "creativity" and "innovation." For one thing, those words often suggest a high degree of freedom of choice. Yet designers typically try to meet specifications — goals and constraints — that severely limit their freedom of choice. Often the greater the constraints, the more challenging the design problem is, and the more praiseworthy and even creative its solution. Edison's problem of designing a device that would produce symphonic concerts and the like in an average person's living room would have been trivial but for the economic constraints: he
could have specified the construction of a fully staffed and equipped symphonic hall next to any living room, along with an connecting pipe. One can imagine constraints so severe that just one possible design fulfills them: the designer must be especially clever even though — nay, just because — he has no freedom of choice.

It may seem paradoxical that the greatest designers can have the least freedom. The solution to the paradox is that constraints, by reducing the range of acceptable alternatives, thereby increase the range of alternative through which one must search for an acceptable one. The broader the range of acceptable design choices — the greater the designer's freedom — the less far he must search for an acceptable choice, hence the less creative he can be. Properly understood, creativity and innovation do not involve freedom of choice. They do involve a breadth of alternative, but in the sense of search alternatives rather than choice alternatives. Reducing freedom increases creativity.

The words "creativity" and "innovation" also suggest novelty. Yet the best solution to a design problem can, as we have seen, be the selection of a ready-made object. True, the use may be novel even if the object is not. Sometimes, however, an old design problem with an accepted solution is reopened and the accepted solution carefully, cleverly, and convincingly shown to be the best solution: an excellent solution need involve no novelty of product or use. Perhaps the best way to handle such cases is by distinguishing solution to design problems from design: the best solution may be no design.

Finally, the words "creativity" and "innovation" suggest a minimum of rules or instructions for the designer to follow. Yet to develop a science of design and a curriculum for teaching design, to find ways of making people better designers, is precisely to seek rules or instructions of some sort for designers to follow.

The apparent paradox can be solved by noting that good rules and instructions would increase a designer's power much as a ladder increases a painter's reach. Using the same ladder, a tall painter will still reach higher than a short one. Using the same rules and instructions, a talented designer would still have greater power than an untalented one. Properly understood, a painter's height consists, not in his total reach, but in his marginal reach —what he adds to the ladder —and a designer's creativity or innovativeness consists, not in his total power, but in his marginal power — what he personally adds to the capabilities afforded him by an appropriate set of rules and instructions.

III. ART AS DESIGN. Design is what the arts share with engineering. Whatever else their differences, artists and engineers are both designers. And it is this fact that spawns the
The problem is that the classification of artists of several types as "designers" appears to clash with ordinary usage. You can design a car, a kitchen, or a concertina. But it is odd at best to say of a painter that he has designed a picture, of a composer that he has designed a symphony, of a writer that he has designed a novel (a publisher's book designer designs a particular edition of a novel, not the novel itself). Why?

One difference between painting and engineering is that the design of a car, for example, can have any number of concrete realizations, whereas a painting is unique: it can be copied (forged, maybe), but no copy can be called the "same painting," and if the painting is good enough, no copy, however faithful, can have near the value of the original. Unfortunately, novels and musical compositions, like car designs and unlike paintings, can have any number of concrete realizations, yet we do not call novelists and composers designers.

One might suggest that in engineering, design and execution are distinct activities, whereas no comparable distinction exists for painting, writing, or composing. But garage inventors often design while executing, and, on the other hand, painters and writers sometimes begin with sketches and outlines.

One might further suggest that a complete design for a novel or musical score — one which omitted feature — would be the same as the novel or score, whereas a complete design for a car would not be a car. One might add that a complete design for a painting would specify every brushstroke in detail, which is either impossible or no different from making the painting itself. But a "complete" car design is not really complete: it allows minute variations in size and shape, in file marks, in the grain of wood and leather, and the like.

Notice, however, that we have conventional (if imprecise) ways of distinguishing those features of a car that are part of its design from those attributable to manner of execution. We can make the distinction by looking at a concrete car, never seeing the engineer's drawing. By contrast, we have no conventions for distinguishing the "design" of a painting or novel or score from other details of it. We could conceivable develop such conventions (we have done so for welded-steel sculpture). The reason we haven't may be economic: owing, perhaps, to scale economies, we find it profitable to distinguish design from execution in engineering, architecture, book-jacket design, etc., hence to create separate markets for designs and designers and for machines and machinists. Cf. scenery design, which is similar in some ways to painting: we distinguish design from product because of the need for a large labor force, replacement of components, and road-company replication.

hope of a broadly interdisciplinary science and curriculum of design.
The key conceptual distinction has to do more with the noun than the verb. A painting has no "design," thanks to the absence of appropriate conventions. Therefore, we don't say of a painter that he designs his product. His intellectual task may, however, be as similar as you please to that of an architect, cereal-box designer, or couturier. For this reason, I suggest that we can usefully allow a science and a curriculum of design to embrace the likes of painting, novel writing, and musical composition.

IV. DESIGN AS ARTIFICIAL. Design distinguishes the professional disciplines, which create new, artificial things, from the theoretical and scientific disciplines, which analyze how "natural" things work. Simon says:

Historically and traditionally, it has been the task of the science disciplines to teach about natural things: how they are and how they work. It has been the task of engineering schools to teach about artificial things: how to make artifacts that have desired properties and how to design.

Engineers are not the only professional designers. Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences...

In view of the key role of design in professional activity, it is ironic that in this century the natural sciences have almost driven the sciences of the artificial from professional school curricula. Engineering schools have become schools of physics and mathematics; medical schools have become schools of biological science; business schools have become schools of finite mathematics. The use of adjectives like "applied" conceals, but does not change, the fact. It simply means that in the professional schools those topics are selected from mathematics and the natural sciences from emphasis which are thought to be most nearly relevant to professional practice. It does not mean that design is taught, as distinguished from analysis.

A similar distinction is drawn when it is said that science precedes technology, that the engineer applies what the scientist discovers. However drawn, the distinction is fundamentally mistaken. Especially if we think of design as an intellectual activity rather than an economic good, science is no less concerned than technology with design. Those who analyze, test, or
deduce are just as much designers as those who create, fabricate, or invent, and there is not much difference between the intellectual problems, norms, and techniques of the two groups.

In the remainder of this paper, I examine the tasks involved in analysis, testing, and deduction, attempting to show that all involve design.

Analysis

Analysis, or analytical reasoning, is the intellectual activity involved in concocting explanations of observed phenomena, established generalizations, or accepted beliefs.

Sometimes a good explanation is deduced from an unchallenged theory, and sometimes we must choose among competing hypothetical explanations, chiefly by testing. I discuss deduction and testing in later sections. Here I look at the process of concocting rather than deducing or testing explanations.

Good explanation requires modeling. It is not enough to say that X happened because Y had happened, nor to subsume X under some descriptive generalization. One must hypothesize a process or mechanism or structure (a model) that would (if it existed) produce the given explanandum. Rarely if ever can one "directly" and completely observe the underlying process. One infers that it is there by showing that its hypothesized existence predicts things we can (more directly) observe and that it does not yield false predictions.

Suppose, as part of a buffet-table decoration, that I want to design a fountain consisting of little angels urinating into a basin. One approach would be to run a hose from a water tap to the angels and another one from the basin to a sewer drain. A superior solution to this design problem would be to have a battery-powered water pump circulate water from basin to angels.

How different is Harvey's celebrated model of the heart as a pump in a circulatory system? His innovation was the hypothesis that the heart is a pump rather than a furnace and that blood does not terminate in target organs but completes a circuit. Although he found confirming evidence for his hypothesis, he did not directly observe the heart acting as a pump in a circulatory system, else earlier anatomists would have observed it as well. He hypothesized a process that he had first to design. Like most designers, he made some use of ready-mades: the pump was an old idea. As in the case of the urinating angels, he had to design a mechanism to achieve a given set of consequences. The difference, of course, was that his consequences were observed rather than desired. But that made no difference to the design problem.
One could make similar points about the Copernican theory, the planetary model of the atom, the molecular theory of gasses, or the double-helix model of DNA and RNA. In each case someone designed a mechanism or process (sometimes drawing upon ready-mades) to produce certain consequences. In each case, technology preceded science.

Here are some social-science examples:

Example 1. Elections and echoes. How to explain that parties and candidates often provide voters with an echo rather than a choice? The problem is to design an electoral process that compels candidates to take similar positions on major issues.

Here is such a design: There are two candidates, L and R. Each can choose any policy position along the liberal-conservative continuum. The sole goal of each is to win a majority of votes. Initially, at least, L chooses a position to the left of R's. Who votes does not depend on the positions the candidates take. Each voter has his own ideal position along the liberal-conservative continuum: he likes a candidate less and less the farther the candidate's position lies from his own.

A *median* of voters' ideal points is a position P such at no majority of voters have their ideal points to the left of P or to its right. To simplify, suppose there is just one such median (there could be two); call it M. Suppose L's position is closer than R's to M. Then the voters at or to the left of M are closer to L's position than to R's, so they prefer L to R. But they are a majority. Likewise, if R's position is closer to M then a majority prefer R to L. Consequently, each candidate will try to take a position closer than his opponent's to M. As a result, candidate positions will converge to M, providing voters with an echo, not a choice.

Example 2. Market failures. Often we notice that each of a group of people would benefit were all of them to behave a certain way, yet none behaves that way. Suppose 1,000 people inhabit the shores of Limpid Lagoon. Each has a choice between installing a septic tank and flushing his sewage into the lake. All do the latter. As a result, all suffer from a polluted lake to such a degree that each had rather that he and everyone else had installed a septic tank. Yet the pollution continues. How to explain this?

The problem is to design a social institution for making such decisions which ensures the observed outcome.

Solution: a market system for providing goods. Under such a system, each person must
pay the cost of his own septic tank. Although the harm he does by polluting exceeds that
cost, the harm is divided among the thousand riparians, so that he incurs only one thousandth
of it, whereas the cost of his septic tank must be born by him alone.

If correct, this institutional design suggests any number of redesigns that would cure the
problem e.g., a pollution tax.

Example 3. Political failures. We observe that an elected legislature often produces
Pareto-inefficient legislation: everyone would be better off if the legislative outcome were
different. How could majority voting have such a consequence?

The problem is to design the legislative process so that majorities knowingly legislate to
everyone's disadvantage.

Solution: In the absence of disciplined parties, majorities shift from issue to issue: the
majority which passes one bill often is different from the majority which passes another bill.
In each case, the winning majority tries to benefit itself, ignoring the interests of the losing
minority. To simplify, suppose there are three legislators and (at least) three bills, a, b, and c,
to which the legislators assign values, in millions of dollars as follows:

<table>
<thead>
<tr>
<th></th>
<th>Mr. 1</th>
<th>Mr. 2</th>
<th>Mr. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4</td>
<td>4</td>
<td>-9</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>-9</td>
<td>4</td>
</tr>
<tr>
<td>c</td>
<td>-9</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Since each bill benefits a majority, all three pass. As a result, everyone loses $1 million
$(4 + 4 - 9)$.

This example is related to an old philosophical problem of institutional design. Rousseau
regarded the General Will as a kind of unanimous will, thought everyone would be better off
if the General Will were sovereign that he would be in a state of nature, and contended,
somewhat paradoxically, that the General Will could be elicited by a direct democracy using
majority rule, but only if there were no fractions. He may have had in mind examples like
this:

<table>
<thead>
<tr>
<th></th>
<th>Mr. 1</th>
<th>Mr. 2</th>
<th>Mr. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>-7</td>
<td>4</td>
</tr>
</tbody>
</table>
Since each bill benefits a majority, all three pass. As a result, even though each bill harmed the losing majority, everyone is better off, receiving $1 million (4 + 4 - 7). But if Messrs. 1 and 2 formed a fraction, they would vote only for a, making Mr. 3 lose $7 million — possibly worse for him than the state of nature.

Compare the two examples. In the first but not the second, everyone suffered because each bill was inefficient, not cost effective: the cost to the losing minority exceeded the gains of the winning majority. Which case is more likely? The problem is first to design a plausible process by which those who pass bills decide levels of benefits and costs, then to see whether this process yields cost-effective legislation.

Solution: Those proposing any project try to maximize the net benefit they receive. Hence, they raise the scale of the project to the point at which marginal benefit to them equals marginal cost to them. So if those proposing a project pay the whole cost, they will make it cost effective. But if the cost is subsidized by others—the losing minority—they will raise the scale of their project above a cost-effective level. With the bill-writing process so designed, the first case is the likely one: like markets, majority voting can hurt everyone; unlike markets, it usually will hurt everyone.

Example 4. Riding Reagan's coattails, Republicans won a majority of votes for the U.S. House of Representatives in 1985. But the Democrats kept a clear majority of seats. How might we design a vote-aggregation process, consistent with the structure of our system of government, that yields this result?

Here is one suggestion: To win a majority of House seats, a party must win a majority of votes in each of a majority of congressional districts. But district boundaries are drawn by state legislatures, most of them Democratically controlled. Their incentive is, therefore, to draw large Republican and small Democratic districts, thereby lessening the value of Republican votes.

This design is fine, except that it is false: since *Wesberry v. Sanders* (1964), congressional districts have had to be equal in population.

Try again: Democratically controlled legislatures will draw equal-size districts but will do so in such a way that part of the Republican electorate is highly concentrated in relatively
few, predominantly Republican districts while the remaining part is dispersed among many, predominantly Democratic districts. Here is a simplified illustration: There are three districts, each with three voters. One has three Republicans while each of the others has two Democrats and one Republican. Therefore, the Republicans have a majority of votes overall, but the Democrats have a majority in a majority of districts.

Example 5. The congressional committee system. The U.S. House of Representatives is divided into small subcommittees with self-selected membership. Legislation reported by any subcommittee is almost always passed by the House. Why?

Here is the design of an electoral process that would produce the above consequence. With decentralized parties, each congressman runs for election and reelection as an individual producer of benefits for his constituency. The residents of each district are more concerned with getting benefits targeted at them (water projects, agricultural subsidies, Defense contracts) than with blocking benefits for other constituencies—something they don’t think their representative could succeed in doing anyway.

If legislation were chiefly decided by the House as a whole, then a U.S. Representative would be just one of 435 voters deciding all legislative issues. But when legislation is decided by subcommittees, he can be one of a few Representatives with common interests deciding those issues that are of greatest importance to his constituency: his chances are enhanced of being pivotal on matters that most concern his constituents.

All of these examples come from analytical as opposed to descriptive social science. Consider the historian, whose product usually is description or than analysis. What does he design? The past that he describes is not presented to him full blown, like a movie, ready to be turned into narrative. It comes in the form incomplete, unclear, noisy, sometimes spurious records. He must write a story that best fits those records. His problem is quite similar to that of a novelist, except that he is constrained by records. Where Harvey explained the cardiovascular system by designing it, the historian describes the past by designing it.

Test Cases

To assess the truth of general assertions — empirical hypotheses, philosophers' definitions and normative principles, putative theorems — we use test cases. But we do not simply look for them, grazing about until one of them strikes our eye. We design them.

We have observed, let us suppose, that certain students regularly sit in the front of a lecture hall while others regularly sit in the rear. Wondering why, we entertain two
explanations:

(A) Those who sit in the rear are shier. They fear being noticed by their fellow students or by the professor and perhaps being called upon to answer questions.

(B) Seeking to minimize the time it takes to enter and leave and walk between classes, each student sits near the door that minimizes his travel time.

To choose between these explanations, we might design a **critical test**. Call the students who sit in the rear *R students*; the others, *F students*. Now alternately lock the front doors to the lecture hall and the rear doors. Or follow a sample of *R* and *F* students to other classes, including ones in rooms with a single door. Observing where the students sit, we might find strong confirmation or disconfirmation for (B). Or we might find that (B) explains the behavior of a cross-cutting category of students, but not all students.

Some empirical tests — notable laboratory experiments — are created. In that sense they are artificial, although not so artificial that they guarantee a specified result — unless they are bad tests. The test above involves the observation of "naturally" occurring events. But although not artificial, it is *designed*. It is more economical to design a test case, then look for naturally occurring events that fit the design, than to gaze about until some event strikes us as a good test case: without a design in hand, we might see an event that would serve as a test case yet not realize it.

Test cases figure in philosophy as well as science. There they are *counter-examples* (or putative ones)—apparent exceptions to general normative and conceptual principles. Designed to refute, they might fail, thereby lending support—not just because the principle fit the case but because it fit a case *designed* to refute it.

A famous example is Edmund Gettier's counter-example to the classic definition of *knowledge as justified true belief*: I recall putting a dollar in my pocket and not removing it. Unbeknown to me, a pickpocket stole it. So I believe I have a dollar in my pocket, and I have justification for this belief, but it is not true. As it happens, my friend Preston just won the Irish Sweepstakes, but I am unaware of the fact: it is true, but I fail to believe it hand have no justification for believing it. Still, I believe the proposition that *either* I have a dollar in my pocket *or* Preston won the Irish Sweepstakes. What is more, I have justification for this belief, and it is true. Yet I cannot be said to *know* it, because I believe it for the wrong reason.

Gettier successfully set out to construct an hypothetical example meeting certain
specifications (justification, truth, belief, absence of knowledge). What he solved was a design problem if ever there was one.

Another prominent philosophical thesis (a definition) is that welfare is preference-satisfaction: something is good for a person to the extent that it satisfies his preferences. A related thesis (a normative principle) is that the criteria of fair distribution (whatever they be) ought to apply to the distribution of preference-satisfaction. Here is a counter-example to both: Crusoe and Friday have contributed equally to the day’s catch of fish, and they have similar metabolisms, appetites, energy and health needs, stomach sizes, and capacity for gustatory appreciation. But Crusoe is greedy, selfish, and gluttonous: he demands more food than he needs or can comfortably consume, and he takes pleasure in the mere perception that he has more than Friday. If the first thesis were correct, we should enhance Crusoe’s welfare by giving him the lions share of fish. this is plainly false. If the second were correct, justice would demand that Crusoe receive more fish than Friday. That, too, is plainly false.

I offer one last example to illustrate the process of designing counter-examples: Some philosophers have defended abortion on the ground that killing a fetus violates no right it may have not to be killed, because you cannot violate a right without violating a corresponding desire, and fetuses have no desires. Now consider the premise (a normative principle) that doing X cannot violate Y’s right unless Y desires that X not be done. What would a test case—a counter-example—look like? It would be a story in which someone violates my right to something but I don’t mind—maybe I am even glad he did so. That is our design specification. Let us try meeting it. How can a person not mind—even want—the violation of his right? One might think of a masochist wanting to be tortured. But such a case is not clear, not convincing: it is not at all evident that a right is being violated. Let us look, therefore, for rights violations of a more humdrum sort: you steal my property, hit my car, insult me, or whatnot. Could I be glad of such an event? Yes. It takes less than a prodigious gift of imagination to realize that I might prefer the insurance money to the property you stole or to an unblemished car body, or that I might want you to insult me as an excuse to do you some harm.

Deductive Logic

We face design problems in deductive logic when we construct or reconstruct arguments, prove theorems, and concoct and investigate axiomatic theories.

Arguments. To defend a conclusion is to design a supporting argument. But the analysis and evaluation of a given argument also is partly a matter of design. Good analysis consists,
to a great extent, in reconstructing the argument as a fully explicit valid argument. That is a design problem based on three constraints: deductive validity, fidelity to the given argument (insofar as it is clear), and plausibility of premises (insofar as this is compatible with fidelity).

Example:

It was wrong to attack Libya. True, Kaddafi had no business supporting terrorism. But two wrongs don't make a right.

This is not a fully explicit valid argument. To make it so -- to design a plausible, valid argument compatible with the given text -- we begin by identifying the conclusion:

It was wrong to attack Libya.

and the single stated premise:

Two wrongs don't make a right.

Invalid as it stands, the argument must be made valid by supplying a likely tacit premise or premises. This one will do:

Attacking Libya was one of two wrongs.

We now have a pretty good reconstruction. It is especially valuable because it brings out a flaw hidden by the original formulation: the tacit premise in question - begging.

Another example:

It would be wrong for the U.S. to violate its part of the hostage deal because it is generally wrong to break promises.

Here is a simple reconstruction:

It is generally wrong to break promises. (express promise)

For the U.S. to violate its part of the hostage deal would be to break a promise. (tacit premise)

It would be wrong for the U.S. to violate its part of the hostage deal. (conclusion)

If it were not obvious what the tacit premise was, we might have found it by means of a simple design heuristic: The stated premise links two terms, "break promises" and "wrong," and the conclusion links the latter to a third term, "U.S. to violate is part of the hostage deal." What was wanted was a tacit premise that closed the circle, linking the third term to the first.

The first premise is open to counter-examples: If you force me at gunpoint to promise
to steal the crown jewels, I have no obligation to keep my promise. Perhaps the argument's author meant to exclude such cases, tacitly qualifying the first premise thus:

It is generally wrong to break promises except coerced ones.

But now the argument is invalid. To restore validity (our chief constraint) requires some redesign: in the second premise, we must insert "uncoerced" before "promise." But that makes the premise flatly false.

Designing Proofs. Constructing a formal proof from premises — the sort of thing taught in baby-logic courses — is another type of design problem. To teach it as such, one must not only distinguish correct from incorrect proofs, but present techniques for finding correct proofs.

A natural-deduction system based on paired introduction and elimination rules for logical constants lends itself especially well to proof-design techniques that might carry over to informal mathematics. This is because of Prawizy's Normal Form Theorem, which says we can construct any proof (at least in part) from the bottom up, applying introduction rules in a unique, mechanical way. Once we have done this as far as possible, either we have a proof, or (assuming provability) the proof can be completed by using only elimination rules at the top. There can, of course, be no algorithm for the latter, but the bottom-up procedure reduces and structures our design problem considerably.

In mathematics, most proofs do not depend primarily on the intricacies of truth-functional and quantificational structure. Many of the most profound and interesting proofs do depend on constructions, however, and these are solutions to design problems. To prove an existence theorem, one designs a mathematical object (a construction) to have the properties stated in the theorem. An example is the Schroder-Berntein Theorem: one gives a general design for constructing a single one-to-one mapping out of two given ones. Another is the proof that the rational numbers are countable: one designs a counting procedure. The proof that the real numbers are not countable involves two constructions: one describes the general design of an hypothetical counting procedure, couched in terms of decimal expansions, then designs a particular decimal expansion (the anti-diagonal) that the procedure perforce misses. In an introductory mathematical logic course, one typically proves Godel's Completeness Theorem by designing a maximally consistent extension of any consistent set of formulas: one actually lays out a plan (nonfinitary, to be sure) for constructing such an object.

Axiomatic theories. Yet another conspicuous deductive-logic activity is that of designing
axiomatic theories. We start with some results — maybe a whole body of knowledge — and reduce it to a simple and variously constrained set of axioms.

To motivate one example, notice that, when a choice is made between two alternatives by voting, majority rule is almost invariably used. Why?

Suppose n voters, Messrs. 1, 2, ..., n, are to make a collective choice between two alternatives, represented by the numbers 1 and -1. Let 0 represent a tie. A vote combination, representing a possible way Messrs. 1, 2, ..., n can vote or abstain, is an n-fold combination, or vector, (x1,...,xn), in which each xi is 1 (a vote for 1 by Mr. i), -1 (a vote for -1), or 0 (abstention by Mr. i). Any two-alternative voting rule may be represented by a function f such that:

(F) $f$ is a function of vote combinations, and for every vote combination $(x_1,...,x_n)$, $f(x_1,...,x_n)$ is equal to 0, 1, or -1.

What further properties might we want to expect $f$ to possess? These three have strong claims on our intuitions:

(A) $f(x_1,...,x_i,...,x_j,...,x_n) = f(x_1,...,x_j,...,x_i,...,x_n)$ (Anonymity).

(N) $f(-x_1,...,-x_n) = -f(x_1,...,x_n)$ (Neutrality).

(T) If $f(x_1,...,x_n) = 0$ and $(y_1,...,y_n)$ is the result of replacing one or more 0's by 1 in $(x_1,...,x_n)$, leaving all else the same, then $f(y_1,...,y_n) = 1$ (Fragility of Ties).

Axiom (A) says $f$ treats voters equally: if any two voters switch votes, the collective choice remains the same. Axiom (N) says $f$ treats alternatives equally: if every vote is reversed, so that 1's become -1's and -1's become 1's, the choice is thereby reversed (or remains the same if it was 0). Axiom (T) says ties are easily broken: if there is a tie and some erstwhile abstainers then vote for 1, other things remain unchanged, that is enough to bread the tie in 1's favor. A voting rule that lacked any of these properties would have a built-in bias against some voters or alternatives, thereby allowing elections to be decided to some degree by factors other than votes, or it would be gratuitously indecisive, allowing ties to persist in the face of new information that plainly tilted the balance in one direction.

Simple-majority rule obviously has all these properties. It holds its distinguished position because it is the only two alternative voting rule with all these properties. That is, our axioms have the following consequence:

$l$ if $x_1+...+x_m > 0$

$f(x_1,...,x_m) = 0$ if $x_1+...+x_m = 0$
Another example: We observe a bipolar world. There is a tendency, at least, for nations to split up into two groups. Any two members of either group are allies, but no one in either group is allied with anyone in the other group. What general properties of bilateral relationships would account for this polarization?

Letting A be the relation between allies, consider these axioms:

(R) \( xAx \) (reflexivity: everyone is his own ally).

(S) If \( xAy \) then \( yAx \) (symmetry: alliances are mutual).

(T) If \( xAyAz \) then \( xAz \) (transitivity: allies of allies are allies).

(NNA) If not \( xAy \) and not \( yAz \) then \( xAz \) (nonallies on nonallies are allies).

The first two axioms imply that alliances constitute a covering of the set of nations, the third that this covering is, more specifically, a partition, and the forth that this partition is more specifically still, a dichotomy.

When we examine axiomatic theories, design problems reemerge: we design models to prove consistency, incompleteness (negation-incompleteness and noncategoricity), and independence (of primitive terms, axioms, and putative theorems). The task is similar to that of designing counter-examples and constructions. (In the latter connection, Godel’s M-sets and Cohen’s forcing are models used to prove consistency and independence, but with a twist: they are inner-models).

In the first theory, majority rule itself proves consistency. Weighted majority rule is an independence example for (A), two-thirds majority rule for (N), and two-alternative sample surveys for (T). Here, once again, we solved design problems by means of ready-mades.

Conclusion

Design is as much a part of science and philosophy as it is of engineering and art. Analysis, testing, and deduction are preeminent design tasks, involving (I hazard to speculate) much the same norms and skills as characterize professional and technological pursuits. Science does not precede technology.

Let none of this be taken to challenge Simon’s chief conclusion: that the professional disciplines have neglected the study of design. So, however, have the theoretical disciplines.
So, indeed, have all disciplines neglected the study of the intellectual tasks on which they rely.

That should be no surprise. Physicists are expert in the study of matter and energy. There is no reason why they should also be expert in the study of the way people study matter and energy. To be good at investigating X is not necessarily to be good at investigating the investigation of X.

How we face a problem: Only those skilled at investigating X can pass that skill onto others. They alone can exhibit the skill and evaluate its exercise. The solution is collaboration between those skilled at investigating X and those skilled at investigating the ways in which things are investigated.
4. ACADEMIC WRITING DESIGNEING FOR THE COMMUNITY

David Kaufer

Writing is a well-known design process. Academics know it as among the most difficult of processes to master. Small wonder. Academic writing is designing for the community. An academic paper is not a stand-alone product. It is not like making a shoe. Instead, it is a strategic effort to make an addition or improvement to an evolving, rather amorphous, product called "disciplinary knowledge." What academic writers design are not words on paper but additions or improvements to this amorphous product. Students can't understand academic writing unless they understand the product they are trying to extend or improve — and the community for whose benefit they are trying to extend or improve it.

In this paper, I shall describe some perplexities in teaching students how to design written products for academic communities. I will also offer some solutions. As a way of introducing the perplexities, I want to characterize academic writing in some detail.

Academic Writing

In academic writing, the writer also produces an immediate, tangible product, a document — essays, reports, monographs — for a community of other writers and readers who are, or may be, drawn to the issue under discussion. The academic writer submits a product as a "contribution" to the community. Academic communities usually rely on screening committees (journal referees, editors) to evaluate the submission before it is accepted as an official product, and, indirectly, as a contribution. When a screening committee agrees to place a paper in a respected journal, their agreement is much like that of an investor whose money puts a play on Broadway. The play may fold after opening night, but at least enough "right" people thought highly enough of it to give it a run.

An academic contribution is something like a turn in a conversation. The academic writer usually must synthesize previous turns in the conversation and then show how his or her own turn emerges as a response to these. An academic contribution is also unlike a turn in a conversation. Conversational turns can be relatively spontaneous. Academic turn-taking involves the strategic organization (or reorganization) of huge amounts of information. Often this strategizing takes place in quiet disinterest from the other goings on in the community. Paradoxically, important writers, writers with a competitive interest to dominate the direction of the community, may invest enormous amounts of time thinking as a "disinterested outsider" in order to gain the fresh perspective needed to change the community radically from within.

The goal of academic writing is to say something new. Saying something new to a
community is not like answering the casual question, "What's new?" It is rather like saying something that will have the leverage to redirect and refocus the community's attention. It is more like saying, "Your underwear is showing." It is a form of exposure, but constructive exposure. "Your underwear is showing, but here's how to cover it up"; or "Here's how to learn despite your inattention to what's staring you in the face." Every academic community relies on a large number of primary situations, cases, exemplars, that people in the community find curious, problematic, in need of explanation. It also relies on a repertoire of analytic techniques to characterize these situations and to deal with them.

Often writers spot the leverage they need to say something new to a community by noticing that the community has remained inattentive to or somehow overlooked an important class of primary situations. That is, leverage comes with the insight that the community hasn't been looking quite at all the right situations or, at least, in all their right aspects. Or writers will notice that the community has been looking at the right situations but that no one has yet to apply just the right analysis to them. In any case, the writer notices a problem, a lack of fit, a discrepancy between where the community is and should be. And the writer takes rhetorical advantage of this gap.

Learning how to find one's rhetorical leverage is an important part of designing for the community. It is, of course, not the only part. If you imagine a contribution to the community as a "new path" through an issue, then considerations of rhetorical leverage can direct the writer to paths that would, were they well-supported, hold leverage. Considerations of leverage, however, should not be confused with considerations of evidence or support. Experienced writers often make the distinction between the paths that carry the most clout and the paths they can actually support. Paths they can support depend on accepted practices of evidence and reasoning within a particular field.

A community classifies writing as polemic when it observes that the writer continues to propose an important path long after the accepted means of supporting it have been exhausted. Academic communities work to maintain a balance between clout and accepted methods for supporting assertions. Important contributions are expected to be well-supported and at the same time rhetorically impactful. Less important contributions need to be well-supported even though they lack the impact of more important contributions. Paths lacking the accepted means of support are to be dismissed without credit. It doesn't always happen that way in practice, since academic communities are also political communities—but that is at least the ideal we preach.
In undergraduate education, students often spend so much time learning the accepted methods for supporting assertions within a discipline that they rarely learn that the real gain of these methods is to empower them to propose new paths of their own to the community. And it is often not until they worry about publishing when graduate students first hear about the importance of rhetorical impact or leverage when designing for the community. Most every graduate student is shattered when he or she first hears the communal "so what?" in the voice of an advisor.

The Current Status of Instruction in Academic Writing

I offer these summary statements about academic writing to illustrate the forbidding challenge confronting writing teachers when they try to teach academic writing to undergraduates. On occasion, my colleagues and I are confronted by teachers in other disciplines who, upon receiving a student paper they believe to exemplify "poor writing", show us the offending evidence and then, in a tone of impatience and indignance, ask "What are you going to do about this?" The summary of academic writing in the last section should make clear the large and amorphous thing their this might be referring to.

Whenever such a question is posed to me or colleagues who also teach writing, our first response is usually something on the order of "What is the this I am supposed to do something about?" — a response, typically, that stops the questioner cold. The questioner presumed a "writing expert" would have a vocabulary for the this and would be able to pinpoint its exact referent on sight. And yet when the questioner finds out that even "writing experts" lack operational names (or correctives) for the most important lapses editors detect in academic writing, the conversation can turn awkward. Teachers in the special disciplines often have a hard time understanding that just as in their discipline, there are far more unknowns than knowns when it comes to writing. They often have a hard time understanding that, just as in their discipline, writing experts have more questions than answers.

Granted, writing teachers can easily enough identify lapses attributable to misplaced grammar or punctuation. We have a standardized vocabulary for the mistakes of visible language. But when the "writing problem" seems to involve the coherence of the paper, its reasoning or evidence, or rhetorical impact, all of us (both writing expert and content expert) find ourselves on shakier ground when it comes to answering what went wrong. The problem is not so much that the writing teacher doesn't know enough about the content domain to diagnose the problem. The problem is that writing experts and context experts alike have been working with impoverished theories of writing as a way of designing for the community. Our current theories usually stop with principles for improving the readability of surface prose.
In the national movement to distribute writing across the curriculum, the limitation in our theory has negatively influenced the way we divide the responsibilities between writing teachers and content experts. Because both writing teachers and content experts have traditionally shared the same limiting theory, there has been little imaginative space for either to work within when dividing responsibilities. The writing teacher is given the duty to enforce principles of readable prose in the writing class. The content teacher is given the duty to enforce these same principles implicitly in the content classroom. What the writing teacher does explicitly, the content teacher can do implicitly. Explicit or implicit, however, what we currently teach our students about writing is usually only a small fraction of what they actually need to know to design for a community.

A Proposal

A healthier division of labor between the writing expert and the content expert would make the writing expert responsible for a general theory of designing for the community. The writing expert would look to designing in specialized disciplines both to test the general theory and to extend and refine it. The content expert, in turn, could work within the framework of the general theory and specialize it to accommodate the particular idiosyncrasies of his or her field.

A virtue of this approach is that it puts the greatest emphasis on what is common about writing across disciplines rather than what is different. When the emphasis is on differences, the writing expert and content expert have little to say to one another. The writing expert can’t possibly master the individuating aspects of a particular field. The content expert can’t possibly turn these individuating aspects into a general theory of designing for a community — which is what students seem to need in order to acquire the skill.

In brief, when the emphasis is on differences, all the writing expert and content expert can do is explain why teaching writing in the disciplines is not their job. But when the emphasis shifts to commonalities across disciplines, then the writing expert and content expert find themselves in a symbiotic relationship with substance. The writing expert needs the content expert to develop general theory. The content expert needs the general theory to teach students how to design for specialized communities.

Challenges in Bringing Design Principles to Teaching Academic Writing

What I am describing is a bold and starkly innovative role for the writing specialist as a theorist of design. It is a role already being played out and tested at institutions like Carnegie Mellon. Nevertheless, the role is still very much in its infancy. It is still being test piloted. It is a role that forces us to feel our way around. Most importantly, perhaps, the role gains
credibility only on the assumption that a steady stream of active research is informing the writing expert's attempts at theorizing.

The role requires a research component because academic writing, on its face, does not seem easily amenable to principles that we'd prefer to take for granted when generalizing about, much less teaching, design processes. We need research in order to learn how better to accommodate these principles to the teaching of writing. Just what are these principles? For the teaching of writing, the most important principles to accommodate seem to be the following:

1. having at least some goals (or subgoals) in the process that are procedurally defined;
2. having a design "space" that is reasonably well-bounded;
3. monitoring the student's forward progress through the space at frequent intervals;
4. being able to direct the student backward through the space to correct poor decisions;
5. allowing room for diagnosis and assessment at frequent intervals.

I will discuss each of these principles and show how our current practice of teaching academic writing accommodates none of them.

**Procedural Goals.** When we teach a design process, we are greatly helped when we can rely on at least some goals at some level in the process being well-defined. By well-defined, I mean they have a precise, operational, even procedural meaning. Many problems in math are ill-defined. But at least they call on a set of subgoals (multiply, add, take the square root) that have precise operational meanings. In many design problems, we don't keep compounding unknowns with more unknowns. Unknowns can be broken down into knowns and the knowns can then be collectively bootstrapped to fill in unknowns.

When we teach writing, our goals for students never quite reach the procedural level of a schedule of explicit activities. We rely on a command language that asks students to "summarize", to "synthesize", to "get a writing plan", to "write with polish and vigor." We don't usually think of translating this command language into more procedural information. Were one to ask a writing teacher, "What do you want students to do when you ask them to summarize or synthesize?", one would get few answers, and certainly few consistent ones. We have an impoverished procedural framework for guiding students on the components of higher reasoning that inform academic writing.67

The absence of procedural goals from instruction in academic writing is apparent in the
conventions which dictate how writing textbooks are authored and deployed for classroom use. As an instructional genre, the best-selling texts are written not as tutorials, but as reference manuals. They don't tell students how to move, moment by moment, through this or that writing task. They don't tell students what to do when. Rather, they tell students what will be "useful to know" sometime for some indefinite writing project. They offer advice disembodied from the rich environment of designing a specific paper under a specific set of constraints. Students are not expected to go to a writing text to write something. They are expected to go to a writing text to read about the activity.

*bounded outcomes.* When we teach design, it is helpful when the student, during the design process, can make ever closer approximations to the product he is she is supposed to be producing. When a student's task is to design a computer program or a quantitative model, for example, it is important for the student to be able to narrow in on what, minimally, it is expected for the program or model to do. The student should get constant feedback on whether he or she is getting closer or farther away from the desired end product. For example, if a student is asked to design a computer program that, minimally, retrieves information from a database, the student should be able to monitor "progress" toward completion. Evaluating the student product should rest with the quality of the completed product, not with the mere fact of completion. Should we ask an architectural student to design a structure, we want to reward him or her for being good, not for being done.

Yet in writing, there are no clear restrictions on what an academic paper need be or do — and thus when it is done. Indeed, as we saw above, an academic paper does not do anything on its own. It rather marks an author's attempt to establish a relationship of leverage in a larger community. Just to understand that this is what an academic paper seeks to accomplish requires a major intellectual achievement on the part of our students. Imagine teaching students how to program computers by saying, "If you can understand (and only understand) what I want your program to do, you will have fulfilled most of the course requirements." In computer programming, the statement is ludicrous. In teaching academic writing, it is less far-fetched than one might think. Certainly, one of the hardest aspects of teaching students to write academic papers is giving them a clear sense of what they are supposed to be writing. As writing teachers, we are often only too happy when we find students writing something that looks (even remotely) like an academic paper and not a personal essay or a high school term paper (relying on cutting and pasting sources with no original thought).

*forward guidance.* When teaching design, it is important to monitor student progress so
that we can guide students forward as they confront difficult decision points. Yet such close and active guidance between teacher and student depends on both "sharing" the same task space. The teacher needs to be able to anticipate the obstacles students will encounter and to provide signposts, hints, clarifications as a way of getting students to avoid them. In math and science, there is a strong presumption that the teacher can, in principle at least, share the student task space. A not uncommon convention is for teachers of math and science to "work out" difficult problems before giving them to students. This allows teachers to draw upon what they did, as problem-solvers, when giving students forward guidance as to what they (the students) might do.

In the teaching of writing, it is almost unheard of for the teacher and the student to share the same task space. For various reasons, some more compelling than others, teachers of writing almost never think to tackle their own writing assignments before giving them to students. (Many, if not most, writing teachers in this country are not themselves people who write.) As a result, writing teachers are often put in the awkward role of not knowing exactly the challenges and traps of the writing they are assigning. Under these circumstances, offering students constructive forward guidance is very difficult.

An interesting consequence of the lack of real forward guidance in writing instruction is that the writing establishment has come up with a host of canned routines or scripts for forward guidance. The establishment jargon for these canned routines is "instruction in pre-writing." Pre-writing is the official word designating everything that happens before the event itself. Instruction in this pre-event consists of exercises for brainstorming ideas and arranging them into patterns. While I have no quarrel with this instruction per se, they are poor substitutes for real forward guidance through a specific writing project. Yet many teachers of writing seem to confuse canned prewriting routines with adaptive instruction in guiding students forward through a specific task.

Backward Guidance. Any environment that teaches design benefits when the teacher can monitor students' progress and send them back to redo poor decisions. In many design tasks, this regular backward movement is called "iterative development." In writing, it is called "revision." To give good backward guidance, a teacher must be able to keep a trace of the decisions that led a student to his or her present state. The teacher must then search this space of decisions and decide which the student should redo.

For reasons that by now should be apparent, teachers of writing have a very difficult time tracing the decisions that the student made — or failed to make — along the way.
Consequently, the teacher will recognize that a rough draft is far from a completed draft and will tell a student to revise. But it is very hard for writing teachers to tell students where in the space of decisions to return and start fresh. The writing teacher's exhortation to revise is often just a blunt pointer backward.

Diagnosis and Assessment. Coupled with forward and backward guidance is the ability to diagnose recurring errors in the student's performance and to evaluate that performance at many intermediate points. In any design activity, the teacher will want the means to diagnose errors and to evaluate at regular intervals. Yet, despite the vigorous efforts of the Educational Testing Service, our understanding of how to diagnose writing problems or to evaluate writing remains quite primitive. As with revision, most of what has been regularized in writing diagnosis and assessment remains at the level of visible language. And even the work on writing assessment that goes beyond the assessment of surface error does not go beyond a "one snapshot" assessment of the finished product. Thus far, no one has devised a good way to implement "multiple snapshot" assessment during the making of a written product.

Writing and Decision-Support: A Computer-Based Solution

I have tried to indicate what principles of design are lacking from the common ways in which we now teach academic writing. So just how do we find a way to accommodate these principles in our teaching of this complex skill? Our answer follows the answer that researchers in other fields have offered when trying to structure for designers what Simon calls "ill-structured" activities: building decision-support systems.

A decision support system is an environment — typically a computer environment — that tries to capture the decisions that a designer needs to make when performing competently at some ill-structured task, that is, at a task that is nonalgorithmic and that relies on heuristics. In the short history of decision-support systems, traditional areas of application have been strategic planning in business management and the building of quantitative models in the social and natural sciences.68

A decision-support system needs to be descriptive in reflecting actual expert practice. But it also needs to be normative, offering "idealized" paths when expert practice gets too complex or goes awry. The main function of a decision-support system, then, is not to model the behavior of individuals on a task. A decision-support system does not simulate individual behavior. Rather, its main function, as its name applies, is to support a designer as he or she engages in a complex task.

Because decision-support systems support work on tasks that are ill-structured, it is
beyond the reach of a decision-support system to determine every micro-step the designer takes. What such systems are intended to do, rather, is to break down a complex and amorphous task into a schedule of well-structured yet flexible goals. They are intended, furthermore, to give designers effective, yet flexible, plans for achieving these goals and even lower-level (and once again, flexible) tools for implementing the plans.

Is it worth trying to make academic writing an area of application for a design-support system? What must one imagine to bring these notions together? To answer these questions, it is useful to contemplate the following: When students first learn to write term papers in high school, they are told to equip their desks with a good dictionary and, perhaps, a thesaurus. For the beginning writer, the addition of a dictionary and thesaurus represents an approximation of the "idealized" desktop for writing. The beginning writer assumes, correctly, that the seasoned writer comes equipped with an internal storehouse of words and synonyms. And for the expert as well as the beginning writer, a good dictionary and thesaurus supplies an external backup when the internal storehouse runs out.

Yet, there is much more to the internal desktop of the expert than simply a dictionary and thesaurus. We can imagine that there is also an internal desktop that helps the expert do productive things when he or she is asked to summarize or synthesize information, or to develop a writing plan that enjoys evidentiary support and carries leverage within a target community. As companions to an external dictionary and thesaurus, a decision-support system for academic writing would simply provide more external aids for these higher intellectual skills of writing. It would extend our notion of what must be part of the expert's desktop for academic writing.

Perhaps the contemplation is working so far. But how much farther can it go? Can one expect to support operations as amorphous as summarizing and synthesizing within a decision-support environment? Can these notions be broken down into structured yet flexible goals, plans, and strategies? It seemed to us that many of the higher intellectual skills of academic writing (summarize, synthesize, developing a plan for writing with support and leverage) reflect not simple words, but complex information-processing environments. When a writing teacher asks a student to summarize or synthesize or develop a writing plan with support and leverage, the teacher is really asking the student to process information in different, albeit complex, ways.

By "externalizing" these representations in a computer environment, we have a natural way of invigorating the teaching of writing by linking it quite explicitly to the teaching of design.
That is, we have a natural way of making the goals we give students more procedural. We have a way of putting helpful bounds on what we want students to do when we ask them to write to academic communities. We have a more principled way of offering both forward and backward guidance through this long and difficult process. And we open up the opportunity of taking "snapshots" for assessment and diagnosis at many junctures, not just with the submission of the end product.

Of course, just because we are successful in externalizing an academic writing environment by computer does not mean we will be successful in improving academic writing. The designer of any decision-support system must ultimately face the question of evaluation. The ultimate goal of any decision-support system, in short, is to improve performance. At the same time, it is well-nigh impossible to test such systems until they are built and in active use. There is, thus, a symbiotic relationship between designing decision support systems for writers and testing them. We should not think of any single system in use as the ultimate cure-all. We should rather think of any one system as an embodiment of ideas — a theory if you will — about how to support writers who are trying to design for a community. We should then look to evaluate our systems in order to evaluate — and improve — our theories as well as the systems they inform.

The Current State of Computers and Writing

Extending decision-support systems to academic writing represents a far different integration of computers and writing than the current integration. Since there is now much "writing" software that is well-known and widely available, it is useful to contrast our approach with the current fashion in supporting writing on computers.

The software now available to support academic writing falls into the traditional categories of pre-writing, writing, and revision. Pre-writing aids, as I indicated earlier, describe any activities that occur before the event of writing itself. As far as computer software is concerned, pre-writing tools include "content-free" software for outlining and arranging ideas. More specific (instructional) software for brainstorming and planning relies on textbook advice that can be transmitted as well (and more cheaply) on paper as by computer.

Writing aids refer to word processing programs of any size or shape. Revision aids refer to grammar drill and practice programs as well as text-evaluation programs like Writer's Workbench. Writer's Workbench, a package written at Bell Labs, is a series of programs that, among other things, hunts down long, passive, and wordy sentences.

Taken together, this collection of software suffers a variety of drawbacks. First, it is
designed to fit into a "linear" model of writing. No conception is provided as to how these programs are meant to influence one another in the actual process of writing. We know that writers do not outline, write, and revise in a straight line. They perform each of these activities interactively. They move backward as well as forward as they write. Yet the current software is designed to encourage writers to think of these processes as moving in one direction only.

Second, this software adds up both to a very incomplete and unconnected model of writing. There is more to planning than brainstorming ideas and putting them on paper. There is more to actual writing than generating characters. There is more to revising than looking for the "long, passive, and wordy." Yet the image provided by this software in the aggregate is that this is the totality of the writing experience.

Third, none of this software offers an image of a writing environment as a learning environment. Writers learn as they go. If writers didn't have to rely on the writing process to teach themselves what to say, writing wouldn't be the challenge we know it to be. Yet the software collectively available as "writing" software affords no notion of how the ideas of writers change and develop during the writing process.

In academic writing, as I discussed earlier, the overriding goal is to figure out how others have organized their knowledge on an issue and then to design fresh organizations for the community. Preparing oneself to acquire a fresh way of seeing things is a sine non qua of academic writing. Yet, the current software for supporting writing is based on a conception of writing as as a "retrieve and polish" process rather than a true learning process. Users are instructed to retrieve what they already know, put this in a coherent format, and then polish. Writing is turned into the formatting of old ideas rather than the discovery of new ones.

Fourth, this software embeds no explicit theory of how writers actually design papers to make contributions to communities. It conceptualizes writing as a relatively goal–free activity (like walking) rather than a goal–directed one (like walking to the store to buy groceries). It misses entirely the representation of writing as a process both continuous, yet focused on adding to or improving knowledge. It fails to situate the the writer in a community trying to learn enough to talk back to it — and to change it for the better.

Given these limitations in the current integration of writing and technology, one should not be surprised to learn that writing teachers, as a group, have been highly resistant to adopting software that is purportedly aimed to support writing. While many writing teachers acknowledge the usefulness of outliners and word processors for their own writing practice and
those of their students, they don't see the current software as theoretically motivated enough to support an actual writing curriculum.

As a final point, let me speculate that, in comparison to textbooks, computers put a much higher burden of proof on information we give to students about writing. As I mentioned earlier, writing textbooks offer advice for indefinite writing projects. They tell writers what may be useful to do at some indefinite time for some indefinite writing task. But information placed on a computer screen conveys a much greater sense of immediacy and intrusiveness. It may be easy to ignore what an author has buried on page 26. It's much harder to ignore advice blinking on a computer screen. Yet it seems to be the very intrusiveness of on-line advice that troubles writing teachers when they look to computers to support writing instruction. How do we actively guide the writing process without stifling the writer? Instructional software conceptualized within the linear model of prewrite–write–revise offers no answer to this question.

The WARRANT Environment: The first Decision–Support System for Academic Writing

In an effort to extend decision–support principles to academic writing, we began, in 1984, work on a computer environment we call WARRANT. Our aim with WARRANT is to provide a prototype environment for supporting academic writing. In order to put reasonable limits on our environment, we have made the following assumptions. First, we have assumed that writers acquire a sense of the community they write for by reading other authors addressing the issue(s) they plan to address. This assumption is too narrow for professional academicians, whose knowledge of an academic community ranges wider than the professional papers they read. But it is a good starting assumption for students, whose sole exposure to an academic community typically comes from assigned readings.

Second, we have assumed that all academic communities acknowledge the plausibility of everyday reasoning as a way of both supporting assertions and acquiring the leverage to say something new. While everyday reasoning is a necessary component for all academic communities, it is not sufficient for most. Most academic communities expect everyday reasoning to be both supplemented with and filtered by formal — mathematical and quantitative — reasoning. In our initial attempts to conceptualize the WARRANT environment, however, we have restricted our focus to academic writing that relies on everyday reasoning. We have, in other words, restricted our problem to academic writing where the writer's goal is to support assertions and seek leverage with a community on the strength of everyday reasoning.
While both these assumptions make our problem more manageable, they are assumptions we can easily relax as we learn more about academic writing. That is, as we learn more about how writers acquire knowledge of the community they are designing for, we can relax the assumption that this knowledge comes solely from reading a specific list of papers. As we learn more about supporting assertions and gaining leverage in the specialized disciplines, we can supplement our environment with tools (statistical packages, theorem provers) for specialized reasoning.

Data Collection

Our biggest obstacle in designing a support environment for academic writing has been to gain a record of the process that is complete enough to be useful. What goes into actually making an academic paper? What decisions must be faced? What is the best way to schedule them without losing desired flexibility? Writing textbooks, as should now be clear, tell a story that is full of holes. To build a computer environment, we have to draw on a much more complete story.

Since we restricted our focus to everyday reasoning and since philosophers often rely on everyday reasoning when writing about ethical issues, we decided to build our story by watching philosophers write. In the Spring of 1985, we asked five graduate-level philosophers to participate in a research project aimed at understanding how they produced an original essay in response to readings. Each participant was given a set of eight articles and was asked to write an original essay on the ethical issue discussed in them: the definition and justification of paternalistic interference. We allowed each participant to complete the task in his or her own way within the time span of about 50 hours of work.

We collected data from these experts in three ways. First, we asked them to verbalize their thoughts out loud into a tape-recorder whenever they worked on the project. In such "think-aloud" protocols, participants are encouraged to say whatever is "on their minds" as they work rather than to "explain" what they are doing retrospectively. Such vocalizations can only provide the tip of the iceberg for the representations that a writer relies upon; but the tip of the iceberg often produces enough information to allow what lies beneath the surface to be modelled. Second, we collected all of the writing the participants produced. And third, we interviewed the participants between working sessions to establish their current conception of the paper. To date, we have completed a preliminary analysis of the protocols, written products and interviews for two of the experts.

To give ourselves an independent basis of comparison, we also asked two freshman
students to read the same articles and write an original essay following the same directions. In addition, we asked an experienced teacher of academic writing to assign the same readings and the same writing project to a class of 25 undergraduate students. We taped half of the teacher's lectures and interviewed the teacher before and after each class as he taught to the assignment. We also interviewed each student in the class twice during the teaching of this assignment. This data gave us a picture of the detailed (and often conflicting) perceptions of the task that the teacher tried to convey and that the students tried to interpret. To date, we have completed a preliminary analysis of the protocols, written products, and interviews of our novice writers.

To repeat, our aim in collecting this data was not to simulate the behavior of any one person writing the assignment. It was rather to get a detailed record of the decision-making that must go into designing an original essay from sources. From our data, we were able to hypothesize a host of decisions that need to be made, the context in which they are made, and the inputs and outputs of these decisions. We took these hypotheses and tried them out in more writing classrooms teaching other issues. We also tried them out in our own academic writing.

The result of this additional testing and data gathering led us to refine our initial hypotheses, to smooth out redundancies and discontinuities, and to aim for greater simplicity in the overall structure of decision-making while trying not to sacrifice important detail. The trick is to give the academic writer the big picture informed by detail but not buried in it. It is to define the challenge for academic writers without taking the challenge away.

WARRANT's Overall Design

At the time this is being written, we have a rough design for a system to support academic writing (under the limiting assumptions we described above). We also have some large components of the system implemented and in actual use in a variety of writing courses. As in many projects of this scope, however, design and implementation often move along independent timelines. In this section, I will discuss the overall design of the system.

Workspaces

The WARRANT system is conceptually partitioned into four workspaces. A workspace is a meaningful and recurring unit of learning that marks "progress" during a writing assignment. The job of a workspace is to move the writer from one structure for organizing information (an input representation) to another structure (an output representation). A workspace is itself made up of a sequence of intermediate goals, plans to reach these goals, and computer tools to
We have given WARRANT's four workspaces rather traditional names: Summarize, Synthesize, Plan, Draft. The names are mnemonic for the activities we usually associate with these terms. But unlike the words themselves, we assign each of these workspaces an explicit interpretation within a computer environment. Thus Summarize in the WARRANT environment, unlike the activity of summarizing in general, is a relatively well-defined activity without the ambiguity of the everyday notion.

Of course, we are betting that Summarize in the WARRANT environment captures much of what teachers want to teach when they teach the everyday, ambiguous, notion. The same holds true for what we call Synthesize, Plan, and Draft. Granted, these are all empirical bets that need to be tested. But more important than our ability to confirm any set of workspaces empirically is the enterprise of making such bets and being willing to test them. It's unlikely we'll make much progress in the teaching of writing if we aren't willing to see how our ill-formed notions are described by notions that are better-formed.

To get an overall idea of how each workspace guides a writer, let's consider the top-level goal of each. The work of the Summarize workspace is to turn a text into a path. When writers first encounter a reading on an issue, they see a text, an amalgam of paragraphs, sentences, and words. This is the input representation to Summarizing. After the work of Summarizing, however, writers should see these structures as forming an author's recommended path through the issue. That is, the Summarizing environment should assist writers in seeing a text as an author's set of directions for going from problem to solution through an issue. Seeing a text as a path through an issue is the output representation to Summarizing.

The work of the Synthesize workspace begins where Summarizing leaves off. In designing an essay from sources, a writer will need to summarize not one but many sources. In the WARRANT environment, this means that the writer will enter the Synthesize workspace after transforming multiple texts into multiple paths through an issue. Like a traveler in a strange land who has been given multiple — sometimes conflicting — directions to a given destination, the writer, having summarized multiple sources, now must travel with different, often conflicting, paths as guide. Adding to the confusion is that each path may be stated in a style and vernacular that makes comparisons difficult.

The work of Synthesizing is to reduce these differences by building a common meta-language for describing similarities and differences across paths. More graphically, in Synthesizing, the writer's chief goal is to construct a map of alternative paths through the
issue. By moving through the Synthesis workspace, writers learn how paths through an issue form alternatives to one another. They learn how paths fork and converge. They reconstruct the decision points that led different authors to take forking or converging paths. All of these decisions are made in the process of the writer's effort to design a map of alternative paths.

The Plan workspace takes the results of Synthesizing — the map of alternative paths — as its input representation. For its output, the Plan workspace guides writers to design a path of their own (from problem to solution) through the issue. The Plan workspace, in other words, helps writers develop a writing plan. In the WARRANT environment, writers design their own path through an issue by testing the strengths and weaknesses of paths charted by other authors. Since a path through an issue connects problems and solutions, writers searching for a writing plan can focus on aspects of another author's path dealing with defining the problem at the heart of the issue as well as aspects of the path concerned with solving the problem.

In the WARRANT environment, writers test the paths of other authors by understanding that a problem path and a solution path have specific functions to perform with respect to problem cases or situations associated with the issue. Every issue is associated with a list of concrete cases or situations that a community finds problematic. The list varies from author to author, but there must be a significant overlap of problem cases across authors for there to be a common issue.

Authors who address the issue propose problem paths in an attempt to characterize these problem cases generally. They propose solution paths in an effort to remove or ameliorate these cases. In the WARRANT environment, writers are guided to test the problem paths of other authors by forming their own list of problem cases and seeing how well the paths of other authors characterize them. They are guided to test solution paths by seeing how well other solution paths handle the problem cases on their list.

From our data collection, we isolated a strategy that goes far in insuring the writer a path through an issue that bears the writer's own signature. We call this strategy "looking for unaccounted-for problem cases." An unaccounted-for problem case is a problem case that previous problem or solution paths have missed. Every previous author has failed to put his or her finger squarely on this case. Every previous author has failed to devise a solution that will effect this case for the better. When a writer finds an unaccounted-for case (or cases), he or she can use that case as the basis for designing new problem and solution paths that will set him or her apart from the community. Unaccounted-for cases become an axis for
rhetorical leverage, for saying something new. The Plan workspace should leave writers with a new path through the issue.

The work of the Draft workspace is the inverse of Summarizing. While Summarizing helps the writer transform a stand-alone text into a conceptual path, Drafting helps the writer transform his or her newly discovered path into a stand-alone text. Writers face many difficulties when they try to install a plan on the page. Drafting often forces the writer to expose him or herself to details that require expanding the plan or even abandoning parts of it. In addition, writers need to smooth out the path to make it easier for readers to follow. They also need to apply close editing principles to make their text easy for a reader to process. The Drafting workspace guides writers through these processes as well.

While Summarizing, Synthesizing, Planning, and Drafting represent logically sequential workspaces, writers (or teachers of writing) can take the initiative in how they work (or how they direct students to work) through them. Each workspace culminates in a paper that reflects the progress the writer has made to date. Working through the Summary workspace, the writer gets directions on how to write a summary; working through the Synthesis workspace, a written synthesis; working through the Plan workspace, a written proposal for an essay; working through the Draft workspace, a polished essay. A teacher can check a student’s progress by examining the paper the student writes at the end of each workspace. If the teacher is satisfied with the results, the student can be guided forward; if the teacher is not satisfied, the student can be guided backward. By partitioning a writing task by workspaces, teachers can more carefully provide both forward and backward guidance.

Many teachers in the specialized disciplines do not expect their undergraduates to offer original paths in their writing — and for good reason. It can take a long time just for undergraduates to learn to read, summarize, or synthesize professional articles. The WARRANT environment accommodates this inasmuch as teachers can decide _just_ to have students work up to the summary or synthesis of articles without going further on their own. Significantly, though, students using the WARRANT environment just to summarize or synthesize will be learning methods for summary and synthesis that _will_ support more advanced work when students are ready for it.

Libraries

So far I have only discussed the very top-level function of each workspace. Associated with each workspace is a library. Thus the Summary workspace has a Summary library, the Synthesis workspace, a Synthesis library, and so on. If the purpose of a workspace is to help writers make progress, the purpose of a library is to store the intermediate results of the
writer’s progress any time he or she makes use of that workspace. For example, during the course of a single writing project, a writer will make numerous runs through every workspace. The libraries associated with each workspace can save the intermediate results of each run. For example, a student may design many different syntheses (maps of alternative paths) of readings. The student can store all of these designs in the synthesis library. The student can then search this library and choose the synthesis he or she wants to build on (as input for the Plan workspace).

Libraries are important because writers need the freedom to experiment with many different summaries, syntheses, writing plans, and drafts. Libraries allow writers to experiment without losing their work. They also allow later workspaces to retrieve and build on the results produced in prior workspaces.

Goal Sequences

So far we have been talking as if the only intermediate results stored in a library was a written text — either a summary, a synthesis, a proposal or draft. In fact, a workspace is itself partitioned into a sequence of goals. Each goal represents a small transformation within the larger transformation covering the workspace. Thus, if the entire Summarize workspace is designed to help writers transform a text into a path, a single goal within the Summarize workspace would be to assist a writer in carrying out part of this transformation.

Writers create an intermediate product with every goal they try to satisfy. Thus the library associated with each workspace contains not only one intermediate product (a written summary), but also an intermediate product connected with each goal in the sequence of goals that makes up Summarizing. As writers use WARRANT then, they are creating a trail of intermediate products, any or all of which can be used for diagnosis and assessment.

Plans and Tools

As writers work on a particular goal within a workspace, they are assisted by what we call plans and tools. Plans are a combination of verbal and graphical descriptions that explain to writers, in more operational terms, how they can achieve the goal. Tools are computer programs built to help writers carry out the suggested plan. Often one and the same tool can help carry out different plans. In addition, some plans may rely on multiple tools.

To be more concrete, let me briefly sketch how goal sequences, plans, and tools interact in part of the Summary workspace. As we mentioned above, the overall goal of the Summary workspace is to help writers transform a text in a path. This transformation is quite complex and can be broken down into a sequence of smaller goals. In the Summary workspace, we call
this sequence:

1. Reading for the Summary
2. Taking Notes
3. Writing the Summary

In reading for the summary, writers learn to organize a text into its path-like characteristics. In taking notes, they take notes on their sources according to these characteristics. In writing the summary, they learn how to organize their notes linearly into a textual summary.

Each one of these goals is broken down into more operational plans. Let's consider plans for reading for the summary:

1. Listening for different voices
2. Molding Voices into a connected path
3. Breaking a path into milestones
4. Following the detail along the way

Each of these plans represents a progressively more detailed way of representing a text as a path. To work on the first plan — listening for different voices — students need to be aware that authors will not only give directions to help readers follow their path through an issue; they will also characterize the paths of previous authors which they believe in some way faulty. They will mention faulty paths alongside the path they actually recommend for a couple of reasons. First, they may want to show readers they are familiar with these paths. Second, they may fear that, without explicitly distancing themselves from these paths, readers may confuse the path they recommend with certain faulty ones. In either case, an author's desire to include faulty as well as a recommended path through an issue means that the author's text will incorporate multiple voices. An early discrimination in summarizing is recognizing these voices and keeping them distinct.

To work on the second plan — molding voices into a connected path — students need to be aware that authors do not simply offer competing voices in parallel. They rather express their dissatisfaction with the faulty paths and show why this dissatisfaction results in staying on course with the recommended path. They thus try to link up their own path through the issue with faulty paths. The faulty paths are now seen as unhelpful "detours" that provide further incentive to the reader to stay on the chosen path.
To work on the third plan — breaking a path into milestones — students need to be aware that authors may present their path through an issue not simply as a single set of detours leading to a chosen path, but as multiple detour/chosen-path pairings. More specifically, an author's path through an issue travels through important milestones, such as seeing the issue, defining the problem, and solving the problem. Authors see an issue through problem cases. They define the problem at the heart of an issue by trying to generalize across these cases. They solve the problem by trying to resolve these problem cases.

Each of these milestones represents the locus of a potential detour/chosen path pairing. That is, authors may see other authors as going astray because they failed to recognize certain problem cases. Or they may see other authors as recognizing the right problem cases but going astray when trying to give these cases a general characterization. Or they may see other authors providing the right characterization, but going astray when it comes to solutions. A more sophisticated tracking of an author's path through an issue will include the knowledge of these milestones along the way.

To work on the fourth plan — following the detail along the way — students need to be aware that authors often use low-level textual acts to help readers follow the fine detail along his or her path. When authors anticipate that the points along their path are too sparse for the reader to follow, authors will amplify. When they anticipate their points will not be accepted on their face, they provide support. When they anticipate the points along their faulty paths have attractive elements or the points along their chosen path, unattractive ones, they will concede. When they anticipate their points will trigger false or unwanted inferences, they disclaim. And when they anticipate their points claim more than is defensible, they will qualify. At the lowest level of detail, writers should be able to discriminate this fine grain of text act.

Each of these plans is supported by on-line tools. In the case of reading for the summary, we have proposed a reading discrimination program which allows a writer to annotate a text according to its path-like characteristics. The tool would also make it easy for writers to compare their annotations with those of other readers. This tool would be helpful to a writer working progressively through these increasingly sophisticated plans for reading a text as a path.

As one might guess, the major conceptual effort for WARRANT has come in designing workspaces, libraries, goal sequences, plans, and on-line tools. The major implementation effort resides in integrating these notions in a single environment and in building robust on-line tools. At this time, we have a robust note-taking system in full operation, designed chiefly by Chris
Neuwirth. Our note-taking system (called Notes) is now supporting a variety of writing courses at Carnegie Mellon.

A standard for all the on-line tools we develop is that they be robust enough to support writers within the WARRANT environment or as a stand-alone writing tool. Consequently, outside the WARRANT environment, Notes is being used for a variety of instructional applications. But inside that environment, is has been designed as an important on-line tool supporting the Summary workspace. Adhering to the standard of robustness allows us to produce useful software long before the WARRANT environment is fully realized.

Space prevents me from outlining the WARRANT environment in more detail. We believe this environment is important not because it, alone, can revolutionalize the way we think about and teach academic writing, but because it gives us an important framework for understanding how the revolution can take place — at least in the melding of writing, design, and technology. WARRANT is important not as an announcement of what has been but as a harbinger of what can and will be.

Conclusion

As soon as we acknowledge that academic writing is more than scribing, more than formatting according to convention, we find it impossible to draw a sharp line between writing and the whole of the educational process. Both academic writing and education can be aptly characterized as "learning to think well and productively in a community." Both can be described as "designing for the community." Yet the notion of writing as design is so broad and flexible that, I fear, we seldom take the relationship as seriously as we should.

Few will blink at the thought that writing is a form of design, no less than art or architecture or engineering. But when we look at how writing has been taught, we see that the principles by which we would reasonably teach any design process have given way to romantic assumptions of writing as mystery. Many well-meaning humanists (some trained in science, some not) see no harm in probing the mysteries of the brain. But they balk at probing the mysteries of writing as somehow "dehumanizing" — as if our human spirit were captured in the way we write.

The inconsistency in this position would be laughable were it also harmless. Unfortunately, it is not. Keeping writing a secret both from ourselves and our students means that we learned to talk to our peers in spite of what we were (or, more precisely, weren't) taught. It means we learned how to write in an unsystematic fashion and so can't be systematic when trying to convey to students what we know. It means we will perpetrate the
uncomfortable hypocrisy of pronouncing writing "essential" while pronouncing writing courses "useless." Metaphors can be so powerful that we are foolish not to take them as more than that. "Writing is a form of design" is a metaphor whose time has come.
NOTES

66Research and development efforts reported in this article were funded by a grant from the Fund for the Improvement of PostSecondary Education, U. S. Department of Education, to the author, Preston K. Covey, Christine Neuwirth, and Cheryl Geisler, now at Rensselaer Polytechnic Institute.

67For better or worse, writing teachers have built their authority on the "split infinitive" and "dangling participle." We have a tool-kit of procedural information about how to avoid these surface patterns.

68For a discussion and references, see John L. Bennett Building Decision-Support Systems Reading Massachusetts: Addison-Wesley, 1983.

When an engineer creates a plan for an aircraft, for a robot to fix it, or for a computer system to land it, we say that person is carrying out the design process. When that same engineer produces a technical analysis, a research proposal, or a public policy statement about that same system, he or she may be going through a closely related intellectual process. In rhetoric the act of creating a piece of discourse wouldn’t be called design, it would be called the writer’s problem-solving process. And the emphasis might be less on the evolving product and more directly focused on the writer’s cognition. But these two intellectual acts—design and rhetorical problem-solving—share some basic features:

- Both are concerned not only with analysis and understanding but with synthesis and production.
- In this emphasis on the process of making, they take a rival position to the dominant teaching traditions in their respective fields.
- Finally, they share a common CMU heritage and, I believe, some common principles that give a unique coherence to the Carnegie-Mellon educational core. At the same time the procedures that support rhetorical problem solving are typically less well defined than those in the sciences, the range of alternative "successful" solutions is often enormous, and the process itself if often perceived as an act of intuition. As a result it can be difficult for student to recognize the strategic nature of their own process as writers and to learn to control that process.

In this paper I would like to consider what an "expert system" in writing might include—what are some chief features of the expert writer’s problem-solving process that resemble the design process of analysis and synthesis in other areas? And—clearly to the point of this volume—can such a process be taught beyond an elementary level?

HOW DID WE GET HERE: A Look at Recent History

Elsewhere in this volume Herbert Simon describes a split that has occurred in professional schools between teaching which is focused on analysis or description, on the one hand, versus production and synthesis on the other. This split is equally apparent in the humanities. In fact, it is part of a minor revolution that has changed the teaching of writing. And it helps explain the important role CMU has played in the recent history of rhetoric in the United States. In 1986 the Rhetoric Program at CMU had only been in existence for five years. Yet in that time it had been cited as one of the top two graduate programs in rhetoric in the country. And in that same year CMU and its partner, the University of California, Berkeley, had been chosen by the Department of Education as the national Center for the Study of
Writing. An obvious question is, how did a small and relatively new program enter the national picture in such a short time?

One reason lies in the CMU tradition of problem-solving and design. The Rhetoric program at Carnegie Mellon is seen as a pioneer in research and education in part because it brought the philosophy of the "design process" and the methods for studying it to the study of writing—and the two turned out to be a powerful new combination.

Rhetoric is an old discipline. Aristotle is usually credited as its major curriculum developer. However, when rhetoric was divorced from the study of logic in the late 17th century, it began a slow decline into the art of speech making. In the more recent past, say 1950, college composition courses reflected the heritage of this intellectually impoverished view of rhetoric. Direct instruction in composition focused on correctness—learning the rules of grammar, spelling etc.—and on style—learning to have the good taste not to split your infinitives (a holdover from Latin grammar). As in the analytical model Simon described, this cookbook view of execution was linked with analysis and description. Students were encouraged to read good models and taught to analyze literary texts as the basis for their own prose writing. In most English departments, the discipline was defined as the study of literature. Writing, identified with style and correct form, was something students "should have learned in high school." The intellectually difficult skills of writing, such as framing an argument, organizing or re-organizing one's own knowledge for a reader—these skills students were left to acquire on their own.

The revolution in rhetoric and writing has occurred on two fronts. One has been a burst of research on how people actually write: research on the function writing serves in schools, professions and business; research into the assumptions and strategies students use as they enter an academic or professional discourse community; and finally research into the underlying cognition or thinking process such writers go through. As this research began to uncover the writer's intellectual activity, it became clear that success in writing depended on how a person managed this thinking process. Knowing the rules of grammar and style or even knowing a topic well simply wasn't enough to make an effective writer.

At the same time, teaching of writing began to undergo a parallel revolution that has been called the "process" movement. CMU, which had been the leader in the cognitive research in writing mentioned above is also the place most strongly identified with a special approach to teaching, that is, with teaching writing as a problem-solving process. It is easy to see the history of "design" at CMU behind this plan. Teaching is based on understanding
writing as a cognitive act and on uncovering the thinking strategies that give experts some of their power. The focus is not on the analysis of "good writing," but on synthesis and production—on teaching students to control those processes for themselves and become effective makers. Many parts of this approach are shared with other parts of the university.

In the rest of this paper, I will look at the unique form this problem-solving approach takes when it is wedded with the classical discipline of rhetoric and the practical goals of teaching writers to communicate. (Cf. The Rhetoric Program: A Strategic 5-Year Plan).

Ideas One Didn’t Expect to Learn

Students have many preconceptions about what they will learn in college English—they often define writing in terms of correct spelling, good grammar, and formulas for format. Unfortunately, that sort of rule-governed knowledge would be rather inadequate in the face of thinking tasks writers actually encounter. The act of writing a paper at CMU may call on a student to plan an argument, to systematically search her knowledge, to connect ideas and build new concepts, to organize and adapt information to a new purpose and to review, evaluate and revise a text in light of how a reader would respond. This gap between high school expectations and reality may affect freshman courses in other areas: students may come expecting to learn facts in history and formulas in physics but they discover that their courses have a different agenda. College-level introductory courses are designed to help students learn to think like a historian and analyze situations like a physicist. Writing is the same. We are after what one might think of as the "university-level" intellectual skills that experienced and effective writers possess.

This approach to writing violates some expectations. And it also raises the standards—it treats writing as the hard task it is for professional people who typically write in order to make something happen. At the same time, it increases the chances that what we teach will transfer to other courses and beyond. In this chapter, rather than describe a curriculum, I would like to focus on five ideas that come as a surprise to many students—ideas which they tell us have made a difference to them. And then I will conclude with some of the principles that stand behind this problem-solving approach to writing.

1. Writing is a thinking process—which I can manage.

Writing is an absorbing activity. When all goes well it seems as though writing is simply the act of transcribing thoughts. When it doesn’t go smoothly, the preoccupation with getting words on paper soaks up one’s attention so thoroughly that people rarely notice their own process. They may not see that an important part of the problem is how they are handling the task itself. The first step in managing a difficult thinking process like writing is just to recognize that it is there—that willy nilly, one is using strategies and making decisions about
Below is a transcript of a writer thinking aloud as she works. These thinking aloud protocols, as they are called, are widely used in problem-solving research because they give us a window on the thinking process as it operates in real time. This protocol excerpt lets us see some of the goals a writer is setting for herself, the strategies she thinks are appropriate, and some of the problems she encounters. Designed as a teaching tool, this protocol is in fact a composite which allows us to demonstrate some effective alternatives to the writer’s initial strategies.

INSERT PROTOCOL HERE

This transcription of a writer’s thoughts demonstrates a number of composing strategies, some of which often cause trouble for writers.

1. **A trial and error approach to producing sentences.** Notice the almost random way this writer tries to combine words and phrases in the hope that one version will finally sound acceptable. Her first major thought goes through six alternative phrasings:

   - in today’s world
   - for today’s student
   - in today’s high-pressure education
   - a matter of
   - a question of
   - a problem of

   This trial and error approach to producing sentences often produces more confusion than prose. In trying to juggle a number of alternative phrasings the writer is likely to overload her "short-term" memory. The capacity of human short-term memory—what we might call our conscious attention span—is rather limited. That is why people often fail at tasks such as trying to listen to two conversations and think of something to say at the same time. Because we can consider only a few separate elements at one time (some say the limit is seven elements, plus or minus two), we are unable to simultaneously consider all the alternative versions of a sentence and make an efficient choice, so we keep reviewing the options. This writer’s performance is also characteristic of trial and error in that she often loses track of her search and continues to reproduce previously rejected versions. A trial and error strategy is not only confusing, it’s slow. For this one sentence alone there are 104 possible legitimate combinations and grammatical transformations she might conceivably try. And in trial and error searches, writers typically forget paths they have already taken and try them again.

2. **The Perfect Draft strategy.** This writer’s trial and error attempt at sentence generation is probably familiar to all of us. The real problem with this procedure is that it is part of a larger strategy—the one pass, perfect draft strategy. Looking at the first paragraph
as a whole, we see that instead of planning, jotting notes, or defining her goals, the writer has started out by trying to produce a perfect set of sentences—a final text which will not need revising. As inefficient at this strategy is, many writers depend on it and spend hours trying to perfect their first paragraph or first page. Once they have sweated out that first paragraph, the rest of the paper does indeed seem to come more easily. But why? It is because, in the act of writing those introductory sentence, they have also been planning the point and the organization of the entire paper. As if planning one's ideas wasn't hard enough, these writers are also trying to produce perfect sentences, create just the right tone, and provide for smooth connections between the points they have yet to articulate. All of these things must eventually be done, but a perfect-draft strategy tries to do all of them at once. One result is that neither the planning and idea generation nor the sentence crafting is done as well as it could be if the writer managed the process differently.

3. The WIRM1 strategy. An interesting contrast to the perfect draft method is a strategy we have often observed in the protocols of experienced writers. When they felt themselves getting bogged down in perfecting sentences, these writers would say some variation on the phrase "What I Really Mean Is..."(WIRM1) and switch from writing prose to simply "talking to themselves." Interestingly enough, these switches from formal prose to a conversational style often produced very effective sentences, as well as a clear statement of the point.

Protocols like this play an important role in teaching. When students can listen to the tape of a writer thinking and have luxury of observing and analyzing a familiar experience from this distance, it makes two points clear. One is that writing is indeed a thinking process—it is a problem-solving event that happens in real time. The second is that writers have choices and that some strategies are likely to be a much smarter choice than others.

In the freshman course we ask students to do a self-analysis of their own writing process—of the strategies they use and the goals they set for themselves—and to diagnose some of their own problems and the solutions they might want to try. The main goal of this exercise is to focus students' attention on their own thinking. However, in 1985 we collected these analyses from 100 freshmen and a group of 30 juniors, seniors and graduate students and found an interesting difference on three of the questions.

Question 1: On the average, how long does it take you to write (drafting, writing, and revising) one page of fairly difficult material (expository paper, report etc.)?

Allowing for all the vagaries of self-reporting, we find that over half the freshmen see
this as an hour to hour and a half task, whereas 60% of the more experienced writers plan in
putting in three to four hours. Freshmen received the results of this survey with some
surprise. Since the experienced students are likely to be, if anything more efficient, the
difference probably lies in students’ image of the task and in the time they allow for planning
and revising. The actual distribution was as follows:

<table>
<thead>
<tr>
<th></th>
<th>FRESHMEN</th>
<th>JUNIOR, SENIORS &amp; GRADUATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than one hour</td>
<td>8%</td>
<td>--</td>
</tr>
<tr>
<td>1 hour</td>
<td>25%</td>
<td>--</td>
</tr>
<tr>
<td>1 1/2 hour</td>
<td>25%</td>
<td>15%</td>
</tr>
<tr>
<td>2 hours</td>
<td>25%</td>
<td>--</td>
</tr>
<tr>
<td>3-4 hours</td>
<td>10%</td>
<td>60%</td>
</tr>
<tr>
<td>4 hours</td>
<td>5%</td>
<td>--</td>
</tr>
<tr>
<td>5-6 hours</td>
<td>--</td>
<td>25%</td>
</tr>
</tbody>
</table>

**Question 2:** When you think of having to write a paper, what are the main
thoughts that come to mind? (This was an open-ended question, so the responses
represent our categorization.)

<table>
<thead>
<tr>
<th></th>
<th>FRESHMEN</th>
<th>JUNIORS, SENIORS &amp; GRADUATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative thoughts</td>
<td>30%</td>
<td>Time constraints 30%</td>
</tr>
<tr>
<td>Subject matter</td>
<td>30%</td>
<td>Outlines, organizing 30%</td>
</tr>
<tr>
<td>Time constraints</td>
<td>25%</td>
<td>Actual steps such as</td>
</tr>
<tr>
<td>Research</td>
<td>7%</td>
<td>planning, revising 30%</td>
</tr>
<tr>
<td>Format, grammar</td>
<td>7%</td>
<td>Research, subject matter 25%</td>
</tr>
<tr>
<td>Actual steps to take</td>
<td>3%</td>
<td>Negative thoughts 18%</td>
</tr>
</tbody>
</table>

(Figures represent per cent of students who cited each topic)

The interesting difference here was that the more experienced writers thought of writing
in terms of procedures such as planning and organizing and in terms of time constraints they
needed to work around. For them, writing appears to be represented as a manageable process.
For the freshmen, writing is identified with subject matter and, unfortunately, a grab bag of
"negative thoughts." One possible reason for those negative thoughts turned up in the next
question.

**Question 3.** Define the major writing problems you have and list steps you might
take to solve them.

There were a number of problems, such as procrastination, which both groups saw as real
but "manageable" problems—that is, they could think of ways to tackle the problem.
However, the freshmen saw many more of their problems as "unmanageable" ones—that is,
they cited a problem but were unable to suggest ways they might try to solve it. Not only
did the freshmen cite more of these unmanageable problems, the majority of their responses
dwelt on such problems. The table below lists the percent of students who cited these
unmanageable problems.
<table>
<thead>
<tr>
<th></th>
<th>FRESHMEN</th>
<th>JUNIORS, SENIORS &amp; GRADUATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressing ideas clearly</td>
<td>60%</td>
<td>30%</td>
</tr>
<tr>
<td>Inability to organize ideas</td>
<td>50%</td>
<td>18%</td>
</tr>
<tr>
<td>Inability to come up with a thesis</td>
<td>26%</td>
<td>18%</td>
</tr>
<tr>
<td>Understanding what the audience expects</td>
<td>20%</td>
<td>6%</td>
</tr>
</tbody>
</table>

It is one thing to face a difficult problem-solving task; it is another to know that you don't know what to do. Making writing a "manageable" problem is an important part of the freshman writing program. Creating an effective text may not be easy, but it needn't be an absolute mystery either. It may be that the most important thing students learn in freshman writing is that a thinking process, governed by their own goals, decisions, and strategies, lurks beneath the surface of composing. And that uncovering that process can give them power they didn't have before.

2. **Representing to yourself** the task you intend to solve is a critical first step in writing.

Writing belongs to a class of problems that are often called "ill-defined" problems. That is, defining one's goals, one's strategies, and the appropriate tests or criteria is itself an important part of most academic and professional writing. Even assigned tasks, such as "analyze the effect of X on Y," only offer a general plan for action—not a well-defined, ready-to-solve problem. An assignment may define a topic, but it doesn't specify such things as: what constitutes a "good analysis," what does the reader need to know, how should you go about generating relevant information, how should you present it? Because writing task typically present such open-ended problems, representing the task to oneself is critical, yet it is a critical decision many students don't even realize they are making. And an unexamined process is not open to choice.

There are also important differences in how experts and novices in many domains represent problems to themselves. For example, in physics Larkin found that students typically represent a word problem to themselves in terms of concrete objects—cats and coffee cups—and in terms of formulas—trying the various ones they know on for size, to see if it will fit the entities in the problem (1981). The experts, by contrast, do not leap immediately to formulas. Rather, they see such problems in terms of scientific principles. They build a mental representation that turns leaping cats and crashing coffee cups into interactions between force, mass, acceleration, friction—in essence, they build a representation based on scientific concepts. In physics, the difference between the student's naive and the physicist's scientific representation helps account for how they then go about solving the problem and for why novice attempts may fail.
In writing, this phenomenon of task representation plays an equally large, but often unrecognized role. As a part of the research at the new national Center for the Study of Writing at Berkeley and Carnegie–Mellon we are discovering some differences in how Carnegie–Mellon students approach tasks that involve reading in order to write.

Imagine that you as a student were given the following task: "Here are a number of articles on the topic of time management. Read these texts and, using all the relevant information, write your own statement about the topic of time management." The readings range from Lakein's popular Get Control of Your Time and Your Life, to William James' claims about the value of persistence and pushing through fatigue, to Pauk's research on study skills. This task was designed to be a typically open-ended college assignment. It asked students to use both their own thinking and the text. And, as is typical in such reading, there are buried conflicts and experts who don't agree. What would you do?

We have used this assignment in both freshmen and advanced classes, because it gives students a window onto how they are managing their own processes of reading-to-write. The students are asked to collect data on their own writing and thinking strategies as they do this task, and to come to class prepared to discuss their own process. We have found that students at CMU do some dramatically different things with this assignment—but that everyone typically comes to class thinking they have done the obvious or correct task. Moreover, we have found that teachers disagree on this task as well. For our purpose here, I will simply discuss the four dominant organizing plans we see students using. Notice what a large impact these decisions would have on how successful this paper would be in your course—or as a report done on the job.

1. SUMMARIZE

One response to this task is to summarize what you have read. I will use one student, Martha, and her self-analysis as an example. She was a junior engineering student—an efficiency expert. She knew exactly what to do on such an assignment. "You use a "gist and list" strategy to read through the material. Find the key words. Summarize and write your paper as you go along. If a new idea occurs to you half way through," she said, "drop it because it will only confuse the paper and you. This," she said, "is how you write a research paper." It was a well-honed strategy. When we gave this assignment in four sections of a current freshmen course, we found that the Summarize plan was used by 43% of the freshmen in the class.

Martha was genuinely surprised when the next student in her class described a strategy in
which even the topic of the paper was not determined the reading material. According to Kate, an economics student, the topic is a function of the audience, so you must spend time inventing what you need to say. This was such a standard strategy for Kate that she even had a formula for it: \( T = f(A) \). Clearly these two women had very different images of the task before them. (I will return to Kate later).

2. RESPOND TO THE TOPIC

In sharp contrast to Summarizing as an organizing plan, some students were not inconvenienced by the assigned text, because they chose to talk about things they already knew. That is, they used their prior knowledge to comment on the topic of "time management" or pursued a related issue that it suggested to them. This plan for the task can produce some excellent papers that are well organized, interesting, and motivating for the writer. However, this plan simplifies the task of reading to write. Just as the SUMMARIZE plan eliminates the need to use one's own knowledge, the RESPOND plan eliminates the need to use or even understand the text. In our current class, 18% of the students saw the task as an invitation to RESPOND.

At this point in the analysis—and the class discussion—a vision of the costs and benefits of any given task representation begins to emerge. Each of these organizing plans has certain strengths—each will work well for certain tasks. But if a professor expects to see you thinking with the material, or if a supervisor wants an analysis and recommendation, the economy and security of the SUMMARIZE plan may be a false economy.

3. REORGANIZE AROUND A SYNTHESIZING CONCEPT.

Some writers added another set of goals to this task: they wanted to integrate the readings around a powerful synthesizing idea. To do this meant turning to a whole new set of strategies for digging out organizing ideas, even when the texts appeared to talk at cross purposes, and for forging and testing synthesizing concepts.

Synthesizing in this sense is a valuable plan, but it has real costs: What if you don't see an adequate concept in the material? What if the experts disagree? You may have to invent a new conceptual structure yourself. As one reluctant synthesizer said in his thinking-aloud protocol as he reread the assignment, "Interpret and synthesize". What the hell does that mean? Synthesize means to pull together, no to make something up. Why would I want to make something up?" And he didn't. In our current class 25% of the students attempted to use this more intellectually costly and potentially more significant organizing plan. Some of
they succeeded in using it well; some didn't.

4. INTERPRET FOR A PURPOSE

Some few students (11%) gave themselves an even more complex task in this assignment. They chose to use their reading—combined with their own knowledge—for a purpose. They typically had an audience in mind and they chose to interpret what they read in light of a purpose of their own. Kate, for example, had tutored other students and decided to use the material to write something on time management that would be useful to her peers. Hiroshi gave himself the task of critiquing these theorists to see if their ideas were sensible in light of the realities of student life. (He decided that many of them weren't.)

Interpreting one's reading for a purpose is a much harder and far riskier task. For one thing, it is easier to be "wrong" when you interpret than when you merely summarize. And the purpose you give yourself may not be one your instructor or reader finds appropriate. On the other hand, this is the task that experienced writers give themselves, that we as academics use in reading and reporting our own research, and that people in business and the professions use in writing analytical reports and proposals.

For many of our students, this range of options in the face of a standard reading-to-write task comes as news. Able students like Martha found that they were giving themselves limited visions of what a writing task could entail—and that they were not using school as an opportunity to learn more demanding tasks they they might be expected to do in their careers. For other students, in fact for most, the value of this experiment lies in forcing them to realize that writing is a rhetorical action. Any decision they make is going to have costs and benefits—there is no simple "right" way to handle most adult writing tasks. Looking at their own thinking processes and at the goals they give themselves in terms of costs and benefits is a new perspective for many writers. This is especially true for students who come to college seeing writing as a matter of talent and inspiration or seeing successful writing in terms of correct grammar, format, and style.

This focus on analyzing and controlling one's own writing process is one of the things students typically cite as the most important thing they learn in courses such as Reading-to-Write or Strategies for Writing. But task representation is still only part of the picture. The other part is skill. Part of the cost of giving yourself a harder task is that you may not control the skills it takes to do it well. You may come to college knowing how to do an "A" summary, for instance, but a "B-" synthesis. Task representation, then, leads us to the question of strategies themselves.
3. Some composing strategies are more powerful than others.

One goal of a problem-solving curriculum is to give writers more understanding of and control over their own cognition. The other is to teach new and more powerful strategies for doing significant tasks. This is another reason our approach to teaching writing at CMU is so closely tied to research in writing and cognition. We want to have a curriculum that makes expert strategies more directly available to students. Practice and trial and error can be a good teacher—but a rather slow one. Our goal is to speed this process up by teaching what we call strategic knowledge in writing.

In the new Berkeley/Carnegie-Mellon Center for the Study of Writing, part of the research agenda is focused strategic knowledge itself—on understanding what information both experts and novice use and on finding better ways to teach it. Strategic knowledge—unlike the facts and formulas students typically learn from texts—is knowledge about how to perform an intellectual task. More specifically, having strategic knowledge about a writing task means having knowledge of appropriate goals for that task, having control of strategies or procedures you need, and finally, being able to monitor and test your own process.

Let's take a case in point. Teachers advise students to use a "draft and revise" method to get better papers. However, when experts and novices translate this maxim into action—when we see their strategic knowledge for how to do revision—we see some strikingly different processes. Here is an overview culled from some of the recent research. [For a more detailed analysis and references see Flower et al., 1986.]

Some Features of a Novice "Draft & Revise" Process
1. Doesn't see revision as a high-priority goal.
2. As a result, doesn't plan time to do more than a first-time-final draft of the paper.
3. If returning to a draft, will begin to revise immediately, before fully rereading the text.
4. If revising, tends to define the task as proofreading for errors and rarely makes changes in meaning.
5. If a global problem is detected, is likely to ignore it, to plan to return to it but not do so, or to turn it into a more manageable, local problem (e.g., a problem with this sentence rather than the logic of the paper).
6. Is likely to depend heavily on a strategy of simply rewriting from scratch rather than diagnosing and revising. This can lead to a fresh vision and improved paper, but also leads to many papers which are merely different, not better, and sometimes worse.

Some Features of an Expert "Draft & Revise" Process
1. Sees (and plans for) a draft as a step in an integrated process. Can partition the task into distinctive sub-processes such as planning, organizing etc., while envisioning the whole. This strategy also appears to boost
2. Delays perfecting or even composing an introduction until the material (to be introduced) has been articulated in a draft.
3. Rereads draft in an effort to extract its gist and to evaluate not just the text, but the underlying plan.
4. May give top priority to refining the problem or issue which the paper addresses.
5. In moving from 1st to 2nd draft, may now add the audience as a major constraint and use these goals to test the text.
6. May try to carry out #5 either by passing the paper around to readers (an overt procedure) or by trying to simulate a reader's response (a more difficult to acquire, but valuable, mental procedure).
7. If global problems are detected, will attempt to diagnose them; may consider alternative strategies and may initiate an extended problem-solving effort that is markedly absent from the novice's process.

Even at this level of analysis, the difference in the strategic knowledge experts and novices bring to writing is striking. The expert calls on goals and strategies that the novice never considers. And the expert has learned how to carry out educated processes such as diagnosis.

Writers, it is clear, depend on strategic knowledge. Schools and textbooks, however, are more comfortable teaching declarative knowledge about a topic, whether it is engineering or writing, than teaching how to perform as an engineer or writer. Teaching performance knowledge and problem-solving processes may require a new look and serious research into how experts do perform. Here is a case in point: Experienced writers, including teachers, often have a wealth of strategic knowledge that they don't teach and may not be fully aware of. Brown and Day at the University of Illinois found that when experienced graduate Teaching Assistants in English were asked how to "summarize," they offered only vague maxims on the order of "be concise" and "showed a surprising lack of evidence that they knew any effective rules for summarization" (1983). Yet when these researchers examined the TA's "long and discursive" thinking- aloud protocols they found these teachers "made frequent mention of basic rules" and clearly possessed a wealth of quite effective and explicit strategies for the act of summarizing. This looks like good news for the TAs. However, the freshmen students Brown and Day also studied did not control these strategies, and we can presume their Teaching Assistants were not teaching them.

One of the exciting new frontiers in writing research at CMU lies here in developing prototype methods for teaching strategic knowledge and problem-solving processes more directly.
4. Strategic knowledge in writing transfers across departments and disciplines.

As a published writer in your own discipline, you have a great deal of specialized knowledge about not only format and style in your field, but also about how to argue, how to appear credible, and how to frame an issue in your discipline. This discipline-specific knowledge about writing in a field is a union card earned by initiates to the field. It is learned in context, and it typically doesn't transfer well to new contexts. The Writing-across-the-Curriculum project at CMU is an attempt to teach some of this more discipline-specific writing knowledge. However, that program is built on the foundation of a freshman program that tries to teach more general rhetorical knowledge.

The freshmen program is concerned with knowledge which does transfer—rhetorical knowledge about how to manage the writing process itself and about the features of standard writing tasks. Let me give an example of this more basic, transferrable knowledge and then say briefly how we have designed our freshman course, Strategies for Writing, around it.

Writing a Problem Analysis: A Case in Point

Writing an analysis of an issue or problem is a familiar school task. People are asked to define and analyze problems in many disciplines—whether they are engaged in historical or literary analysis (e.g., interpreting the conflicting forces and values behind the Civil War), in philosophical inquiry, or in management decision making. Later, many on-the-job writing tasks start with an analysis or description of a problem. We might also add that an important part of personal development in the school years is learning to analyze conflicting values and problems in one's own experience.

Writing an effective problem analysis is also difficult. It means creating an essay or report which defines, explores, and structures one's understanding of a problematic issue. Moreover, it means pushing one's understanding to the precision and organization of prose and explaining one's own conception of that problem to a reader. Perhaps it is not surprising that writers don't always rise to the occasion. In place of a bone fide analysis, students may simply describe a situation, tell the story of what happened, or perform what in computer science would be called a "knowledge dump." Analyzing problems in an articulate written way is a skill people learn with some effort, it seems.

Analyzing problems in this way appears to be important to success in many areas. In the social sciences, for instance, the significant difference between expert and novice policy analysts in one large study was not attributable to the amount of topic knowledge each possessed, but to the experts' process of elaborating and structuring their exploration of a new problem (Voss, et al., 1983). The experts in the Voss study, who were asked "as Minister of Agriculture to
solve the Russian wheat problem,” simply managed their thinking process in a different way. In
fine arts, Getzels and Csikszentmihalyi found that one of the best predictors of a student’s
success ten years out of art school was the student’s ability to define and develop an artistic
problem out of a given set of materials (1976). In modern rhetoric, the process of discovering
and communicating problems is seen as central to writing (Young, 1970, 1974). More recent
research on composing has begun to show us how this cognitive process develops and why
beginning writers find it understandably difficult to move from knowledge telling to analysis
and to imagine the reader as a separate interpretive mind (Bereiter and Scardamaia, 1982;
Flower, 1979).

There is yet another wrinkle to this task for a writer. Writing always occurs within some
discourse community—some body of readers who accept certain values, assumptions, goals. A
problem analysis written for a public policy argument differs from an analysis that begins an
engineering report—even if the same writer is composing both on the basis of the same
research. Student also need to learn the ways their writing must respond to the expectations
of the community to which they are writing. They must learn to adapt their own knowledge
to their readers.

Writing a problem analysis, then, exemplifies a writing /thinking task which crosses many
disciplines. It illustrates the way writing can foster many educational goals, if we are prepared
to teach the intellectual processes that cross boundaries.

In designing our freshman course, Strategies for Writing, we have tried to achieve transfer
in two ways. One, which we discussed in the early parts of this chapter, is by teaching
strategic knowledge for how to manage one’s own writing process—whatever the task. The
second is by choosing to focus the course on three kinds of papers: a problem analysis, a
thesis with support paper, and a proposal or recommendation (followed by an oral presentation
of that proposal). These are not the only genres we could have chosen, but we felt these tasks
were "core" tasks that turned up in much of the writing CMU students will do. We felt that
if students could handle these tasks with a new level of intellectual awareness and confidence,
the freshmen course would have made a real contribution to a liberal-professional education.
Moreover, these three tasks build on one another. Papers which support a thesis are usually
grounded first in an understanding and analysis of a problem or issue. And effective proposals
must turn a problem analysis and evidence into a cogent plan for action.

One must be cautious making claims about transfer across courses. Transfer is as difficult
to achieve as it is critical to do so. Designing a course that "calls for" or "allows" transfer is
no insurance. In educational research this is sometimes called "blind training," in which the instructor with the big picture knows what a skill is good for, but the student merely learns to do it. The problem is that a theoretically "transferrable" skill may henceforth only be cued by the classroom situation in which the student learned the skill, or by an assignment or exam that explicitly calls for it.

We have approached the problem of transfer in the same spirit as the general problem-solving course designed by John Hayes in Psychology. We treat transfer as a problem in metacognition and place it directly in the hands of the students. That is, we ask them to look at their own knowledge, to look at the strategies we are teaching, and to tell us ways they can transfer what they are learning. Knowledge doesn't transfer—people transfer it. We believe that for writing, teaching rhetorical principles, powerful strategies, and how to manage one's own process has the greatest potential for long term learning and transfer.

Principles of a Problem-Solving Approach to Writing

In a sense, the writing program at CMU is itself an example of the means and ends of design. Rooted in the tradition of classical rhetoric, it is based on an image of a speaker or "rhetor" performing in a community of peers. The field of rhetoric is a 2000 year analysis of that situation—of how speakers, audiences and knowledge interact. To that tradition we have added a rapidly growing body of knowledge about cognition and writing and about the ways discourse communities operate and speak to one another. We have tried to design a curriculum that turns this new knowledge into instruction that works. But at the same time, we are treating our own efforts and our students' development as the subject for continuing research in how strategic knowledge can be learned and taught. Some of the principles that stand behind the design of a problem-solving approach to writing and much of the work in the Rhetoric Program are these:

1. Teach students new, more powerful strategies for how to conduct the process of writing itself. That is, strategies for planning and generating ideas, for examining issues and problems, for organizing, for analyzing and simulating a reader, and for testing, diagnosing and revising their own writing.

2. Teach students how to manage "core" tasks, such as analyzing problems, making proposals and supporting arguments, that are building blocks to discourse in many disciplines and in many genres. At the same time, teach students to recognize the distinctive features of discourse in different intellectual communities—and the special demands to the communities to which they aspire.

3. Teach a problem-solving approach to writing. Instruction in strategies and tasks isn't enough. We also need to teach the student. That is, to teach writers to understand their own process, to attend to their own cognition. We want students to learn to manage their own thinking process as a writer, to evaluate its effectiveness, and to promote their decision making to the level of conscious choice. We want them to
be able to enter new discourse communities, to learn new genres when we aren’t there to teach. And we want them to be able to build a thoughtful, considered representation of a new task—to give themselves the goals that matter and to call on the strategies they know. In essence, managing one’s own thinking is a form not only of intelligence, but of power.

4. Teach writing in the context of real, knowledge-rich tasks. The best method of teaching writing is, we believe, a cumulative one. At CMU students start, as freshmen, with an intensive and systematic introduction to rhetoric and problem solving. But a good program can not stop there. Students need opportunities to transfer those principles to writing in their own field—to learn the discourse-specific conventions and strategies for handling knowledge in their field. This happens in part in the sequence of upper-level professional writing courses we offer, in the Writing-Across-The-Curriculum projects, and in those upper-level "design" courses which make the writing process a substantive part of the design process. The principle behind all these efforts is not mere "practice" but "cumulative education" in writing. When students shift from the role of learner to maker, they have the rare opportunity to write as a professional in a learning setting where they can get an articulate and supportive response from a real academic and professional community.
6. DESIGN INVISIBLE, DESIGNED VISIBLE

Akram Midani

The object of this discussion is to investigate certain fundamental aspects of design as they are applied, consciously or otherwise, in the quest for making a play for the theatre. The basic assumption is that drama, as it is known in the western tradition, is an invented form. As in the case of all inventions, it is the culmination of a number of innovative measures taken by many such as Arion and Thespis prior to the time of Aeschylus who, according to Aristotle was credited for the creation of a definite design based on the two-actor pattern.

The model which is discussed is that of _Oedipus Rex_ by Sophocles and its analysis of its various attributes by Aristotle.

* * *

It was the French Poet Paul Valery who said that reading Dostoevsky makes him always shiver, for he realizes that someone, in solitude, has constructed with the use of the pen a device aimed at manipulating his feelings. "Literature is the art of playing on the souls of others. It is with this scientific brutality that our epoch has seen the problem raised of the aesthetic of the Word, that is the problem of Form. Given an impression, a dream, a thought, it must be expressed in such a way as to produce the maximum effect in the soul of a listener—and an effect entirely calculated by the Artist." A poem, said Valery in a similar context, is "a sort of machine for producing the poetic state by means of words."

The importance of Aristotle's systematic reading of a Sophoclean "machine" and his analysis of its "design" must be seen in light of the fact that he was somewhat a contemporary, being born twenty one years after the death of Sophocles, and sixty years after the initial performance of _Oedipus Rex_. Far from intending to produce a manual, Aristotle's objective seemed to have been an explication of the new tragic form by taking one of its "machines" apart.

In the past decades there has been a trend among many interpreters of the arts to link artistic creation with unexplainable mysteries, the domain of hallucination and the supernatural. A prevailing theory in the defense of the irrational was to view art as a form of primordial order on the level of magic, sorcery, and alchemy. We know now that this trend recurs when illusion and trickery gain the upper hand over imagination and creativity as offsprings of reason; and when the deliberate building of effects aimed at exacting the reactions of the mind and the senses is ignored.
What has been perceived by some as magic is a system of procedures precisely administered.

"So many words." says Stravinsky, "hazy at least, that keep us from seeing clearly in a field where everything is balance and calculation through which the breath of speculative spirit blows."

Aristotle observed the recently emerging art form, the drama, as an invention for the "imitation [or simulation] of men in action" which differed from narration by enactment (doing) and impersonation.

What are the "building blocks", depicted by Aristotle, for the designing and hence inventing of this form?

First there was the storyteller, the narrator of epics, the singer of songs: one actor telling all events and assuming all characters. At a certain time, in the latter part of the sixth century, Aeschylus introduced a second actor and with him the dialogue which required individualized enactment.

It is interesting to note that the Greek word for actor is hypokrites which means an answerer.

Throughout, a discovery has been made, that the actor [answerer] is the essential "building block", a unit. Interaction between the first and the second units has altered the way information is conveyed from the narrator's mode to the answerer's mode, and that endless possibilities can be realized by increasing the number of units, that is the answerers. The introduction of the third actor is a contribution attributed to Sophocles by Aristotle.

Thus a definite design has been achieved. The play Oedipus Rex has to be decoded by Aristotle so that he can expose its parts and show its functioning. I must bring to attention a contemporary dramatic treatment of the Oedipus myth by Jean Cocteau which he entitled La Machine Infernale.

The tragedy by Sophocles begins with:

An omnipotent Oedipus, the king of Thebes, addressing a delegation from his kingdom, seeking to know their concern; the leader explains that the city is being ravaged by a deadly pestilence, so they come to him, their savior, to rid them of this black death. He did it once, when he overwhelmed the Sphinx.
(The reference to the Sphinx is to trigger the memory of the audience of the strangler who stood at the gates of Thebes destroying everyone who failed to answer the riddle: "what is that walks on four in the morning, on two at noon and on three in the evening?"; only Oedipus was able to answer the triadic riddle. It was upon his first arrival at the gates of Thebes when he gave the answer which allowed him to demolish the Sphinx by meeting his challenge: it is man, said Oedipus, who crawls as an infant and walks with the aid of a stick in his old age. Thebes rewarded him by making him king.

Such a recollection of the Sphinx, the strangler, in the opening dialogue was directed at an audience who must recall the preceding events in the course of the enactment of the familiar myth through a concentrated plot derived from it; they must supplement the information while the plot is unfolding in a series of flashbacks. The collective memory provides a cohesion in a manner akin to the function of the persistence of vision in the cinematic experience).

- Oedipus tells the delegation that he will save the city. He has sent a prince, his brother-in-law, to consult the oracles and identify the cause of the plague.

- The prince appears to say that when King Laius, the former king of Thebes was slain his murderers remained at-large. The criminals must be caught and punished so the plague will be lifted.

(Here begins the machine within the machine).

- Oedipus, driven by his irrepressible power for cognition, will preside over the investigation.

- He calls in a blind prophet to learn that the assassin of the king is himself a king; that this king shall be proved a father and brother of his children and the husband of the woman who gave him birth; a usurper of his own father's bed.

- Oedipus is disturbed. He asks about the former king. The more he hears the more he suspects a conspiracy to dethrone him. Yet with his insatiable mind he insists on questioning further.

- As Queen Jocasta enters, she tells him of King Laius, her former husband, who was told that he will be killed at the hand of his own son. The old king was killed by foreign highway robbers at a place where three roads meet; as for his son, their son, when he was three years old, his ankles were pierced and he was given to others to cast him upon a hill.

- Oedipus knows that he is the son of of the king of Isthmus, and that he fled his home so that he will not kill his father, King Polybus, and marry his mother as he learned from another prophecy. Yet, he remembers the crossroad where, on his way to Thebes, he encountered a caravan whose leader tried to force him off the path. He became angry and killed all those in the caravan including their leader.
• Jocasta advises him not to attach importance to such a coincidence: King Laius's party was made of many men who where attacked by many robbers, so she learned from the only man who escaped the massacre.

• This man must be brought in, insists Oedipus.

(Up to now the collective memory of the audience is assisted by the recognition of a triadic pattern: the riddle, with its triquestion and solution; the triple crossroad; and the set of three secondary informers: the prince, the blind prophet, and Queen Jocasta.)

Aristotle observed that drama has generated a synthesis of pre-existing art forms namely painting and music, the latter is laden with recurring and recognizable patterns).

• A messenger arrives, asking to see King Oedipus who has gone back into his palace. He tells Jocasta that King Polybus of Isthmus is dead. She expresses joy that her husband is the son of the king of Isthmus. When Oedipus returns he receives the news calmly, and he questions the messenger as to the cause of the King's death; he learns that it was a natural death, of old age. He is still afraid to go home lest one part of the prophecy is fulfilled: that he may still marry the queen, Polybus's wife.

• The messenger assures Oedipus that he is not the old King's son and that it was he, the messenger, who found him with pierced feet on a slope and gave him to the childless King Polybus and his wife. After further interrogation by Oedipus the messenger reveals the truth, that he received him as an infant from a shepherd.

• Jocasta is terrified. She begs Oedipus to stop hunting for information about his birth. He refuses: "I will not be persuaded to let be the chance of finding out the whole thing clearly."

• The shepherd is brought in. Under strenuous cross-examination by Oedipus, he corroborates the messenger's information. The Queen enters the palace. Now the truth that Oedipus searched for is upon him, "All true, all clear. I who have been born from whom I ought not, marrying whom I ought not, and having killed whom I should not". A conclusion of yet another triadic pattern.

• After this "self-recognition" proud Oedipus, reduced by shame, goes into the palace.

• A second messenger (thus completing a set of three primary witnesses) appears to tell the delegation from Thebes and the audience what he saw: the death of Queen Jocasta by her own hand; when the King returns, they know that he has blinded himself.

• As he is led into exile we hear the townspeople bidding him farewell: "Oedipus, who knew the famous riddle, a man most masterful. Count no mortal happy till he has passed the final limit of his life free from pain".

To examine the parts of the tragic form the isolated which make a whole as meticulously designed as this example by Sophocles. In his Poetics, he observes, there are six constituent
elements in the design of the tragic form: Thought, Plot, Character, Diction, Music, and Spectacle. The first three of these elements; namely "thought," that is the thesis or the idea of the play; "plot," which is the story to be enacted; and "character, or characters" through which the impersonation of the actors takes place, can be discerned from the text; the rest, however, can only be realized in a performance before an audience.

The tragedy of Oedipus is wrought from the thought that pride does not alter human vulnerability; a plot based on the myth or the story of the house of Thebes; and characters which are economically and functionally selected to convey the thought and to tell the plot by imitation enacted in an articulated diction and song with the help of visual effects which constitute a spectacle.

A design must simultaneously reveal form and function, and the integral relationship between them which governs its effectiveness. The argument whether drama is a text or a performance, literature or theatre, seemed to have been settled at the outset. A play cannot come in to being unless it is played before an audience. What is to be played is both the design and the designed.

When plays become performance, they exist in a domain shared by space and time. The permanent and the ephemeral: as text their evidence is permanent; as enactment they only happened once, and when they are repeated they will not be the same. They don't result in a tangible object for adequate future reference. In the words of Henri Gouhier: "plays are made of acts which are actions in search of actors who actualize them." The paradox here is that the artifice of design for the ephemeral and mostly that which is related to actors is the play's strong and only link to reality.

The character Polonius says to Hamlet in Shakespeare's tragedy that he was an actor once, and that he played Julius Caesar and was killed by Brutus. The Elizabethan audience must have derived some pleasure from such an invitation to have a peek at the design of the play in performance, since the actor who is playing Hamlet is the same who plays Brutus and who will eventually kill Polonius, a character played by the actor who plays Caesar. It was up the audience to make reality take place, a function clearly defined within the design of the play, often between layers of theatricality. In Oedipus Rex it was emphasized by its actor-surrogates, i.e., the delegation representing the people of Thebes; in the effect of the unfolding of events on them and subsequently on the audience.

The curiosity of the audience is enhanced by that of Oedipus; their knowledge is augmented by his; in the end, however, they don't share his fate. The actors, while enacting
the plot, fill the audience and their surrogates with pity and terror so they usher poor Oedipus off stage with a recognition of the human condition: a sense of purgation or catharsis.

The last three effects: pity and terror concluding with catharsis are essential for the understanding of Aristotle's decoding of the design of the machine. They are directed with accuracy toward the audience who reciprocate with a suspension of disbelief.

All this sounds neat, and with Aristotle's help, too orderly. Yet the same basic design remains even when characters become complexed with the imitation of contradictory men in action; when stories are woven within stories extending and multiplying and concluding with an integrated structure. The audience are given a thought, a line, or a shape to be interrupted by an afterthought, a line coming from an unexpected direction, or a shape which is not allowed to be fully reached. Plots are embodied in other plots, as if all events, characters, and thoughts, are at different intervals, subjected to the syncope, i.e. the chopping up or the cutting off of elements which had they been allowed to stay, could have given the semblance of continuity.

In this developed model of the dramatic form syncopation must be understood not as a change in the fundamental elements of the design but as a shift of accents and rhythms. Interruptions and resumptions keep the audience in a perpetual state of expectation. The plot, with its sub-plots never loses the thread and the stories all come to a comprehensible conclusion.

In this basic design, as we see it codified by Aristotle, with its development through the ages since Aeschylus and Sophocles, the most satisfying results have been those constructed with calculated sequences reflecting the kinship between reason and imagination.
A direct descendant of the narrative tradition, the plot, or the story being told, remains the most attractive to the audience, and the most vital for the structuring of the dramatic form. As drama developed, its plots multiplied, along with the diversity of the characters played distinctly by "answerers", into an array of sub-building blocks. It was the Italian playwright Carlo Gozzi (1720–1806) who maintained that all plots can be categorized in thirty-six dramatic situations, each can also have a number of variations. Goethe commented that "Schiller took great pains to find more, but he was unable to find even so many as Gozzi". Quoted in an updated version of Gozzi's Thirty Six Dramatic Situations by Georges Polti. (1916).

Quoted by Gilson in Forms and Substances in the Arts. p.248.

"He was an actor once?" We imagine the audience reflecting silently and fleetingly: "What do you mean once? you are an actor now! "

Midani, in "Abe Ajay's Untangled Thread". introduced a second actor and with him the dialogue which required individualized enactment.
7. WHY TEACH DESIGN

Stefani Ledewitz

One can envisage a future...in which our main interest in both science and design will lie in what they teach us about the world and not in what they allow us to do to the world.

Herbert Simon, The Sciences of the Artificial, p. 188

Although the design studio has a long history as the core of architectural education, it struggles today for intellectual rigor and academic credibility. The central role of design in the architectural curriculum can be traced to the ateliers of the Renaissance, which in their day—until the late nineteenth century—encompassed all aspects of architectural education. Even in the modern university, where knowledge areas are increasingly segregated into separate courses, design still claims one-quarter to one-half of total credits. It typically absorbs an even greater share of students' time, as well as a high proportion of teaching time. This represents a significant commitment of resources, especially in view of the ever-expanding base of knowledge on which professional competence depends.

To judge from architectural curricula today, there is a prevailing consensus that design is an essential part of professional education. Although there is widespread agreement about the necessity of teaching design, there is not such an agreement about why it should be taught. From time to time, objections to the dominant role of design are raised. From one such point of view, design is only one of many professional specialties in architecture today, and should be equivalent to other professional sequences, such as building technology or architectural history. From another viewpoint, the major role of design is to integrate the content of other courses. Design is considered to lack an independent knowledge base, and its proper role is seen as a component of other professional courses. From a third perspective, design is only "practice for practice", in which students play architect so that they can learn how to do it "for real". In this light, the design studio is seen as a weak substitute for actual practical experience.

In addition to these questions of professional preparation, it is argued that the subject of design is not sufficiently rigorous, by university academic standards. Compared to the precision and clarity of thought that is valued in the university, design is a "messy" activity. In design, there are no unique solutions to problems. No prescribed methodology exists to insure steady progress toward a solution, nor are there absolute criteria for evaluation. Some critics argue that design, and the evaluation of design, entail a great deal of individual judgement as opposed to a general body of theory. Others see design as no more than the exercise of a
vocational skill, without any intellectual content of its own. Still others consider design lacking in rigor because it is insufficiently rational. They see it as a "black box" creative process that depends on innate and inherently unteachable talent.

These objections, it seems to me, are based on an insufficient understanding of the design process, along with a too-limited view of the purposes of education. I believe there are good reasons to teach design that go beyond the practical value of preparation for practice and are well within the purposes of the university.

In 1970 Peter Stringer, a psychologist with the Architectural Education Research Unit in London, proposed establishing an undergraduate course in architectural design, "architecture as education", for students throughout the university. In his view, the architectural design studio offered an educational experience uniquely suited to preparing students "to meet the unpredictable problems that a rapidly changing world will present". The benefits of a design education, he argued, are that it addresses the quality of life and organization of society, that it teaches sophisticated problem-solving, and that it explicitly involves value judgments. "This type of basic course, which incorporates a mixture of physical sciences, technology, social sciences, arts and design, could be a good preparation not only for specialization in architecture, planning or the construction industry, but for any field which required a broadly based intellectual education, with a particular emphasis on complex, value-informed problem solving."

Herbert Simon and others have made strong arguments for introducing design into professional education programs as a creative problem-solving process. Observing that design is not unique to "design professionals", Simon defines design as an intellectual process independent of any particular field of knowledge.

Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences. Schools of engineering, as well as schools of architecture, business, education, law, and medicine, are all centrally concerned with the process of design.

Along the same lines, it could be added that most people, not just professionals, are engaged in design processes in the course of everyday living—putting together a family financial plan, planning a community fund drive, furnishing a house, and so on. Learning to
design is developing a capability to solve problems in many different situations.

Yet not all problem situations call for design. The kinds of problems that design solves are complex and ill-defined, problems in which the characteristics of neither the solution nor the problem itself is obvious. Simon describes a continuum of problem-solving that extends from routine problem solving (familiar problem situations) on one end to blind search (wholly novel problem situations) on the other. Designing is a creative problem-solving process that lies in the middle of the continuum. However at any point between routine problem solving and blind search, Simon asserts, the fundamental cognitive processes are the same: "Problem solving involves selective trial-and-error search in a vast space of possibilities." It is only the balance between selectivity and trial-and-error that changes. The more complex and ill-defined the problem, the more critical selectivity becomes. It can be either conscious or unconscious. The intuition of an experienced designer is largely-unconscious selectivity that short circuits trial-and-error. Much of design process research has aimed at making selectivity conscious. Simon points out that various heuristic techniques, derived from mathematics, statistics, and operations research, are available and teachable. Although it should be acknowledged that these methods still deal only with well-defined aspects of problems and are relatively ineffective compared to the power of an expert designer's intuitive knowledge, learning design methods and, more importantly, experiencing the design process itself can lead to more sophisticated problem-solving capabilities in a wide range of disciplines and therefore play an important role in a university education.

I would like to suggest, however, that design has more to offer as an educational experience than the skills for professional practice or even the ability to solve problems creatively, and I believe these qualities relate directly to the basic goals of a university education. It is based on a somewhat broader look at the design process, which conceives of design as a process that not only changes an external situation but also changes the designer. Design is an interactive process between the designer and the world. It is this two-way dialogue that has not until recently been recognized in models of the design process. Not only does the interactive model of design account better for the observed behavior of designers, but it also begins to convey the experience of the design process and the motivations for designing.

Dr. Simon hints at this in a short section at the end of The Sciences of the Artificial, in which he speaks of design as a "valued activity": "The act of envisioning possibilities and elaborating them is itself a pleasurable and valuable experience. Just as realized plans may be a source of new experiences, so new prospects are opened up at each step in the design process." It is this experience that I believe motivates those who choose to make design their
life's work, and that also motivates professional designers to teach design. It is, to use Abraham Maslow's terms, an experience of "self-actualization" that design affords, and it is as an opportunity for self-actualization, as well as for professional preparation, that design should be considered a valued educational activity in the university.

Maslow describes self-actualization as a continuous process of human growth:

...Healthy people have sufficiently gratified their basic needs for safety, belongingness, love, respect and self-esteem so that they are motivated primarily by trends to self-actualization, defined as ongoing actualization of potentials, capacities and talents, as fulfillment of mission (or call, fate, destiny, or vocation), as a fuller knowledge of, and acceptance of, the person's own intrinsic nature, as an unceasing trend toward unity, integration or synergy within the person.77

The purpose of education, Maslow asserts, is self-actualization: "...the goal of education—the human goal, the humanistic goal, the goal so far as human beings are concerned—is ultimately the self-actualization of a person, the becoming fully human, the development of the fullest height that the human species can stand up to or that the particular individual can come to."78

Acquiring the skills and values of a profession is the means to the more important end of becoming a mature, healthy, and wise person. Although Maslow intends the concept of self-actualization to be understood more holistically, I would like to suggest that it provides an appropriate framework for considering three aspects of design that make it both personally fulfilling and educationally valuable: as a learning experience, as a creative experience, and as a playful experience.

Learning that contributes to self-actualization is what Maslow calls "intrinsic learning", which he distinguishes from the learning of "external skills and capacities" in response to expectations by others—parents, society, or professional peers. Intrinsic learning, learning for its own sake, is internally motivated by our search to understand ourselves and the world. In this sense, Maslow sees learning experiences as a way of growing into a self-actualizing person. He describes creative experiences as equally fundamental to self-actualization. "The concept of creativeness and the concept of the healthy, self-actualizing, fully human person seem to be coming closer and closer together, and may perhaps turn out to be the same thing."79 He asserts that creative experiences, such as in the arts, deserve to be at the core of education, rather than the "whipped cream". Similarly, playfulness is described, not as a frill, but as one of the central values (being-values) in a self-actualizing life. He describes self-actualizing people as persons who experience playfulness in their work. In their dedication to a cause outside themselves, they fuse work and play. Moreover, they value their work for the fun and joy it provides.
Design as a Learning Experience

Architectural design studio teachers often tell their students that design is always a learning experience, that the difficulty and uncertainty they experience in school will characterize their whole professional career. Although this sometimes worries students more than excites them, the intention is to convey a fundamental source of enthusiasm for the practice of architecture: the experience of self-actualizing learning.

Self-actualizing, or intrinsic, learning is what Jerome Bruner calls "learning by discovery", which he describes as "rerearranging or transforming evidence in such a way that one is enabled to go beyond the evidence...to new insights". According to Bruner, learning should be seen as a process of discovering insights that modify the model of the world we have constructed in our minds in order to adapt that model to new experiences. Learning by discovery is characterized by three features: insight, inquiry, and experience. An insight is a mental construct that enables us to make sense of our experience. Gaining insights, as opposed to acquiring knowledge, is the purpose of learning by discovery. Learning occurs through a process of inquiry, which requires an attitude of expectancy: "For the person to search out and find regularities and relationships in his environment, he must either come armed with an expectancy that there will be something to find or be aroused to such an expectancy so that he may devise ways of searching and finding." Learning by discovery is also learning by doing, that is, acting and then reflecting on it and representing it. Our actions are a primary source of learning. Inquiry is stimulated by our own experience: by inquiring into those experiences, we gain new insights by which we enrich and deepen our internal representation of the world.

Design is a search for order. The task of design is not merely to satisfy the requirements of the problem, but to do so in such a way that the satisfaction of each requirement contributes to the order of the whole. In trying to reconcile diverse and conflicting demands, the designer is always looking to simplify, to impose a more encompassing order on the situation. Lesser orders are gradually subsumed into greater orders. The designer recognizes a greater order by its power to gather lesser orders into itself—a kind of "gravitational" force that pulls the disparate aspects of the project together. An architect, for example, may discover a single geometry that unites the movement through rooms in a building with the sequence of movement across the site. Or he might reinforce spatial relationships through the placement of structural and mechanical elements. Elements that initially serve only a single purpose are made to serve many purposes.

It is the search for "regularities and relationships" of an imposed order that makes the design process a learning experience. The designer's understanding, or internal representation,
of the problem consists at first of fragmentary and inconsistent constructs, which are gradually replaced by more complete ones. The increasing order that evolves in the design solution is a reflection of the designer's "higher-order cognitive structure" that "has the effect of supplanting the niggling complexities...of the less-good orders that precede it." 82

Thus design is a search for both an external and an internal order, not only for an external course of action but also for an internal model of the problem. When the designer succeeds in solving the problem, he also experiences the delight inherent in the discovery of a new order. It is the experience that Bruner calls "intellectual potency", the joy of learning for its own sake.

Every design problem offers an opportunity for learning, a new problem that the designer, however experienced, has not solved before. The task of design is to solve problems that are, by definition, unique, complex, and underconstrained, and each of these factors creates a opportunity for learning. Although a particular design problem may be similar to other problems or belong to an identifiable class of problems, it is always in some respects a unique situation, to which any previous solution must be adapted. In architecture, circumstances of site and program (as well as client and budget) are rarely comparable, and considerable creative effort goes into finding appropriate precedents. Students in design studio are taught that a good building reflects its unique circumstances and, moreover, that these circumstances should be welcomed as formgiving opportunities. John Archea points out that architects generally enjoy this aspect of design: "They prefer commissions to design one-of-a-kind buildings. When they are commissioned to design a typical building, architects superimpose their own agendas on the client's program to make it unique." 83 In fact, a widely held image of the "ideal" architectural practice consists of a great variety of projects that continually presents new learning opportunities.

However, design is not commonly thought of as a learning process. In part, this is because the outcome is not what is traditionally thought of as knowledge, but rather, "insights". These insights may concern the nature of the design problem (e.g., how sensitive the spatial organization of a building is to its site); they may concern the design process (e.g., how early in the process building service requirements should be considered); or they may concern the designer himself (e.g., how his attitude toward nature is reflected in the siting of the building). Insightful knowledge, or "seeing into" the problem, is a "grasp of things that goes deeper than words". 84 It is tacit or, in Polanyi's terms, "personal knowledge" that we cannot make explicit. "...To the extent to which our intelligence falls short of the ideal of precise formalization, we act and see by the light of unspecifiable knowledge and must acknowledge that we accept the
verdict of our personal appraisal." Spatial experiences are often difficult to put into words. Many people, for example, know the experience of emerging from the Fort Pitt Tunnel into downtown Pittsburgh, but would find it hard to express in words.

Although this knowledge cannot be communicated directly, it is not necessarily unique to each individual. A piece of music or a particular environment, for example, can elicit a common response from a number of different people. Such a response, in fact, may be sufficiently predictable to enable the designer to anticipate the effect of his decisions, based on his own observations. The fundamental difference between insightful and empirical knowledge is not so much the degree of predictability as whether it can be explicitly communicated and tested. Insightful knowledge is no less susceptible to correction than empirical knowledge, although it is often by means of attentiveness to our own informal experience rather than formal evidence that we examine and refine our insights.

Design differs from other learning experiences not only in the kind of knowledge it produces, but also in the process of learning. Progress in design is a path that alternates forward and backward, through the classic "feedback loop". The forward movement represents the development of the design proposal (action), while the backward movement represents the development of the designer's knowledge and intentions (reflection). When the designer moves forward, he generates new information; when he moves backward, he incorporates that information into his model of the world, thus learning from his experience.

A closer look at the design process reveals a spiraling series of cycles, in which each cycle consists of generating a proposal and evaluating it. A cycle begins with the designer's internal representation of the problem, his understanding of the design "task". The designer generates a proposal by conceiving of an idea (imaging) and externalizing it in a way that can be communicated to himself and others (presenting). It is then evaluated (tested) against what he knows about the solution. The outcome of the cycle is a new representation of the problem from which to begin a new cycle. The designer's first representation of the problem is typically vague and incomplete, since he is not yet aware of the conflicts and opportunities latent in the problem. The initial proposal, therefore, may capture only the broad outlines of the final solution. More importantly, however, it enables the designer to develop a better understanding of the problem itself. Each succeeding proposal more closely approximates a satisfactory solution, as the spiraling cycles tighten to within the "domain of acceptable responses".

Learning through design is "backwards". In the activity of imaging, the designer discovers
insights; in presenting, he acts on those insights; in testing, he reflects on them. In this learning process, action precedes reflection: synthesis precedes analysis. Representation is not so much the product of completed ideas as the vehicle for developing them. Design itself is a process of inquiry: inquiry is the driving force in moving from one cycle to the next. Because the purpose of the cycles is heuristic, the questions that the designer asks of one proposal must serve to generate the next. Asking the right questions is more important—and more difficult—than finding the right answer. Thus the most significant difference between novice and experienced designers is in their ability to represent the problem appropriately, which is the ability to raise relevant questions at the appropriate point in the process. It enables the experienced designer to sustain a design process through many cycles to a closer approximation of a solution, whereas a novice either settles for a premature solution or repeatedly interrupts the process to try new approaches. The more experienced the designer, the greater is his capacity for learning from design.

A student who is learning to design is therefore learning something much more fundamental than how to solve problems: he is learning how to inquire. By developing an open, inquiring mind, he can discover new insights from his experience. That is, he learns how to learn: he experiences "the power of the mind" that is the delight of learning.

Design as a Creative Experience

At the heart of the design experience is what Rollo May calls the creative encounter: "Creativity occurs in an act of encounter" between two poles, the self and reality. This creative moment is surely one of the most rewarding aspects of design, heightened by the hours of frustrated efforts that often precede it. It is what Maslow refers to as a "peak experience" in life, "a transcendence of self...a oneness where there was a twoness, an integration of some sort of the self with the non-self". It is an experience of timelessness, when one "loses his past and his future and lives only in the moment".

As every designer recognizes, the creative moment is only an instance within a long sequence of experiences. "It is misleading to refer to the creative process as though it were a single, unitary process. The term should be thought of as no more than a convenient summary label for a complex set of cognitive and motivational processes, and emotional processes too, that are involved in perceiving, remembering, imagining, appreciating, thinking, planning, deciding, and the like." Each cycle in the design process, consisting of imaging, presenting, and testing, is in its entirety an engagement in creativity.

The constraints and goals that characterize a design problem at the outset are always extraordinarily diverse, ranging from architectural "themes" and historical precedents to user
requirements, soil conditions, and product delivery times. The disorderly circumstances of the
design task are not only tolerated but welcomed by a creative person, who typically prefers
chaotic and complex situations. "Behind this inclination to like and to construct what is not
too simply ordered, there appears to be a very strong need to achieve the most difficult and
far-reaching ordering." While the designer must become thoroughly familiar with the
problem circumstances, the design concept that orders them does not emerge from the
circumstances themselves but must be "imposed" onto them. It is experience and intuition that
enable the designer to generate a design concept that gathers up the many bits of information
and begins to bind them into a larger whole.

With all the complexity that is inherent in a design problem, the greatest challenge to the
designer's creativity is ultimately the freedom it affords. He must introduce into the design
process his own images, aspirations, and values as "autonomous constraints" or "internal
variety reducers". Even deciding in what order to consider different aspects of the problem
represents the designer's own imprint. The designer cannot be detached from the problem,
but rather must engage himself intellectually and emotionally in the problem. The capacity to
fuse self with world for a time demands a strong sense of security and identity, but, as
experiments in creativity have shown, "the objective freedom of the individual is at a maximum
when this capacity exists, and creative potential is directly a function of freedom".

Creativity is most obvious in the process of "imaging" in design, the generation of mental
representations, which may be visual pictures, analogies, or abstract ideas that provide visions of
solutions-in-principle or present implications for physical form. An image for an arts center
might be a pile of children's blocks, a picture of a church by Alvar Aalto, a ring of trees
around a grassy hollow, or the idea of "intersection". Images can be especially powerful if
they link very different realms of experience, suggesting different meanings simultaneously.
For example, in designing a synagogue, a student took the images of a "polycentric geometry",
"form proliferating from the division of unity", and "praying between columns" from his
experiences with painting, physics, and Jewish history. Imaging is the rich synthetic aspect of
design, in which intuitive connections or reconstructions open up the problem to new
possibilities. The designer needs to be able to draw from an inner store of memories,
associations, and feelings, which arises from a heightened consciousness of his inner world of
ideas and emotions.

But the creative experience of design is not limited to imaging. The "presenting" and
"testing" of images engage the designer in what Maslow calls the "secondary processes of
creativity". Presenting is the process of transforming mental images into physical form, such
as sketches or models, which demands and reinforces the designer's commitment to his ideas. In imaging and presenting, which are the generative events in the design process, the designer tends to identify strongly with the evolving design, even in its early stages. For this reason, many novice designers have trouble stepping back to test their proposal against the array of constraints and intentions to which it is accountable. However, the cyclical process of developing an initial idea into a fully documented building proposal, with the personal risk-taking that it entails, ultimately contributes to the realization of creative autonomy. Out of the loss of the sense of self emerges a stronger self, a more independent self more clearly differentiated from the world. Design thus affords an opportunity to experience creative autonomy. It is the opposite of the attitude of compliance, in which an individual lacks a sense of control over his life, but instead is an attitude of self-acceptance and confidence in one's own potential to act in the world.

Design as a Playful Experience

The boundary between the inner world of the self and the outer world of reality is where the creative experience of design occurs. From the tension between the two worlds, the designer tries to create a synthesis by discovering relationships that rejoin the two worlds. The creative task of design is to bring together the two worlds, to create a higher order out of existing fragments of order, to reconcile the ideal with the real.

The boundary between the two worlds where design occurs is also an in-between world. It shares aspects of both worlds but is distinctly an "other" world to itself, "a transition from one realm of being into another." This is the world that psychiatrist D.W. Winnicott describes as a "third area of human living, one neither inside the individual nor outside in the world of shared reality." It is the world of play, a world Winnicott sees as essential to human growth.

On the basis of playing is built the whole of man's experiential existence....We experience life in the...exciting interweave of subjectivity and objective observation, and in an area that is intermediate between the inner reality of the individual and the shared reality of the world that is external to individuals.

Huizinga defines play as "a free activity standing quite consciously outside 'ordinary' life as being 'not serious', but at the same time absorbing the player intensely and utterly."

Huizinga's description of play might well be applied to the experience of a design studio, especially late at night, when the "real world" of the daytime has faded. There is often a little craziness and some good camaraderie that develops in the after-hours studio: a model is dissected and turned inside out, one of Aalto's towers is added to Mies' Berlin museum,
somebody decides to design a "habitable fireplug", a paving pattern turns into an odd game of chess. The intensity of work gives way to a feeling of fun.

Design at its most inventive is also at its most playful, as when the designer finds himself creating a new world. What might begin as a statement of user requirements spins itself in the designer's mind into a scenario of people and places, places he begins to experience himself as he creates them. Through the active engagement of his imagination, he "transcends the immediate constraints of reality". Some studio problems are designed to encourage students to stretch their imaginations by transporting them into a wholly novel or unlikely setting. However, even where it is not the major focus of a studio, the ability to experience an imagined world is an essential aspect of learning to design.

Design is also playful to the extent that it is intrinsically motivated—that is, to the extent that the designer has either adopted the problem goals as his own or has read into the problem his own goals. While this does not necessarily change the goals of the project, it does transform the designer's efforts into a voluntary engagement that is therefore wholly his own: "Since play is a process that is sustained by an individual's intrinsic motives and utilizes content determined by him, it is unique to that person." Moreover, this experience can become a self-reinforcing source of enjoyment in design: one writer speaks of the "euphoria arising out of the voluntariness" of play, which reflects the commitment of having freely chosen to engage in the design process.

Play is, moreover, a valuable learning tool. Bruner and others have pointed out that play is, in fact, a process of learning in a low-risk setting. It offers a way of "trying out" new meanings and behaviors that are experimental. A number of studies have shown correlations between playful activity and both creativity and problem-solving, suggesting that "one function of play is an enhancement of problem-solving skills and the kind of innovative behavior which often helps to solve problems". Einstein observed an essential relationship between play and subjective, or tacit, knowledge: "...Combinatory play seems to be the essential feature in productive thought—before there is any connection with logical construction in words or other kinds of signs which can be communicated to others." What we observe in design, then, is the blurring of the boundary between work and play that occurs when a person is committed to a cause or effort beyond himself. Robert Frost described this about his own life in his poem, "Two Tramps in Mud Time":

But yield who will to their separation,
My object in living is to unite
My avocation and my vocation
As my two eyes make one in sight.
Only where love and need are one,
And the work is play for mortal stakes,
Is the deed ever really done
For heaven and the future's sakes.

From the beginning of their design education, students are taught that the reason designers design is because they love designing. Students who are unwilling or unable to enter into the spirit of designing rarely continue very long in an architecture program, and even many of those who graduate will lack either the opportunity or the talent to become professional designers. What they take with them, in any case, is the valuing of design as not only a way of solving problems, but as a way of realizing one's human potential, as a way of living.

Some Implications for a Course in Design

If design is conceived as a self-actualizing experience, the teaching of design should reflect a concern for intrinsic learning, creativity, and play. The most significant implications of these concerns are surely the intangible factors of a teacher's attitude and expectations. However, there are also a number of more tangible implications that deal with the course structure, assignments, and teaching content. They are offered here in general terms, in recognition of the great variety of teaching styles and student needs. Some of these suggestions are not unique to teaching design, but are consistent with recommendations by learning theorists for education in general. Indeed it might be, as Stringer suggested, that the teaching of design might serve in some ways as a model for the teaching of other subjects. These recommendations, however, are intended, in particular, as considerations for designing an introductory course in design.

1. A course in design should give students a "hands-on" experience. If learning to design is a process of reflection-in-action, then we learn to design best through the act of designing. Bruner writes in On Knowing, "It is my hunch that it is only through the exercise of problem solving and the effort of discovery that one learns the working heuristics of discovery; the more one has practice, the more likely one is to generalize what one has learned into a style of problem-solving or inquiry that serves for any kind of task encountered."

2. The course should emphasize strategies for problem definition over methods of problem solution, such as structural calculations or optimization techniques. In order for students to learn how to employ these methods in the design process, they must learn to recognize the need for them. Learning to "frame", or represent, the problem to be solved is more difficult to learn than solving it. If, as Simon has pointed out, the problem is represented in the right way, its solution becomes self-evident. Design assignments should therefore offer students the opportunity to frame the problem in different ways and discover the consequences of these
3. The required products of design assignments should include preliminary as well as final solutions. The preliminary submissions should be versions of the final product, whether it is a set of drawings, a computer program, a marketing plan, or a written document. Students should learn that design is an iterative learning process, not a hit-or-miss "black box" exercise.

4. The instructor should provide good feedback during the process, such as after preliminary submissions, as well as at the end. These evaluations should be seen not only as a gauge of students' progress, but also as a model for them to learn how to evaluate their own design proposals. The evaluation should respond to the way each student has chosen to define the problem. For this reason and because design evaluation entails a high degree of judgement, evaluation criteria set out in advance can only be very general in nature. It is especially important, therefore, to articulate the basis for evaluation when giving feedback and to welcome students' questions about the evaluation.

5. Students should be encouraged to look to many sources of learning, to understand that their own experience can be a source of learning. They should learn to be creative in looking for information and ideas, which may come from unexpected places. Above all, they should be encouraged to learn from each other, sharing ideas, practicing criticism, and learning the value of mutual respect and cooperation. By sharing and comparing design experiences, students gain an appreciation of their accomplishments and support through their difficulties.

6. A high level of order in the design product should be demanded. A well-designed artifact or process should exhibit a unity or wholeness that comes from the successful working out of relationships among the parts. The solution should be elegant, such that ideally nothing can be added to it or subtracted from it without diminishing it in some way. This goal should apply not only to the design proposal itself, but also to the way in which it is presented. Visual order is one aspect of an underlying and pervasive coherence that is the object of design.

7. The instructor should provide examples of good design to communicate to students most effectively what they should strive for. Such examples can help students see possibilities they could not foresee and provide a repertoire of design strategies. They also serve to balance the tendency toward negative criticism in evaluating student work, the tendency to describe what should not be, as opposed to what should be.

8. Design problems should be selected in order to enable most students to experience
success. It is said that we learn more from our successes than from our failures. This is particularly true of the design process, in which many aspects cannot be understood until they have been experienced: for example, to be able to distinguish a strong ordering idea from a weak one, to sense that everything is suddenly "coming together", or to anticipate that one approach will be more effective than another.

9. The course should stress an awareness of the design process, to help students understand what they are learning and how. For novice designers the design process can be a puzzling and even threatening experience, since it is unlike other academic experiences. Students who approach design with unrealistic expectations, such as expecting to find "correct" solutions, can be especially disheartened. By reflecting on their experiences with them, the instructor can help students understand what they are doing and see their own strengths and weaknesses.

10. The course should be fun. In the teaching and the selection of design problems, it should stimulate students' enthusiasm and imagination. Problems might be chosen that suggest different ways of looking at the world; resource material drawn from unfamiliar sources might enable students to break through preconceptions and discover analogies or other relationships in a creative thought process. Problems that incorporate "real world" constraints give students a beginning point, but if too much emphasis is placed on real-world feasibility of solutions, some students, especially novices, have difficulty engaging their imaginations in trying out unorthodox possibilities.

11. The instructor should recognize that subjective judgements are necessary in design, acknowledging his own and making students aware of theirs. This can help students develop their own capacity for critical judgement, as well as helping them avoid confusing intuitive and empirical knowledge. Students should be encouraged to acknowledge their subjective responses in order then to examine them and understand them. Through design, students can experience a holistic engagement of their intuitive and rational capacities, and thereby discover in themselves a greater potential for creative work and growth.

Why Teach Design?

If the purpose of education is fundamentally self-actualization, then we should look beyond the obvious value of problem-solving capabilities in considering the role of design in the curriculum. We have a tendency to focus on the more instrumental values of education, at the expense of losing sight of its ultimate purpose. While design is by no means the only way to introduce learning from experience, creativity, or playfulness into education, these aspects of design are significant and should not be ignored.
This is particularly important since these aspects of design education are also traditionally the most problematic. Although the design studio has become institutionalized within university curricula, its relationship to the broader educational goals of the university have often been challenged on the grounds that it lacks intellectual rigor. The "hands-on" nature of learning from experience is still considered "vocational". The teaching methods are generally ad hoc, and the environment is not "serious". The language is sometimes technical, sometimes poetic, and sometimes utterly opaque. The basis for the evaluation of student work is at times confused because intuitive and empirical knowledge are interwoven in design. These and other criticisms of design education are each in some ways valid, especially the difficult problem of the relationship among different kinds of knowledge in the studio.112 But the greatest threat to the value of design education is the tendency toward pseudo-objectivity as a way of demonstrating "intellectual rigor". Not only does this exacerbate the ambiguity among opinion, fact, and ideology in evaluating design, but it denies the importance of the subjective characteristics of design, such as intuition, empathy, or imagination, that are an essential part—along with reason and technical knowledge—in self-actualization.

Jerome Bruner, in a wonderful essay called "On Learning Mathematics", asks the question, what knowledge is worth teaching? He concludes it is the structure of knowledge that is worth teaching, but most of all, "knowledge that gives a sense of delight,...knowledge that bestows a gift of intellectual travel beyond the information given".113 We should teach design, as we should teach mathematics, to enable students to experience the delight of "learning how to learn", of actualizing their talents, fulfilling their vocation, and accepting their own nature. The challenge of teaching design is to avoid hiding ourselves inside a shell of pseudo-objectivity and instead, as Donald MacKinnon writes, "to encourage our charges by ourselves being those creative persons in whom the opposites of our nature have been reconciled, creative persons with whom they can identify. Thus we each would become an educator in the original meaning of the word—one who brings forth or educes from another that which exists as a potentiality within him through being an example of that which is desired."114
NOTES

70 See Schon's analysis of the epistemology of technical rationality as it is applied to professional practice in *The Reflective Practitioner*.

71 Stringer, "Architecture as Education", p. 20

72 ibid., p. 20


74 Simon, "Understanding Creativity", p. 10

75 ibid., p. 9. The concept of selectivity is similar to Jones' "strategy control" or "meta-language", Schon's "reflection-in-action", and Zeisel's "shifting vision".

76 Simon, *The Sciences of the Artificial*, p. 188.

77 Maslow, *Toward a Psychology of Being*, p. 25.


79 Maslow, ibid., p. 57

80 Bruner, *On Knowing: Essays for the Left Hand*, p. 82

81 op. cit., p. 84

82 ibid., p. 77

83 Archea, "Architecture's Unique Position Among the Disciplines", p. 20

84 Bigge, *Learning Theories for Teachers*, p. 177

85 Polanyi, *Personal Knowledge*, p. 53

86 Zeisel discusses this model at length in Chapter One of *Inquiry by Design*.

88May, "Creativity and Encounter", p. 284.


90ibid., p. 61

91MacKinnon, "Creativity: a Multi-faceted Phenomenon", p. 21

92Barron, "The Psychology of Imagination", p. 155


94See Hillier, Musgrove, O'Sullivan, "Knowledge and Design".

95Simon, op. cit., p. 6.


97Imaging is described in detail by John Zeisel, in Chapter One of Inquiry by Design.


99For "presenting" and "testing", see Zeisel, Inquiry by Design, Chapter One.

100Sutton-Smith and Kelly-Byrne, "The Idealization of Play," in Smith [ed]. Play in Animals and Humans, p. 311

101Donald Winnicott, Playing and Reality, p.110. The concept of design as play is discussed by Beinart in "The Structure of the Content of Design."

102Winnicott, Playing and Reality, p. 64


104Ellis, Why People Play, p. 123.

105Ellis, Why People Play, p. 121.

106Sutton-Smith, "The Playful Modes of Knowing", p. 21
107 Smith and Simon, "Object Play, Problem Solving, and Creativity in Children", in Smith [ed], Play in Animals and Humans, p. 199

108 Quoted from a letter to Jacques Hadamard in Hadamard, The Psychology of Invention in the Mathematical Field, pp. 142-143.

109 Frost, Complete Poems of Robert Frost, p. 359

110 See, in particular, the "goal-insight" theories of learning, which are described by Bigge in Learning Theories for Teachers.

111 Bruner, On Knowing: Essays for the Left Hand, p. 94

112 See Stringer, "The Myths of Architectural Creativity".

113 Bruner, On Knowing: Essays for the Left Hand, p. 103.

114 MacKinnon, op cit. p. 32.
REFERENCES


Rowe, Peter. "A Priori Knowledge and Heuristic Reasoning in Architectural Design", in


Thiel, Philip. "Professional Intention and Public Results – Closing the Gap", unpublished


8. THE DESIGN PROCESS IN MUSIC COMPOSITION

Marilyn Thomas

Design as a Common Process

Composing a piece of music entails a design process not unlike that of building a house or engineering a bridge. On the surface, this statement may appear to arbitrarily link two very different worlds—the creative world of the artist and the more practical, problem solving world of the engineer. To illustrate the similarities in these working processes, let us first consider the architect, who may be more readily envisioned with one foot in each of these worlds.

We can easily appreciate the creative vision of, for example, Frank Lloyd Wright's breathtaking Fallingwater. Yet, we can also appreciate the functionality of this masterpiece as a private residence. Here is a prime example of the creative use of space to uniquely satisfy the aesthetic and practical needs of a client. Every detail of the design contributes to the original concept of integrating the house with its natural surroundings. The specific building materials, the location of various rooms, the style of interior furnishings were all chosen by the architect, so as to best realize this initial concept.

Further, the problems inherent in building the structure had to be anticipated. Obviously, Frank Lloyd Wright did not just decide to build a house with a stream running through it. Every technical problem this caused had to be considered and worked out as the final plans were completed. It is easy to imagine the interaction of creative processes with problem solving procedures in progress as this renowned architect designed his magnificent structure.

This same interaction is less obvious as we turn to the work of a master like Beethoven. Perhaps the fact that the product itself, a piece of music, cannot be put to use in the same way a house or a bridge can be used causes us to regard this work in a different light. We tend to consider all of the arts—music, painting, dance and drama—as creative processes, whose finished products come into existence seemingly without the type of problem solving processes we observe in architecture or engineering.

Yet, the composer, too, begins with a given space and a concept for using that space. Here, the space is a slice of time, an aural space, rather than a visual one. The composer's task, like the architect's, is to use this space to uniquely satisfy the aesthetic needs of his audience. Often, a piece of music is written on commission for a specific patron, or a performing group, or a special occasion. The composer chooses accordingly the instruments to be used, the number of sections or movements in the musical structure, even the types of
rhythms, melodies, and harmonies, the musical materials, to best represent this concept.

Just as specific building problems have to be anticipated by the architect, so do performance problems have to be considered as the composer completes the score. While the architect and the composer are not normally involved in the actual production of the product, the builders and the performers must be able to rely on them to present a well conceived "doable" plan, and quite frequently to oversee the building or performing process itself. Certainly, neither composer nor architect considers the work complete until it is actually realized from the sketches and presented to the people for whom it was created.

The composer's work, then, like the architect's, involves both creative and problem solving procedures. Beautiful music does not just happen in a moment of inspiration, any more than does a beautiful building or a stunningly engineered structure like the Golden Gate Bridge. The initial concept may, indeed, be a vision that appears only to the gifted, but the representation of this concept and the realization of its plan require a thorough knowledge of the specific field and a keen grasp of its tools, whether this field be music, architecture or engineering. The working process of each of these areas of specialization is remarkably similar.

The thesis of this essay is that the design process used by the architect and the engineer is the same process used by the composer. Teaching university students this process and helping them to experience it in a variety of professional areas can be an exciting and effective way to promote cross fertilization of ideas and to develop an awareness of how one's own area of specialization relates to the rest of the world. A core curriculum course in design would be broadened and enriched by including music composition as a component.

Three Stages of the Design Process

The design process, as it is described here, may be regarded as three interrelated stages: first, the conceptual stage, where the architect, the engineer, or the composer considers the goals of the project, the specific site to be used, the materials available, the financial constraints, the people who will experience the finished work, and the visual or aural impact this structure will have on the surrounding environment. As these factors are considered, the designer gradually shapes a concept to satisfy all of these concerns and translates this concept into a vision of the finished form or a part of the finished form. This stage of the design process may be considered the conceptual stage.

A second stage of the design process might be called the representation stage, where the designer translates initial ideas into concrete plans—a blueprint or a musical score. This part
of the process requires extensive and specific knowledge of the designer's field, be it architecture, engineering or music. Here, the engineer may become engrossed in the mathematics and the physics of the problem; the composer worries about the acoustics and the physics of sound. To both, the selection of building materials becomes the primary concern, whether these materials are bricks and mortar, iron and steel, or pitches and rhythmic durations. Specifying these materials precisely on paper and describing how they are to be combined into the overall structure is the fundamental activity of the representation stage.

Since this written plan must be interpreted by others who will actually realize the project, build the bridge, perform the piece, the designer must work in a universal language understood by all in this field of specialization. Much of the education of each professional discipline is concentrated in the area of representation. Whereas, the conceptual stage relies heavily on the individual's innate creativity and perceptual strengths, the representation stage demands a complete mastery of the "tools of the trade".

The last stage, the realization stage, might at first seem to be out of the hands of the designer. After all, it is the performer's task to interpret the score and bring it to life for the listener; it is the building contractor's job to construct the bridge. However, the engineer or the composer must be continually concerned with the feasibility of the plan; a bridge that cannot be built as specified is as useless as a piece of music that cannot be performed. Problems must be anticipated and avoided; the difficulty of producing the concept directly relates to the successful completion of the project.

Whereas, the engineer must know the capacities of chosen materials, how they respond to environmental changes over time, how they react to stress, to intense weight, to rubbing, scraping, high velocity, etc., the skilled composer must know the capabilities of every instrument for which the music is written, their ranges, their timbral qualities in various registers, their dynamic capabilities, their flexibility, responsiveness and the acoustical effects of their blending with other instruments. In short, the composer must be able to mentally simulate the realization of this composition and must avoid writing "unplayable" music, just as the engineer must avoid designing an "unbuildable" bridge. Thus, the implementation of the project must be clearly and accurately simulated in the mind of the designer during this third stage, the realization stage, of the design process.

The Compositional Design Process

Let us now take a closer look at the design process in music composition.

Figure #1 COMPOSITIONAL PROCESS
For the composer, concepts are aural images which must be represented in some standardized form of musical notation; performers then interpret these symbols, recreating or realizing the composer's original intent.

Each of the three stages of the design process embodies a complex mixture of cognitive processes. The conceptual stage may include nonmusical ideas. A composer is often inspired by a painting, a photograph, an experience, a scene, perhaps a poem or even a building. These images must then be interpreted and translated into the language of sound. Sometimes the initial concept for a piece of music is a structure or a form; in such cases, the writer searches for the sounds to represent this structure, to fill out its framework. Other times the impetus for a composition is sound itself—a particular instrument, a melodic or rhythmic idea. The composer, in a sense, paints a picture with a palette of sound colors, writes a poem with rhythm and pitch.

The representation stage involves the most difficult phase of the compositional process—defining the musical concept. What normally begins as a fuzzy nonspecific image must somehow be captured and described in precise musical terminology, so that others who read it can recreate the sounds just as the composer intended them to be heard. The choice of appropriate symbols to represent every aspect of the sound experience—frequency, duration, intensity, and timbre—requires a finely tuned inner ear and an extensive knowledge of musical language.

The realization stage requires the composer to anticipate the eventual performance of the work, and make decisions regarding this performance. The orchestration of an idea, or the assignment of a part to a particular instrument is, in essence, a part of the realization stage. Often, the composer simulates the actual performance at the piano to test specific harmonies and combinations of melodic lines. But most musical decisions are made internally, depending upon the composer's previous aural experiences to guide the selection of timbres.

The compositional design process, however, is far more chaotic and unpredictable than this overview suggests. In practice, all three stages of the compositional process are usually in progress simultaneously. It is rare for a composer to conceptualize an entire piece, notate the complete score and then hand it to the players for performance. Typically, the composer first develops a small musical idea (conceptual stage); often this idea is tested on the piano (realization stage), perhaps modified as a result of hearing it (conceptual stage again), and finally notated (representation stage).
Sometimes the process of developing an accurate representation helps to define the aural images themselves, feeding the initial idea with new identity.

Each phase of the process, then, helps to define the ideas more specifically and helps to develop these ideas into a musical structure.

What begins as a nonspecific concept becomes more and more focused as it is fed by the realization and representation processes. For most composers, the realization process (or a simulation of the actual realization) plus the representation phase are totally integrated into the conceptual process. Few composers work as Mozart purportedly did, hearing the entire composition before writing it down.
Teaching Composition

But if music composition is to be used as an ingredient in a core curriculum course on design, it is essential that the process be taught to persons with little or no musical background. Is this possible? To answer that question, let us first consider the way in which professional composers are trained.

Training the Professional Composer

Effective teachers of composition focus on the process, not the product. Composition teachers' primary responsibility lies in helping students learn to express themselves through the medium of sound. Teachers, who through their own work as composers have already gained the aural experience and expertise needed to know how their students' writing will actually sound in performance, act as mirrors, reflecting back to the students the ways in which their music may be perceived differently than intended. Through discussion, the students' desired musical statements are kept in front as the goal; compositions are finished when the music written achieves this intended expression. In short, composition teachers help students learn how to say what they want to say as effectively as possible using sound as a means of expression.

Much is learned through the study of music of the masters; students listen to designated works while studying the written scores to learn how master composers notate their musical expressions. Yet, composers' skills are developed primarily by doing it; they learn to write by writing and by hearing the results in performance.

In order to compose music, then, students need tools for the accurate representation of aural concepts, guidance in refining the concepts themselves, and skill in simulating the realization of ideas. The technical expertise needed to convert aural concepts into written musical notation is extensive and requires a great deal of specific musical knowledge. Much depends upon the study of music courses other than composition itself.

Composition students learn the language of music in courses such as music history, harmony, counterpoint, form and analysis, ear training, and orchestration. They learn about performance through keyboard instruction, participation in ensembles (choral or instrumental), conducting lessons, and as much exposure as possible to the techniques of each family of instruments (string, brass, woodwind, percussion.) All composition students complete a rigorous program of study—listening, analyzing and writing music of various periods, genres, and styles.
Teaching the Neophyte

To what extent could a student without this extensive musical training benefit from exposure to the design process in composition? If we consider any of the other arts—painting, dance or drama—we find the neophyte may experience the creative process quite naturally as an introduction to that art. When college students take a beginning course in art, they are given the materials, taught how to use them, and guided through the process of creating their own work. While the masterpieces of the past may be referenced and are certainly the focus of courses in art history and art appreciation, the emphasis of the studio course is on developing the students' own creativity and self expression. As the neophytes' use of the materials becomes more assured, their artistic work grows more skillful.

It would be a strange course, indeed, if beginning art students were asked to spend a semester learning how to recreate the Mona Lisa. Yet, this is precisely what we do in music. The beginning courses in music, available to nonmajors, are too often limited to music appreciation courses, music theory, and music history, all of which focus on the music of the masters. Students who actually want to participate in music may be permitted to sing or play in a departmental ensemble, once again performing the music of other composers.

If students are lucky enough to be in a university where private studio is offered to nonmajors, we might expect that here they would be given the skills to develop their own musical creativity and self expression, as the art students do in their studio courses. But once again, students are taught to recreate music written by the composers of the past. Instead of providing beginners with the tools for musical experimentation and encouraging them to find their own styles of musical expression, we sit them down with complex machines and teach them which valves to depress, which keys to strike, which holes to cover to reproduce the notes someone else decided sound "good". The idea of making their own music usually comes years after they have learned to play an instrument, if at all. Whereas, we may teach students to create their own art before introducing them to the masters of the past, in music, we teach them to recreate the music of the masters long before encouraging them to compose their own.

There is a reason for this: the representation of music through standard musical notation acts as a barrier blocking the creative process. Even composition majors, who typically enter the university with years of musical training behind them, find the act of notating their ideas often inhibits the flow of the very ideas they seek to notate. Most young composers wrestle with this problem, trying to get their musical concepts down on paper before they forget them. Sometimes, they can improvise freely at the piano, but get stuck trying to figure out what they
played. Other times, they can describe vividly what they hear in their mind's ear, but don't know how to capture these nebulous sound ideas on paper. Usually, the written version falls far short of the intended musical statement. The transformation of aural concepts into written symbols which others can interpret as sounds, the design process described at the onset of this paper, depends upon intense, specialized musical training.

Yet, there is a growing number of what may be called musical "Dabblers" in our society today. Perhaps because of the difficulty of earning a living in the arts, many musically talented students choose other vocations while maintaining a high level of interest in music. Most of our music schools today are opening their doors to these nonmajors, who elect courses in music appreciation, music theory, and music history, often studying an instrument privately and participating in university ensembles, both choral and instrumental.

The importance of this development should not be underestimated. The arts depend upon an appreciative audience for their very existence, and a knowledgable listener has the capacity to enjoy the finest music presented in the concert halls. The question remains, how can we enrich these musical dabblers even further, so that they might experience the entire creative process in music as they may in the other arts?

The first step is for professionals to appreciate the importance of nurturing all levels of participation in the art of making music; creativity must be encouraged at least as strongly as is the study of an instrument. For once a person begins to discover music as a means of self expression, there is a strong incentive to gain the specific knowledge needed to represent future ideas. The creative process whets the appetite for more knowledge and greatly increases a person's understanding and appreciation of the music of the past. If music is one of the creative arts, then students should certainly be encouraged to experience it through creating their own music.

But this can only be achieved by finding ways to circumvent the problem of music notation. Many dabblers have original musical ideas but get stuck in the process of writing them down, just as composers get stuck. The conceptual stage of the design process may be quite sophisticated, but lacking extensive musical training, beginning students cannot move into the representation stage and are, therefore, prevented from realizing their concepts in performance. The solution to this problem is not immediately apparent, but clearly the type of representation used must not impede the compositional process.
Alternative Forms of Representation

The second step, then, in enriching musical dabblers is to develop alternative forms of representation. Such a move is certainly not without precedent. Our present system of musical notation has evolved gradually over time from a much simpler form, neumatic notation, which in its earliest stage described pitch alone (650–1350 A.D.) Mensural notation (measured notation including symbols representing rhythmic duration) came later (1250–1600). Even the use of five lines per staff to represent pitch emerged from a more direct system called tablature notation, in which lines represented the strings of an instrument, and numbers placed on these lines instructed the player to press a particular location (fret) on a specific string to produce a desired frequency (pitch).\(^{116}\)

Although we in the western hemisphere tend to believe our system of notation is THE way to represent musical sound, other cultures have devised completely different systems.

For example, as described in the New Grove Dictionary of Music and Musicians, "the Javanese use a graphic notation showing pitch on the vertical lines and time on the horizontal. Heavy dots mark this intersection of pitch (actually the place on the instrument) and time (the diagonal strokes to the left of the dots indicate notes in a lower octave).

**Figure #6 Javanese Notation**

Graphic materials are used also at the right-hand edge of the grid to indicate the principal drumbeats, to be performed with the notes.

**Figure #6a Drum Beat Graphics**

But to the left of the column appear shapes—Javanese characters—which represent syllables that in turn are abbreviations for words: ta for ketuk, na for kenong, ga for gong (combined together as na–ga) and wa for wela (figure #6b). They refer to the colotomic structure of the composition, the first three being colotomic instrument names, and the fourth a beat’s rest."\(^{117}\)

**Figure #6b Structural Syllables**

Furthermore, some music has survived the test of time without ever being written down. Our vast repertoire of folk music from various countries is a superb example of unnotated music, as is the highly developed music of the Indian raga.

Even in our Western culture, musical notation continues to change and evolve as composers find new ways of expressing themselves with sound. Many twentieth-century
composers have been forced to devise their own means of specifying sounds, because traditional metric relationships are not always employed, aleatoric passages may require less specific descriptions and extended instrumental techniques often produce sounds with complex timbres and pitches falling between the cracks of our equal temperament tuning system. Representative samples of graphic notation may be seen in the electronic music of such composers as Karlheinz Stockhausen and George Crumb. A particularly striking example is Ligeti's "Artikulation"; a listeners' score for this work features a graphic representation of the various types of electronic sounds in the piece using several contrasting colors.

In the same article on notation mentioned above, from the New Grove Dictionary of Music and Musicians, notation is defined as "a visual analogue of musical sound, either as a record of sound heard or imagined, or as a set of visual instructions for performers." The idea that there is a single, "correct", way to notate music is totally false. As long as the system of notation used can be read and understood by those who seek to perform the piece, it has served its purpose.

Professional composers, it is true, must master the traditional Western notational system with all its complexities, as well as the more contemporary forms of graphic notation, for these are the tools of their trade, just as interpreters of foreign languages must learn every nuance, every facet of the languages they will use in their life's work. Tourists, on the other hand, can experience much of the flavor of a foreign country without a thorough grasp of the native language; they learn to communicate adequately with hand gestures and facial expressions augmenting the key words they have learned.

Is it not possible that music, often called the universal language, can also be experienced without getting stuck on a particular form of notation? If we are sincerely interested in teaching the design process in music composition to nonmajors, we must not be deterred by surmountable matters of form. Let us use whatever type of representation best serves our purpose.

The Role of the Computer

This brings us to yet another potential source of help: the computer. In recent years, the computer has become an effective catalyst in involving the musically untrained in the compositional process. Since the 1950's, when Max Mathews of Bell Labs designed the digital to analog converter, computers have been capable of making sound. Throughout the 60's and 70's, the complexities of the music languages, such as Music V\textsuperscript{119}, Music 1\textsuperscript{120}, and "C" Music\textsuperscript{121}, needed to translate this sound producing capability into machine code, discouraged all but the most stalwart composers. Thus, much of the music throughout these decades was
created by scientists with musical skills rather than by musicians with computer knowledge. It was definitely easier for the scientist to cross over and dabble in music than for the musician to dabble in computer science.

But much has changed in the past five years. It is no longer necessary to study complex programming languages to make music with a computer. The smaller, less expensive microcomputers of the 80's often come equipped with their own sound producing capability. Through the industry standard MIDI interface, an acronym for Musical Instrument Digital Interface\(^2\), most computers can be connected to musical synthesizers with the digital to analog conversion process already programmed in by the experts. A user may simply indicate the desired pitch, duration, intensity, and timbre, and the computer sends the appropriate commands to the synthesizer. Sounds are heard through internal or external speaker systems. With the sound producing capability well established and readily available to both the musically naive and to the technological novice, virtually anyone can make music on a computer—perhaps not sophisticated music, but music nevertheless.

Overview of Music Software

Now, software is being produced to take full advantage of this capability. Some music education software deals with the teaching of music notation and fundamentals. In 1974, the University of Delaware pioneered in developing its Guido music learning system for the micro Plato display.\(^3\) Other programs have followed, written for the Apple IIE, IBM PC, Commodore, and more recently for the Macintosh. Most help the users become familiar with the vocabulary of music—names of pitches written on the staff, the basic relationship between various note durations, the meaning of key signatures, meters, intervals, scales, etc.

With this software, it is now possible to teach oneself how to read music and even to hear the musical elements as they are learned. In this way, the dabbler can acquire some of the skills normally taught to a professional musician—skills that definitely help during the critical representation stage of the compositional process. This type of software falls into the category CAI (Computer Assisted Instruction). Some of the newest CAI software, developed at Carnegie Mellon, "Camus"\(^4\), by Colette Wilkins, and "MacVoice"\(^5\), by Marilyn Taft Thomas and Peter Monta, focuses on the next stage of theoretical training, where students learn to use these fundamentals of music to develop aural musicianship and four-part writing skills.

There is another form of software on the market that facilitates the printing of musical scores and parts. The most ambitious in this category presently is Mark of the Unicorn's "Professional Composer"\(^6\), which permits the user to write entire orchestral scores, using the computer to store the music, edit various aspects of the piece, play it back, and print either
individual parts or the entire score. This program and others like it, ("Musprint" by Keith Hamel, for example,) serve the needs of composers who want professional looking printed scores without doing their own calligraphy. Since the computer only prints what the user selects, this type of software depends upon a person's mastery of notation. It is not recommended for the musically untrained.

Yet another type of music software focuses on the performance aspect of the compositional process. "Concertware" by Greatwave Software, is a current example. The user may compose sounds by actually designing the instruments to be used, building timbres, and describing the envelopes and relative intensity of the fundamental and each partial to be included. The computer then performs the piece through a synthesizer. This type of software automates the realization stage of the compositional process; it is an excellent aid for students who lack the skill to perform well enough to test their own ideas.

It is also possible to have automatic representation through today's computer software. Many programs are now available that permit the user to compose music without worrying about traditional notation. Some, like "Music Works" by Hayden Software, uses a simpler form of graphic notation which may be converted by the program into standard notation. Others, like Mark of the Unicorn's "Performer", permit the user to play on a musical keyboard; this performance is recorded digitally, translated into machine language and transcribed by its companion "Professional Composer" into traditional music notation, which may then be printed.

The computer may well be the catalyst needed to teach the design process in composition to nonmajors. To summarize, we have available at least four types of software programs: those that teach musical vocabulary and the basic symbols used in traditional Western notation, programs that produce professional looking scores and parts from traditional Western notation, programs that perform the music placed on the screen that is represented either graphically or traditionally, and those that automatically notate, again either graphically or traditionally, the music that is played on an attached musical keyboard.

The first two types of software, computer assisted instruction and document production, could prove very useful if traditional Western notation is employed in the core course; however, as stated previously, the extensive body of specific musical knowledge needed to work effectively with this form of representation makes its use by nonmajors in a course on design extremely problematic, and perhaps even self defeating.
The other two types of music software, those that automate either the performance of the music (realization stage) or its notation (representation stage), appear to be the most promising. Here is a method of introducing the dabbler to composition, without requiring a grasp of musical notation. The problem is that in taking this approach, the user participates in only one or two stages of the design process, the conceptual stage and either the realization stage, if notes are played on the keyboard, or the representation stage, when notes are selected graphically and played automatically. By reducing some of the drudgery, we might unwittingly eliminate critical learning experiences associated with the design process. Students, for example, could conceivably make random marks on a graphic score, hear the musical results and print the score without coming to grips with any of the creative or problem solving issues inherent in composing a piece of music.

These programs have tremendous potential for increasing the level of participation by the musically untrained and could, in fact, make a significant contribution to the general public's interest in music. Yet, their value is questionable in a course specifically geared toward developing a grasp of the design process in various professional areas, because they actually exclude students from the central portion of this process. By solving the representation and/or the realization issues for the users, only the conceptual stage remains.

This is certainly not a condemnation of the software—it's purpose is to permit the musically untrained to experiment with aural ideas, to see them in print and to hear them in performance. The purpose of the course in design is, of course, quite different. Whatever the approach, it is imperative that the design process be preserved and experienced fully by the students. How, then, might we proceed?

Solving the Representation Issue

Recognizing the representation stage as the problematical area for the musically untrained, three possible solutions may be considered: 1) teach the musical knowledge needed to notate one's ideas; 2) change to a simpler, more general form of music notation; and 3) avoid the problem altogether by using computers to notate the concepts for us. The first choice, while certainly the most desirable from an educational standpoint, is not feasible in a brief course designed for nonmajors of many areas of specialization. The third choice, while potentially effective in raising the level of participation and appreciation of music, as has been pointed out, is not suitable for teaching the complete design process. (This is not to say, however, that computers should be avoided altogether. Their effective use in a design course with a music composition component will be proposed at the end of this essay.)

The best solution to the representation problem seems to be the use of a simpler form of
notation. Since our Western staff notation is not the only system used to represent music anyway, why insist on its use to teach the compositional process? If the goal is to teach the design process in music composition, we want students to experience the problem of defining, organizing and representing aural concepts on paper.

The Quaker Valley Model

Let us consider an example of how this might be achieved. In 1981, the Quaker Valley School District organized a special program called "Arts in Education" for high school seniors enrolled in the GATE (Gifted and Talented Education) curriculum. Students who chose music as their primary interest had diverse backgrounds, ranging from nine years of training on an instrument to a vague interest in rock with absolutely no knowledge of musical notation. To accomplish anything beyond the music appreciation level in four sessions, especially considering several weeks lapse between meetings, required an approach which could be used in the design course discussed in this paper.

In this experimental project, students were given paper and pencil and asked to listen carefully to the sounds of the room, the adjoining hallway and the outdoors. During a timed period of one minute, they were to write down everything they heard. A stop watch was used and students were given a signal to begin and end so that everyone recorded the same one minute period.

Once this was completed, the students were reminded of their introductory discussion on "What is music?" during which they had been taught some basics on the acoustics of sound. In this first session, they had also been introduced to the concepts of John Cage, who in the 40's defined music very broadly, including noise as well as "so-called musical sounds," and to others, like Edgard Varese, who in the early twentieth century had stated simply, "Music is organized sound." The objective of that introduction was to open their minds to a general, more universal concept of music, to challenge their narrow view of music as the "top ten on the charts" or that of the Western classical tradition.

Referring to this discussion, it was a small step toward accepting the one minute of sound they had just experienced as a potential piece of music, which could be notated and recreated in a concert. But first, the sounds had to be arranged into some sort of musical score. A group discussion followed, resulting in a list written on the blackboard showing all the sounds that had occurred during the one minute period, in the order in which they were heard. Some sounds overlapped, some were sustained during the occurrence of others. All of this was established through sometimes heated discussions between students who had recorded the sounds in a slightly different order. The sound events ranged from a dog barking outside, a chair
being dragged across the floor in an adjoining room and conversation in the hallway, to the lighting system of the room itself and the scraping of their own pencils recording the sounds they heard. Occasional laughs and coughs punctuated the overall atmosphere.

Once the soundscape was thoroughly described and the group satisfied that their list was accurate, a discussion was held about the timbre of the sounds and the best way to recreate them using either musical instruments or nonmusical objects. Students who played musical instruments were encouraged to recreate some of the needed sounds on their specific instruments, and were asked to demonstrate this for group approval. Everyone was encouraged to think experimentally—perhaps using mouth pieces alone or overblowing, etc.

This led to a brief explanation of the use of such extended instrumental techniques in today's new music by composers of both jazz and "serious" concert works. Some sounds, it was discovered, could be recreated vocally or by clapping or tapping an object in the room. Each sound on the list was carefully considered in terms of its main elements: pitch, duration, intensity, and timbre.

As the piece was orchestrated, the instruments needed were written beside the sounds on the list. Relative dynamic levels were also established using the standard musical terminology ranging from pp to ff. Accents were marked \( > \). Gradual dynamic changes were shown by \( \) and \( . \)

Before concluding the session, students were told that they had just experienced part of the compositional process. Composers listen to the sounds of their own imaginations and attempt to represent these aural concepts just as they had listened to the sounds in the room and were attempting to represent them. During the next session, they would learn to create a musical score.

Session #3 was an ideal time to introduce various types of notation. Graphic scores were shown and parts of some electronic pieces were played so the students could see how various sounds were notated by other composers. A brief description of the Western tonal system of notation as well as the more contemporary graphic systems helped to illustrate the need for different types of notation for different types of sound.

Using a time line, students were asked to figure out roughly where their sounds fell within the one minute span of their piece. The onset time of each sound event was given a specific number (11.5 sec.) and the release of this sound was also marked (13 sec.). It didn't matter if a sound was placed a second or two away from its original sounding; it just
had to be placed somewhere; a specific decision needed to be made so that a written score could be prepared. Changing densities became apparent as many sounds sometimes occurred within a few seconds of one another, while during other time periods only an occasional event took place.

It would have been possible to use a tape recorder to preserve the original soundscape. Students could then have replayed the piece as they sought to notate it. But working solely from aural memory more closely models the design process of the professional composer. It also permitted the students to shape the piece themselves, using the initial listening experience as merely a starting point.

The actual preparation of the score required a listing of the instruments to be used. This was a good opportunity to show the students some standard orchestral scores, illustrating the way in which instruments are universally grouped on the page: woodwinds on top, then brass, percussion, keyboard, and finally strings at the bottom. Each group of instruments, it was pointed out, were ordered from highest to lowest. The necessity of following this standardized format for the sake of the conductor was emphasized. Individual players' parts were also shown, so the students had an overview of the process of preparing the written materials.

Since this group of students would be performing their own work in a public concert at the conclusion of the seminar series, orchestration depended upon "who could do what". A conductor was elected and everyone volunteered for the vocal or instrumental parts they could perform. Some consideration had to be given to simultaneous events, since a single student couldn't play two things at once. But everyone had to do something!

The list of instruments and/or voices needed was then arranged according to the standard orchestral format, and the students were shown how graph paper could be used to represent the piece. (Each square equalled 1/2 second—two squares per second.) Taping pages side by side made a continuous time line. Sounds were placed in whatever instrumental part they occurred, according to the point in time at which they would be played. Students were then asked to work on their conductor's score together during the coming month.

The final session was devoted to a dress rehearsal of the piece, now entitled "Soundscape #1". Time did not permit the preparation of players' parts as well as the conductor's score, so students needed a good idea of the pacing of the piece. Since they had been involved throughout the design process, they already knew the work quite well. At this point, the conductor became extremely important. Cues had to be given clearly; dynamics and tempi had to be controlled carefully. Students argued a bit about these issues and the conductor learned
The logistics of performing the piece presented new problems. Locating an instrument on stage so that the sound would come from a particular place was sometimes not feasible. Methods of muting instruments that were too loud for the rest of the ensemble needed to be devised.

The actual performance of "Soundscape #1" was prefaced by a brief description of the learning process. The performance was not without flaws and the students themselves were a bit shaken by the brevity of the actual product of their labors. Yet, they were also aware of how much they had learned. Through this project they had experienced the complete compositional process—taking an initial aural concept, writing a piece of music based on this concept, orchestrating it, conducting it and performing it. In the process, they learned about critical listening, music notation of various types, the organization of orchestral instruments, and the basic acoustics of sound. Specific elements of sound, like pitch, duration, timbre, and intensity took on new meaning as nonmusical sounds were analyzed and defined. The students also experienced a general broadening of their views toward music: What is music? What is noise? They learned to open their ears and to open their minds to new sounds.

The University Core Course in Design

What was accomplished here in this experimental set of workshops could serve as a model for a university level core curriculum course in design. Four to six classes would be sufficient to lead the students through the entire design process, exposing them to a wide range of musical concepts along the way. The conceptual stage, wherein everyone listens to the same one minute of sound is prescribed so that all students can use the same material and can work together, devising their own method of representation and realization. The conceptual material is, in a sense, standardized so that the students can be guided through the design process as a group.

The compositional process having been experienced, students could then be expected to create their own individual pieces as their final project, using whatever initial musical concepts they select. Several more class sessions should be devoted to listening to each other's pieces and discussing them. Some students might choose to interpret a favorite poem or a painting into a piece of music, or produce a completely personal musical statement as a final project. Some, with a knowledge of standard musical notation, might decide to compose a more traditional piece of music; still others might follow the method devised for the group project. A concert could include these individual projects, performed by friends and/or classmates, or such performances could be done in class with the listeners providing a critique.
Computers could be used to enhance the experience, but should not be permitted to interfere with the compositional design process itself. For example, once the soundscape is organized in time, it would be possible to lay out the score graphically using the computer instead of doing this phase by hand. It would also be possible to translate this graphic representation into traditional notation, via a program like "Music Works", and even to print the finished score and parts using "Professional Composer". Further, a program like "Professional Performer" or "Concert Ware" could be used to test the performance aspects before the finished piece is given to the live players. Computers may help the designer simulate the results, prepare the plan, or even shape the concept; but computers should not be permitted to hide the process or to rob the designer of his freedom to create.

It is important to note that the use of computers in a core curriculum course such as the design course proposed here requires a large number of individual computer work stations readily available to students. While such facilities already exist at Carnegie Mellon, most of these facilities are not equipped to produce sound. "The Musician's Workbench"[^33], currently under development in the Center for Art and Technology, will fill this need. When completed, the "Musician's Workbench" will provide the type of music editing, performing and printing tools needed to compose music at an individual computer. A large number of these music work stations would be needed to support the musical activities of a core curriculum course in design. (A short term solution, using the MacIntosh clusters already on the CMU campus, could be implemented until the "Musician's Workbench" is completed.)

The feasibility of including music composition in a design course does not necessarily make it a desirable component. After all, given the difficulties of the representation stage which must be circumvented and the need for special music work stations if computers are to be incorporated, why bother?

Electronic and Computer Music I

To answer this question, let us turn briefly to a course already implemented as part of the music department curriculum, Electronic and Computer Music I. The emphasis of this course is on composing pieces of music. As the students learn to use the audio and digital equipment in the studio, they complete a series of four individual projects, each one a piece of music composed and recorded in the studio. A public concert of selected student works is presented upon completion of the course.

In this class, the representation problems of standard musical notation are avoided; music is represented either by a computer program or by the recording of the sounds directly from the equipment. Since sounds can be produced in the studio as the composer works, there need
be no score and no performers' parts. In a sense, the making of sounds through the analog or
digital equipment is the representation stage and the actual performance (on tape or in real
time) becomes the realization stage. The composer is also the performer. Thus, the course may
serve as a model for teaching all three stages of the design process with a different solution to
the representation problem than the self designed graphic notation used in the Quaker Valley
Project.

Electronic and Computer Music I is open to all university students and typically draws a
wide variety of students—composers to electrical engineers, dramats to English majors,
undergraduates, graduate students and occasional staff members. Limitations of space and
studio time for individual work necessitate a class size of twelve students per semester; there
has been a waiting list for enrollment in the class ever since its inception in 1982.

The focus of this course lies in introducing students with diverse backgrounds to the
compositional process. Often, students find the dependence upon creativity to be unsettling.
Many, who are accustomed to learning what they are asked to learn, who perform exceptionally
well on tests and who are generally considered to be high achievers in their major subject
areas, respond to the first compositional assignment with pleas for specific instructions.
Questions like, "What do you want us to compose?", "What kind of sounds should we use?"
and "How should we organize the sounds?" are familiar ones to those who have taught the
course. Learning to choose their own sounds, design their own musical forms, express their
own concepts is, of course, the goal. The faculty's role, as described previously, is to guide
the students through the design process and help them learn the technical skills they need to
say what they want to say as well as possible.

All projects are played in class and critiqued by their peers. This inevitably stimulates
discussion about "What is music?" Even though the equipment to be used for the first
assignment is carefully defined and quite limited, the diversity of musical approaches among the
students in the class helps everyone see that there are a myriad of acceptable solutions to a
problem in which creativity is a major factor. By the end of the semester, most students have
gained sufficient confidence from the open and supportive classroom atmosphere to attempt a
real personal musical statement for their final project.

This example, Electronic and Computer Music I, is not meant as a model for the core
curriculum course on design, because once again the tools required (the use of complex
electronic and computer equipment) take too much time to learn in one or two sessions. In
fact, it takes at least a semester of work to master studio techniques sufficiently to create a
piece of music. Rather, this course is mentioned here to illustrate the potential of teaching music composition to nonmajors. Such a course fulfills one of the main objectives of a university: to stimulate thinking and to open doors. Students gain new perspectives about their own areas of specialty and a much deeper appreciation of another field of study. They also learn to use their creative skills along with their analytical skills to explore the design process in music.

**Summary**

In conclusion, it is both feasible and desirable to include music composition in a course on design. By adopting a more general graphic system of notation, and by utilizing computers to enhance the experience, even the musically untrained can participate in the entire compositional design process, gaining an appreciation of the art of music and perhaps awakening an interest in learning more about it.

One of the primary objectives of a core course in design, moreover, is to help the student see similarities in the design process between different fields of science, art and the humanities. Creativity is needed in the conceptual stage of any design process, whether it be the design of a new building, a new bridge, a new painting, or a new symphony. Skillful representation of ideas in blueprint form, on canvas, or in a musical score is necessary for the successful realization of the design in brick or steel or paint or sound.

Design is a way of thinking, a way of working, a way of creating, which we all share. Teaching the process means teaching students to appreciate the universality of design itself.
NOTES

115 Fallingwater was built in 1935 as a residence for Edgar J. Kaufmann, Sr. in Ohiopyle, PA. The Western Pennsylvania Conservancy conducts guided tours of this site.


118 Ibid.

119 Music V was designed by Max Mathews at Bell Telephone Laboratories, based upon his previously designed language, Music IV. Music IV was the first computer music language, written in 1957.

120 Music 11 was developed by Barry Vercoe at MIT. It evolved out of Vercoe’s Music 360, a high-speed music program written in 1969.

121 "C" Music was developed by F. Richard Moore at CCRMA (Center for Computer Research in Music and Acoustics) at the University of California, San Diego.

122 Further information about MIDI may be obtained from the International MIDI Association, 11857 Hartsook Street, North Hollywood, CA 91607

123 "Guido" is an acronym for Graded Units for Interactive Dictation Operations, developed by Michael A. Arenson and Fred T. Hofstetter.

124 "Camus" is a computer-assisted instruction program that provides drill-and-practice exercises in solfege, an ear-training technique for accurately transcribing music as it is heard. "Camus" runs on the IBM PC and the Apple IIE and is available from Conduit Software Company, University of Iowa, Oakdale Campus, Iowa City, IA 52242.

125 "MacVoice" is an interactive program for the Macintosh that assists music theory students in learning to write four-voice chorales, based on 17th and 18th century practices. It is available through Kinko’s Academic Courseware Exchange, 4141 State Street, Santa Barbara, CA 93110.
"Professional Composer" is a music editing application program for the Macintosh produced by Mark of the Unicorn, Inc. 222 Third Street, Cambridge, MA 02142.

"Musprint" is a music editor for the Macintosh available from Keith Hamel, 466 Albert Street, Kingston, Ontario, Canada K7L 3W3.

"Concertware" consists of three programs: Writer (a music editor), Instrument Maker (an interface to the built-in synthesizer), and Player (for playback of works created by the Writer). This package is available from Great Wave Software, P.O. Box 5847, Stanford, California 94305.

"Music Works" permits the composition of music using graphic or standard notation, playback, editing of waveforms and printing of music. It is available from Hayden Software, 600 Suffolk Street, Lowell, Massachusetts 01853.

"Performer" is a MIDI sequencer, editor, and performance tool for the Macintosh 512K computer. It is available from Mark of the Unicorn, Inc., 222 Third Street, Cambridge, MA 02142.


The Musician's Workbench Project is described thoroughly in the soon to be released book, Advances in Computing and the Humanities, edited by Ephraim Nissan, JAI Press, Inc. 36 Sherwood Place, P.O. Box 1678, Greenwich, Connecticut 06836.
References


9. REFLECTIONS ON THE EXPERIENCE OF DESIGN

J. M. Ballay

It seems that a common pastime of industrial designers and graphic designers is to invent new definitions of design, each hoping to arrive at the statement that will outlive him. None of these definitions has ever really achieved immortality, except maybe Louis Sullivan's "form follows function" or Mies van der Rohe's "less is more" and these are both defensible only in the right circumstances. My own definition of several years' standing is "design is the search for essential form". Admitting it needs it's own right circumstances, I have found this definition to stand me in good stead and I will use it as a point of departure for this paper.

Search for Essential Form

Calling design the search for essential form immediately stresses action - the action of searching. It is the verb design, the act of designing that is being defined. In my dictionary, the first meaning of the word "search" is both "thorough examination" and "exploration"... that fits. The adjective "essential" is intended to convey two of its meanings: essential - meaning the indispensable part, and essential - meaning the part that carries the intrinsic qualities (essence) of a thing.

The object of the search is "form". Form should not be confused, as it often is, with the simpler word "shape". Rather, form is an inclusive word which conveys a larger concept of visual organization, the cumulative effect of structure, shape and surface qualities. Form is the property of an object or concept which gives it identity. The essential form of "table" involves a flat top surface (usually level), a size that is within human scale, and a determined relationship to the "floor". To become a particular kind of table, a "Parsons Table" for instance, the form has to acquire a particular rectangular geometry and proportion, and a uniform finish.

By this combination of meanings then, design is a process of exploration and examination by which one arrives at the visual organization of a thing, which is required to reveal its intrinsic qualities.

If we accept this informal definition of design we are plunged directly into two issues with which all designers deal: how to conduct the process of search, exploration or examination; and how to recognize an essential form when you see one. I will argue that a science of design, as it seems to be emerging, deals with the search process but not with the recognition. I will also offer an approach to the recognition issue which I have found to be pragmatic and productive in the context of my field, industrial design.
Analysis of the Design Process

Since approximately the nineteen-sixties there has been a steadily building body of experimentation and writing regarding the aspects of designing that seem to be knowable as a science. These have mostly been methodological studies and the major authors have been people like Christopher Alexander, Bruce Archer, and Christopher Jones. Buckminster Fuller went so far as to proclaim the beginning of a new discipline which he called "Anticipatory Design Science". It attracted attention principally because of Fuller's own provocative and persistent personality. It resulted in a variety of "Dymaxion" objects: cars, houses, etc. – all of which were one of a kind – but it was a sterile movement in terms of it's effect on the way design is practiced.

At Carnegie Mellon we know a fair amount about how the design process progresses. In some work that Dick Hayes and I did for IBM, we studied videotape protocols of a designer as he was designing a new product. A brief summary of his process would characterize it as a "cooperative game" involving both the designer and the client and would categorize design as a complex, "ill-defined construction task". The task leads to an end product that is to be organized or constructed out of elements, but it is ill-defined in that no particular end product is specified. There is not a unique solution. Rather there is a set of requirements which could be met by a variety of designs. Some of the requirements, such as the function of a proposed product, are part of the problem statement. Some, such as the structural limitations of materials, are part of the real world context of the problem. And some requirements, principles of visual aesthetics for example, are brought to the problem by the designer.

The number of requirements is large enough, by far, for the problem to be considered complex. It is clear from the protocols that genuine optimization is not considered, even if it were possible. Instead there is a focusing on a few key requirements and on the generation of a few alternative configurations that satisfy those requirements. The alternative solutions are tested against a broadening view of the problem, choosing the most promising one and taking the opportunity to satisfy as many additional requirements as can conveniently be done without compromising the underlying concept of that solution. Only the best of the alternatives is refined; the rest are left incomplete.

Overall, the experience of designing doesn't seem to be as programmatic as the description above might imply. It often takes on the qualities of creative play, a sort of game. It imagines a situation that might be, complete with rules for the game and a set of relevant principles, both known and imagined. Then this scenario is played through to be come real, at
least temporarily in the experience of the designer. As in other games, unexpected things may happen. Someone throws that first forward pass; is it illegal or a creative redefinition of the game? It is the role of the designer to take advantage of the unexpected.

The play is directed toward a goal but not bound by a specification. It has purpose but recognizes that the purposes are complex, and that there are many ways to achieve those purposes such as adopting unexpected and seemingly purposeless mutations. It is the purposeful intertwined with the purposeless.

The results of this game are concrete representations of possible realities. They must have form in our earlier sense of the word, for without form they can not be perceived or transmitted to the others who cooperate in the design process. So the making of representations becomes extremely important. If the designer is handicapped in his ability to make concrete representations from abstract concepts, those concepts are eventually aborted and never reach visible reality.

Bruce Archer has focused on the importance of making. He has argued that designing should be thought of as the "fourth R", for "wroughting" (with apologies for the phonetic word play) and suggested that the process of wroughting or making could be studied and described in a way that would be helpful to those who want or need to design something. I believe Archer is looking in the right direction because it leads to understanding based on making as the externalized experience of design. If we do not deal with the experience of designing, our understanding of design lacks a major part. So, if we understand science to be an important way of knowing, but not the only way, it becomes important to match carefully the things to be known with the appropriate ways of knowing them.

Three Levels of Knowing

John McQuarrie defines three levels of knowing, each deriving from a different relationship between the knower and the person or thing to be known.

The most common level is objective knowing. In such knowing we transcend what is known, mastering it in the process. The knower is the active agent an the object of our knowledge is for the most part passive. The processes by which we arrive at this knowledge include observing, experimenting, measuring, and also deducing and demonstrating. This is the level of knowing that includes both high technology and even the low technology we use to get through our daily activities.

At the other extreme is revealed knowing. In it the initiative passes to the thing which
is known so that we are seized by it and it impresses itself upon us. The process by which it is achieved is more contemplative in character and contrasts with the probing of objective knowing. It accounts for the experiences of revelation upon which are built the great inspirations of religion, poetry or music.

Between these lies experiential knowing. It may be characterized as a kind of dialogue between the knower and the thing to be known. The object of our knowing is thought of as having an existence of its own — a thing that is or wants to be. Our part as the knower is to bring it into the realm of human experience through the dialogue. The process is one of participating with the object to be known or "thinking into it".

These levels, of course, are not exclusive categories; any area of human knowledge may reach into all three. I use them as a framework for considering the nature of designing. With respect to these levels, we can see some areas of emphasis. Technology lies largely at the level of objective knowing. Religion depends for its essence upon revealed knowing. Design, I claim, principally involves experiential knowing and is a rich embodiment of the thinking and doing which constitute that level.

Design as a Way of Knowing

At its most creative moments, designing focus thought on what might be or ought to be, not on what is or already exists as a concrete reality in human experience. The object of design knowing exists as a potential. It is a set of relationships which can be brought into being as an essential form. To come into being requires participation or involvement on the part of the designer rather than just objective observation and description. It is a dialogue, a series of exchanges or transactions between the designer and the design. Through this give and take the form possibilities develop some boundaries. To be sure, there are points in designing which require objective processes, but they are not the points at which designs are brought into being for the designer and his colleagues to know them. In my experience as a design teacher, this is one of the most frustrating traps into which students are prone to fall. The student seems to be making great progress in understanding the problem — understanding what is. Rational observation and description have initially yielded understanding about the circumstances that define the design problem (but not the form which represents a design solution) and the feeling of progress in understanding has become its own reward. So, when the time comes to get involved with the creative consideration of what might be, the student recoils away from the experiential dialogue and into ever finer observation and description because that was the path to past success — and eventually fails to find a creative solution to the problem.
Recognizing An Essential Form

The full range of design experience is not limited to objective and experiential knowing. There is some of the quality of gift that typifies knowing by revelation. There is a well known point in creative activities (creating visual images in my experience) at which the image "talks to" the designer. It is a common strategy in studio design classes to encourage students to hang their partially completed work on the wall and study it for an extended time. They are encouraged to let the work "tell them" what it wants to be rather than to indiscriminately impose their will on the work. This strategy is only appropriate at the refinement stage of the design process when there has been enough dialogue to produce several alternative preliminary designs. This listening to the voice of the work is not as mystical or undisciplined as it might sound. It is a time in the design process when previously assumed requirements or objectives should be reconsidered. Any incomplete representation of a design can be thought of as a node in the process with an opportunity for branching. The work can continue to develop in the direction dictated by the original requirements and objectives or it can take a new direction based on new information. The representations of alternatives are, themselves, a primary source of such information.

What is looked at or listened to, of course, is evidence of the essential quality of the form. That evidence is the simultaneous fit of many and diverse requirements, perhaps using parts of different alternatives. It is an unforced quality, a kind of economy of form that is achieved without ignoring essential requirements. This fit is not necessarily to be seen in the direction that a particular solution is currently taking; it might just as well require a major reorientation. But such a change will not be understood unless the designer can get his preconceptions out of the way and listen for the work to speak for itself.

An Example: The Cardboard Stool Problem

I have given the cardboard stool problem to beginning design students for many years. It is one of my favorite problems because it illustrates so many issues that have been discussed here, but does so at a simple enough level that students from a broad range of backgrounds can contribute to the class work and derive useful information from it.

There are some variations of the problem, but basically it is this: Design a body support device (stool) that will support your weight 16 inches off the floor. The device is to be made from one square yard of ordinary corrugated cardboard (36 inches square, available in our Art Store for about a dollar per sheet) no other materials, additives, fasteners etc. can be used. You should try to minimize the amount of cardboard that is used, yet your solution must hold your weight in reasonable comfort with a reasonable
margin of safety (to test this you will be asked to sit on it while holding your feet off the floor). Of course, the form (structural organization, shape, detailing, and craftsmanship) is a major consideration and should be of the highest quality that is consistent with the materials and tools (ruler, matknife, etc.) that are available to you.

The problem presents many opportunities to talk about the design process. The facts of the situation are pretty simple. They are mostly given in the problem statement. If anyone wants more information about the structural capacity of the cardboard, a simple desk top experiment can provide empirical data. There are some simple techniques to be learned, again empirically, about cutting and folding corrugated cardboard (the wroughting part of the problem). Then the fun begins.

It turns out that for most students the structural aspects of the problem are easy to solve. The cardboard is quite strong at that scale. For average sized students, the most efficient solutions consume only about one-half to two-thirds of a square yard.

Photographs of cardboard stools can be inserted here

The real problems are in finding the right (essential) form. First the student has to get past a lot of preconceptions about what a stool ought to look like – that it ought to have three or four legs and a round seat, all of which are difficult to achieve within the constraints of the problem. Often there is also a cultural bias against letting the joints show; it seems to imply to some that the work is unfinished. Finally there is a competetiveness among the students that makes them want to produce a solution that is "different". Lots of bizarre forms are tried and eventually abandoned in the striving for difference.

There is the opportunity to explain that when a problem is constrained like this one is, the solutions will tend to converge to a narrow range. The real opportunity for difference is in the details, the subtleties of proportion, a joint that works structurally and looks neat.

Most important to me, there are many opportunities to encourage the experiential dialogue between the designer and the designed object. It works very well. For most of the students who are given this problem, it would not do much good to suggest an abstract analytical approach, they don’t yet have the repertoire of analytical tools. But in working with an actual stool it is easy to see how the orientation of the material and the proportions of the joints affect its structural integrity and its appearance. As the essential form begins to emerge
through alternative solutions produced by the class, students remake their designs, responding to
information from their stool and other stools they see; information about ever finer nuances of
proportion, detail and craftsmanship. Most students make about five stools before they are
satisfied that they have found an essential form.

Three Roles of Information in Designing

Problems such as the cardboard stool require students to encounter a variety of
information from a variety of sources - from the instructor, from the task environment in
which they are working, and from the emerging design solution itself.

As the students' design sophistication grows it becomes appropriate to point out that
designs and designing can be thought about from the point of view of the information that is
involved. This view is compelling considering the directions that the professional practice of
design is taking and particularly in the context of the intellectual niche that Carnegie Mellon
has adopted. Consequently, the concept called "information design" has been adopted by the
design department as a leading part of its intellectual thrust.

There are three ways in which information concepts are embedded in designs and the
designing process. In the information design concept I describe them as information through a
design, information about a design and information in a design. At the various points in the
lifecycle of a product or image, each of these can be important to its designer, its producer,
or its user.

Information through a design recognizes that so many of the products and images that
designers now work on are concerned with the delivery of information from one person or
machine to another person or machine. In graphic design the situation is obvious - from
posters to the interface screen graphics on computers. Now, even the industrial designer is
finding information principles are important to the success of the products he designs. A
multitude of new microprocessor based products are essentially boxes of electronics which look
very similar to the casual user - a TV set looks a lot like a microwave oven to me. Once
I've decided whether the product lets me watch the evening news or warm up some leftovers,
it is ease with which I can get information into or out of the control panel which makes the
product satisfying to use.

Information about a design concerns the multitude of information that a designer
processes in solving a design problem. It includes project proposals, specifications about
materials and components, and the documents that a designer generates as his main tangible
product. These documents are the means by which a designer describes a design to himself, to
his client and ultimately to a producer. His success as a designer depends on the quality of
the product which is ultimately produced; and his input to the production is through these
documents. They are his only product.

Information in a design is a more aesthetic concept. It concerns the information which
conveys the qualities of "essential form". It consists of visual messages that have been written
in our collective cultural experience. Through them the product gives us information about
itself. It answers questions like: does it look reliable? do all the parts seem to go together? is
this product fun to use? will it symbolize the status I aspire to? A concern for this kind of
information is behind the "product semantics" movement in industrial design. As there are
fewer opportunities to express more tangible, mechanical functions in microelectronic products,
designers are beginning to study the ways in which meaning is designed into products and read
out of them. It is information in this sense which may emerge as the most important issue
for industrial designers in the next decade or so. And it is information in this sense which is
at the center of my interest in design as a way of knowing.

Summary: Implications for a University Core Course in Design.

1. Any core course in design should present a complete and
balanced picture of the design process, both the "design
science" and the "design art" views. Within this spectrum,
students should be exposed to diverse methods of solving
design problems, from analytical to playful.

2. The course should serve to explain and extend the
pervasive concepts of information and information processing
into the very human process of designing.

3. The course should be housed and scheduled in such a way
that the students can experience the design process first hand
and have continued involvement with their work.

4. Students must look at things, many things.

5. Students must generate alternatives, many alternatives.

6. Problems will have to be developed which, like the
cardboard stool, provide opportunities to teach many design
issues but with economies of scale that are appropriate to an
all-university course. Perhaps design problems should all be
doable on the distributed computing system.
10. ON ENGINEERING DESIGN

S. Talukdar and E. Westerberg

BACKGROUND

DESIGN TECHNIQUES HAVE A COMMON CORE

Simon has pointed out that, "everyone designs who devises courses of action aimed at changing existing situations into preferred ones." Engineers are designers by profession who are trained to provide preferred properties to material products like bridges, symbolic products, like computer programs, and abstract products, like algorithms. The training is provided by university departments that concentrate on different disciplines (different areas of applied science like electricity and metallurgy). Design techniques vary across these disciplines. But they have a common core. The intellectual activities involved in ensuring that a bridge will have desired properties are similar to the ones that ensure a microelectronic circuit will produce desired outputs, or an algorithm, desired results. Moreover, engineering design problems tend to have common kernels that are represented in common terms (like equations and graphs) and solved by common technologies (like numerical analysis and mathematical programming). The argument will be continued in a later section.

PRODUCT LIFE CYCLES INCLUDE THE DESIGN OF MANY PROCESSES

The life cycle of a product begins when a need for it is perceived, and ends when its decommissioning is complete. In between, it goes through a variety of processes including production, delivery, use, maintenance and modification. The total design effort for the product includes the design of these processes and usually extends over the entire life cycle, though activity levels tend to be highest during the early parts. As an illustration, consider a typical line of American cars. Design activity is concentrated in the three to five year period preceding its appearance in dealers' showrooms and is aimed at determining not only every physical detail of the car, but also, every detail of the processes by which it is to be built and sold, including its manufacture, marketing, distribution, and maintenance. Later, other processes may have to be designed—recall procedures if defects are detected and recycling procedures to recover scarce materials, for instance. The design of all these processes is no less a part of the overall design of the car than the design of its physical details.

Henceforth, we will use the term life cycle design to refer to the sum total of design activity over a product's life cycle.
LIFE CYCLE DESIGN IS A MULTI-STAGE PROBLEM

There are two reasons why life cycle design must be broken into a number of separate and distinct stages. First, the life cycle design problem is too large and complex to tackle all at once; it must be broken into pieces of manageable size. Second, it is impossible to anticipate all the processes that will be needed over a life cycle. For instance, the architects of the Taj Mahal (a marble tomb built several hundred years ago) could not possibly have foreseen modern air pollutants and therefore, could not possibly have designed to protect against them.

The fact that life cycle design must be treated as a multi-stage problem raises two key questions, both of which are as yet open.

The first question is: how is a life cycle design to be decomposed into stages? (Where are the boundaries between stages to be placed and how are the stages to be ordered?) Some of the decomposition decisions are obvious. For instance, the skills required to design the outer shape of a car are quite different from those needed to design its manufacturing process. Therefore, it makes sense to put these two activities in different stages. Also, since it is easier to design a manufacturing process when one knows what is to be manufactured, it makes sense to put the outer-shape-stage before the manufacturing-stage. In other cases, however, things may not be as clear. For instance, should one design for reliability and maintainability in one stage or two? If two, should they be in series or parallel?

The second key question is: how are decisions made in upstream stages to be kept from thwarting the success of downstream stages? How, for instance, is the stylist of a car to be kept from making decisions that will make the car difficult, perhaps impossible, to manufacture?

EACH DESIGN STAGE GENERATES A NEW LAYER OF INFORMATION THROUGH SEARCH OR COMPOSITION

As the previous sections have made clear, a life cycle design is a collection of individual designs, each for a different aspect of a product, and each generated in a different stage. The aspects include everything that must be designed, particularly, the product’s form and how it is to be made, tested, distributed, used, maintained, modified, and decommissioned.

The input to a stage is a set of specifications on the aspect it is to address. This set can come either from preceding stages or from outside the design process.

The output of each stage is a chunk of information that is expressed in forms such as
blueprints, formulae and flowsheets.

The process by which the output is obtained can be thought of either as a search through a set of alternatives or as a composition. In either case, the same three steps are involved:

1. acquire a concept (in the search paradigm, the concept determines which set of alternatives will be searched; in the composition paradigm it can be thought of as skeleton or outline).

2. generate a tentative design from the concept (find what seems like a good alternative, in the search paradigm; flesh out the concept with detail, in the composition paradigm).

3. evaluate and analyze the resulting chunk of information to see if it satisfies the specifications.

These steps are performed in various nested iterations until the specifications are met, and each may involve activities that range from routine to inventive.

As an example, consider the design of an engine for a car. Suppose that the specifications call for the engine to develop in excess of 650 horsepower, weigh no more than 800 pounds, and last, under racing conditions, for at least 400 miles. To acquire a concept, the designer might borrow from an existing design (say, 8 cylinders in a 120° vee with fuel injection), or he might invent a completely new concept for internal combustion engines. To perform the second step, the designer must transform the concept into a complete set of blueprints or some similarly detailed description of the engine. Again, the intellectual processes involved could range from routine through innovative. The third step is to evaluate the engine. This might be done in the traditional way (build a prototype and test it) or by some new way that the designer invents. If the engine fails, the designer may redo the second step, or if he feels the faults are more basic, the first and second steps.

THERE ARE THREE LEVELS OF INTELLECTUAL ACTIVITY FOR EACH STEP

As the preceding example points out, each of the three steps in a design stage involves intellectual activities that can range from discovery and invention at one end, through routine and algorithmic, at the other. We will arbitrarily divide this range into three levels:

Invention—creative, original thinking of the sort used to invent new things. This level of thinking is little understood and is well beyond our capabilities to automate.

Expert—thinking at the level of the experts in a field. This sort of thinking involves complex pattern recognition and qualitative reasoning, both of which humans acquire through
long, on-the-job experience. Examples include: managing information flows among CAD (Computer Aided Design) tools, managing design projects, synthesizing new systems from existing components, and evaluating designs in terms of non-quantifiable attributes like quality and safety. There are few automatic tools for design at this level but the technologies to create them are rapidly emerging from the field of artificial intelligence.

Algorithmic—detailed, step-by-step progressions of the sort involved in numerical algorithms for simulation and optimization. This level is well populated with tools. In fact, most existing CAD (Computer Aided Design) tools belong in this level.

STYLES

Besides the degree of innovation used in each step, the design processes used for a stage can be distinguished by style. Some examples of styles are:

- Extrapolation—search for an existing design that is close to the one needed; then modify the existing design.

- Top-down—begin with the specifications and transform them into a design.

- Bottom-up—begin with a set of components and attempt to arrange them to meet the specifications.

- Strongly iterative—iterate many times through the generation and evaluation steps in an attempt to compensate for weaknesses in the methods available for generation.

We do not know how close this comes to being a comprehensive list of styles—the subject needs much more study.

KNOWLEDGE SOURCES AND FLOWS

Where is design knowledge to be found? How does it evolve?

Over the last few decades, much of the qualitative parts of design knowledge have been driven out of universities and can now be found only in industry. The rationale for this expulsion has been that engineering schools should concentrate on applied science rather than art; that they should teach rigorous, quantitative material, rather than heuristics. As a result, universities have not played a large role in developing or teaching the qualitative and nonroutine parts of design.

Of course, the boundary between the routine and nonroutine is continually moving. As innovative achievements become better understood, they can often be translated into recipes. A good example is provided by expert systems. Many of these programs, which start out
containing relatively little known and fuzzy reasoning, evolve into algorithmic codes.

New design knowledge usually appears as "evaluation knowledge." That is, one usually discovers how to test for a property in a product before discovering how to generate it. But generative processes are more useful to designers. Unfortunately, there are no generally applicable methods for transforming evaluation knowledge into generation knowledge.

Summarizing, four open questions are: How can one accelerate the production of new design knowledge? How is one to promote flows of expert knowledge from industry to universities? Can one develop systematic methods to transform nonroutine into routine knowledge? How can one speed up the evolution of evaluation knowledge into generation knowledge?

RESEARCH ISSUES AT THE CORE OF DESIGN

The preceding sections have discussed design in general and identified some open questions. This part of the paper focuses on the open questions that we feel most deserve attention and outlines a plan for attacking them.

BETTER AND FASTER DESIGN TECHNOLOGIES ARE BADLY NEEDED

The two most pressing needs of engineering designers are for better and faster techniques for life cycle design.

The need for better designs is easy to understand; improved performance over a product’s entire life cycle is a desire that all the parties associated with the product (its makers, users and society at large) will always have. Of particular concern are techniques for coordinating design stages. At present, upstream stages often thwart downstream stages. For instance, the design of the shape of a car in an upstream stage may make the car difficult to manufacture and maintain.

The need for greater speed becomes clear when one recognizes that it takes years to design a major engineering product—about three for a car, five for a computer and ten for an aircraft. The time constants of the marketplace have become much shorter. As a result, the window in which a product is marketable is often considerably narrower than its design time. For instance, a computer that takes five years to design may be marketable for only a year and a half. Consequently, it is quite easy to miss the window and build a product that becomes obsolete or unwanted before it goes to market. Worse, development and innovation are stifled.
A WIDENING DESIGN GAP

The needs for faster and better design processes are not new. Nevertheless, little relative progress has been made in either area. Design times, at best, have remained constant (as in the case of buildings), and at worst, are growing rapidly (as in the case of very large scale integrated circuits). Nor has overall design efficacy made any relative improvements. For instance, it is no easier today, to design a product to be manufacturable than it ever was. Of course, today's products and manufacturing processes are much more complicated. In fact, the problem is that hardware and manufacturing technologies have been developing faster than design technologies. A good part of the reason is that design research has been fragmented by discipline with no recognized common core. Imagine how much slower computers would have developed if their evolution had followed similar lines.

THE DRC AND THE EDRC

In 1974 the DRC (Design Research Center) was formed at Carnegie Mellon to bring together design experts from a wide range of disciplines to meld them into a cohesive group with a common culture. In May, 1986, the DRC was transformed into a much bigger organization—the EDRC (Engineering Design Research Center)—with an injection of funding from the the National Science Foundation ($14.9 million over five years). The mission of the EDRC is to make significant progress in meeting the needs for faster, better design techniques by concentrating on crucial, core issues of design.

META-DESIGN PROBLEMS HAVE MORE IN COMMON THAN DESIGN PROBLEMS

We now return to the matter of a common core for design and the question: what do design problems from different disciplines share?

Certainly the problem solving infrastructure—computers, operating systems and languages—is shared. Also, there are a few kernel problems that occur repeatedly. Consider two design problems called A and B. Their common kernel, which we will call C, is their intersection. This kernel will almost always contain one of a few mathematical problems that can be represented in terms such as equation, graphs, finite elements and optimizations, and can be solved with techniques from fields such as numerical analysis and operations research.

Many engineers recognize the existence of mathematical kernels but are unconvinced that the commonalities go further. Their arguments are often based on unions of A and B which we will call D. If A and B have very similar domains (for instance, if A is the problem of designing houses for various terrains, and B, for various climates), then it might be advantageous to undertake one project to solve D instead of two separate projects, one for A
and the other for B. However, when A and B are from different disciplines, their unions are seldom of much interest. Consider two manifestations of the floor planning problem. In kitchens, the goal of floor planning is to lay out rectangular appliances to meet the needs of cooks; in microelectronic chips, to lay out rectangular packages of transistors to meet electrical constraints. These two sets of constraints probably have no common solutions. In any case, it is unlikely that one would ever want to lay out a kitchen to behave like a microprocessor. Even if the constraint sets are to be handled separately, they are so different that no single, coherent process could handle both with equal facility.

In other words, domain-specific issues exert so strong an influence on design problems that generalization of the sort described above are of little use when disciplinary boundaries are crossed. Nevertheless, people in the DRC found themselves exchanging ideas of a quite general nature and covering more than just techniques for solving mathematical kernels. We believe the explanation is that design problems have much more in common than mathematical kernels, but these commonalities are obscured by the details of the problems. A useful analogy is provided by two computer programs written in superficially different but substantially similar languages and addressing similar but not identical problems. In all likelihood, the intersection and union of these programs would emphasize their differences and hide their similarities.

This brings us to the matter of meta-design problems which are concerned with methods to construct design methods and tools to build design tools. For instance, how to use CAD tools to generate the blueprints for a bridge, is a design problem. How to design a computer environment in which to build CAD tools for bridges, is a meta-design problem.

We suspect that meta-design problems have much more in common than their lower level, discipline-specific offspring, and great gains in research productivity can be obtained by identifying and exploiting these commonalities.

A RESEARCH PLAN

We have pointed out the importance of meta-problems. Unfortunately, not enough is known about these problems to tackle them directly in a top-down manner. Instead, an indirect bottom-up approach must be used. The specific approach we have selected is:

• Distribute two or more projects from each of the critical areas (listed in the preceding section) in different disciplines. The intent is to have the projects proceed concurrently and to promote cross-fertilizations among them. From these cross-fertilizations we will hope to get a better understanding of the corresponding meta-problems and its solution.

• Designate clients for each project from disciplines other than the one in which it is being done. The purpose is to have the clients interject "users' views" as the project
proceeds, and to keep the discipline-specific and general developments from becoming inextricably mixed.

An added benefit of this approach is that real problems that are important to industry will be tackled in each project. Thus, the projects will provide strong, two-way communication channels to industry.

It is interesting to note that projects and clients from each area could be placed in any of our engineering departments. This is one more piece of evidence for a common core of design. Of course, the decisive evidence will come from the success or failure of the projects to precisely define and solve important meta-problems. This evidence should be in by 1991.

SOME CRITICAL AREAS

We now turn to the specifics of where research is most needed.

Over the years, large numbers of computerized tools have been developed to aid the engineering designer. As previous sections have pointed out, most of these tools are restricted to the lower, detailed levels of design activity and predominantly algorithmic in nature. Moreover, most of them are for evaluation and analysis. The other aspects of design are left uncovered. In particular, there is a great scarcity and need for tools for the steps of concept acquisition and design generation. Methods are badly needed for coordinating multiple design stages. The greatest opportunities are in formalizing intelligent and qualitative approaches. While these approaches play a very large role in design processes, they remain to be studies, catalogued, described and automated.

The the remainder of this section we will list five areas that are critical to better, faster design, and have high potential for progress.

Understanding the higher, conceptual levels of design activity, especially those involved in innovation, discovery and invention. If we are ever to increase the innovative content of designs, we must first understand what intellectual processes are involved. Is a "garbage can" of vast amounts of random, unconnected information necessary? Can innovation be automated?

Generative Techniques. The bulk of the tools available to the designer are for evaluation, analysis and graphic display. In principle, optimization and heuristic search techniques can be used for the generative step of a design. But in practice, applications have been few, largely because of implemental difficulties. Much more of the generative step must be automated, if we are to design better and faster.
Lifecycle Design. There are no good methods for coordinating stages and, in particular, for keeping upstream stages from making choices that render it difficult, or impossible, for downstream stages to achieve their goals.

Design Environments. The environment in which a designer works includes the tools and computers he may use. In existing environments, the connections between tools invariably require human intervention. The effort expended in managing these information flows can exceed that for any other part of the project. In addition, humans must handle all the supervisory chores, such as selecting sub-goals, choosing the most appropriate tool for each subgoal, and overseeing the tool's activities. If design speeds are to increase, much of this information management and tool supervision will have to be automated.

Rapid Prototyping. Complex products, like the composite lens of an automobile headlamp, often have to be built before they can be evaluated. This makes the design time a multiple of the time it takes to build a prototype, which can be months or even years. The "rapid prototyping" idea is to automate every step in the production of prototypes for a class of products, so that one can go from specifications to a prototype in just a few days.

EDUCATION

How should engineering design be taught? The key is the creation of a rich design environment, in which students and faculty can work on challenging problems. Some elaborations on this thought and other elements of a program are given below.

Intellectual Challenges. Many design skills are best learned through "doing." But it is difficult to work up a great deal of enthusiasm for a problem whose only purpose is to determine a grade in a course. In contrast, "real problems" can capture a student's imagination, cause him/her to make heroic efforts, come up with innovative solutions and have a very beneficial educational effect. The software field is full of examples. EMACS, a full screen editor widely used at CMU and elsewhere is the result of student effort at MIT. Putting EMACS into a UNIX environment was done by students here at CMU. SCRIBE, the most widely used word processing package at CMU, was from a student effort here. We need similar activities in other design disciplines. The best way is to arrange for students to know what the needs, challenges and developments are in different fields. Design problems from industry and interactions with faculty, staff and industrial visitors interested in design research must be promoted. Real challenges will promote innovative skills.

State-of-the-art Tools. Design work is tool-intensive; in order for students to make designs and evaluate them, they need a considerable array of tools. It is impractical to
develop these tools from scratch. The natural source for them is the research community. The first step in the transition is to put a research tool into an educational network. Next it should, if necessary, be changed to make it easier to use; it should be made more robust, and given better explanation facilities and error tracing capabilities. Usually these capabilities must be paid for by reducing the tools capabilities for research. However, one can also expect some beneficial feedback to the research version.

Feedback. It is important for students to get feedback on their designs, to be able to evaluate and test them, to see how they would actually work. The importance of this feedback has been substantiated in VLSI design courses, where students design a microelectronic circuit on a chip, get it fabricated and then test it to see if it really does what they thought it would. These courses have been enormous successes. However, it is not always convenient to build the product that has been designed. Simulation and evaluation tools are one way to do the testing in the abstract.

Ph.D. Theses. Engineering colleges usually want Ph.D. research to be analytical. Designs, no matter how ingenious and attractive they may be, are usually unacceptable unless they are accompanied by significant amounts of analysis, theorems, insights or empirical data. This policy discourages creative designers in favor of analysts. Engineering colleges should follow the lead of computer science departments in accepting original and good designs as being sufficient in themselves for doctorate dissertations.
NOTES

134 The Sciences of the Artificial, by H.A. Simon
11. HIERARCHICAL DESIGN STRATEGIES IN 3-DIMENSIONAL CHEMISTRY

Jonathan S. Lindsey

The creation of artificial chemical systems represents one of the most challenging areas of chemistry. Past and current approaches, however, seem insufficient to prepare systems having the desired level of sophistication and 3-dimensional order. The hierarchical nature of biological organization provides some paradigmatic design strategies for formation of chemical systems. These strategies in conjunction with the advent of automated synthesis may provide one attractive solution for the design of complex chemical systems.

The design and creation of complex chemical systems represents one of the most fertile areas in the science of chemistry. Two forces appear to be slowly evolving what may be referred to as the era of 3-dimensional chemistry. On one hand is the relentless march of biochemistry and molecular biology, which are elucidating in ever-increasing detail and clarity the molecular structural basis of living systems. On the other hand is the 20th-century realization of the alchemists' dream—the synthesis of raw substances into complex artificial materials of enormous variety, some literally worth their weight in gold many times over. Despite the value of the latter, even the most sophisticated artificial materials are primitive in comparison with the simplest living systems. It therefore seems germane to examine in a very general manner some of the gross organizational features in biological systems, compare these with current dogma and methodologies in the design of chemical systems, and consider in a somewhat provocative fashion the feasibility of their eclectic fusion to create a more powerful science of design in the chemical arena.

Biological systems are distinguished at the molecular level from artificial chemical systems by an extraordinarily high degree of 3-dimensional order (Figure 1). Though individual molecules in artificial systems may be highly structured, it is the long-range order and cooperative interactions of ensembles of molecules which collectively give rise to phenomena characteristic of biological systems. Included among these phenomena are molecular recognition, chemical catalysis, energy conversion, contractility, sensory evaluation, homeostasis, and self-replication. The long-range order found in biological systems does not arise from a simple repeating pattern of one type of molecule, but instead consists of a richly structured milieu of many different types of molecules. This is in contrast with crystalline materials, which do consist of 3-dimensional repeating patterns of one or at most several types of molecules. Because of the simplicity of the repeating pattern however, crystals do not in general exhibit phenomena characteristic of biological systems (Figure 1).
The conversion of light into chemical energy in the photosynthetic systems exemplifies the dramatic functional processes which may accrue from structurally organized ensembles of molecules (Clayton, 1978). The photosynthetic apparatus is a sub-micron device comprised in part by hundreds of antenna chlorophyll molecules (each 15 Å in diameter) distributed over a span of hundreds of Angstroms. The antenna chlorophyll function to absorb light and funnel the photon energy to the reaction centers, where the electron transfer cascade originates leading ultimately to the storage of chemical energy. The energy transfer process in the light harvesting apparatus proceeds via stepwise migration of the energy from one chlorophyll to another until the reaction center is reached. The overall migration resembles a random walk among the entire pool of chlorophyll molecules, and is complete in several hundred picoseconds. The distances of separation and mutual orientations of the chlorophyll molecules are critical determinants of the efficiency of energy transfer. The 3-dimensional arrangement of molecules is rigorously controlled by a structural latticework of protein molecules, providing in turn for the fine tuning of the transfer process. The energy transfer process represents only the initial phase of photosynthesis, however the essential point is that hundreds of molecules participate in a cooperative manner to convert light into chemical energy with remarkable efficiency.

The Challenge:

The challenge to chemistry is to create artificial systems having biological complexity and reap the benefits of the collective interactions of structured aggregates of molecules. The goal extends beyond the mimicry of biological phenomena to encompass materials science and the creation of artificial systems with features not represented in the animate world. One example is the emerging field of molecular electronics, widely suggested to represent the ultimate stage in the miniaturization of computer circuitry. The challenge is to design and create molecular switches and molecular conductors with which molecular electronic devices (MED's) can be fabricated. One great attraction, for example, is to build circuitry in three dimensions rather than only in two as currently practiced. The expected benefit is computations and memory densities faster and higher by orders of magnitude than current technologies.

The successful development of a science of molecular electronics requires dramatic advances in the triad of theoretical design, chemical synthesis, and characterization of the complex assemblies of molecules which collectively and coherently function as a molecular electronic device. Current prowess in molecular electronics is quite primitive and the eventual form that MED's will take is open to conjecture (Haddon, 1985). Regardless of their eventual form, the functional requirements of MED's dictate that the structural organization of the ensemble of molecules comprising a MED be comparable to the high 3-dimensional order of
biological systems.

The Problem:

The three legs of the triad: design, synthesis, and characterization, form the common themes in the creation of functional artificial systems. We shall center discussion around the problems of the design and synthesis of complex chemical systems, and focus in particular on the question 'How can we most efficiently create 3-dimensional chemical systems having virtual biological complexity?'. The magnitude of this problem is thrown into sharp relief by contrasting the strategies in synthetic organic chemistry with some of the most general themes found in biological anabolism.

The synthesis of organic compounds involves the conversion of simple feedstocks via chemical reactions into more complex target molecules. The types of target molecules which occupy the interests of most synthetic chemists can be arbitrarily divided into two categories, those which are polymeric and those which are not. Polymeric molecules are long chain molecules composed of repeating monomeric units, exemplified by polystyrene. Polymers typically have high molecular weights (into the million atomic mass range) and are formed in one step from the corresponding monomer. The product obtained in one-step polymerization reactions is not unique in mass or length but consists of a distribution of many molecules closely related in size. Each individual polymer molecule can usually exist in a variety of conformational states, hence the structural characterization of the product at best represents an average over many different states.

The non-polymeric molecules occur in immense variety and nominally do not consist of repeating units. Though the size of these molecules is not large (usually less than 1000 atomic mass units), their diversity arises from the detailed structural arrangements of the various atoms which comprise the molecule. The synthesis of each different compound is performed through a sequence of chemical reactions which may number as many as 15 steps. At each step in the synthetic pathway the intermediate product can be purified, characterized, and its molecular structure determined in detail. The final product obtained after purification is unique rather than consisting of a distribution of closely related species (Chart 1).

To create complex chemical systems with a high degree of 3-dimensional order, both approaches have merits though neither alone is sufficient. The approaches of polymer chemistry readily convert small monomers into high molecular weight products, but without the desired rigorous control of structure. Traditional organic chemistry affords detailed control of structure, but as yet most efforts have been restricted to molecules which are small in comparison to the long-range order found in biological systems. New approaches are needed
which blend the best of current methodologies with hierarchical strategies for creating complex systems.

Biology as a Paradigm:

Biological systems differ radically in their gross organization and biosynthetic construction from the previously described chemical systems. Three themes: hierarchical organization, modular synthesis, and self-assembly most markedly distinguish biological systems. At the risk of over-simplification, we can briefly explore these themes before examining their implications for the creation of chemical systems.

Examination of the organization of biological system from either a size or functional viewpoint results in the same hierarchic decomposition from animals to atoms.

Atoms
Small Molecules
Large Molecules
Molecular Assemblies
Sub-cellular Organelles
Cells
Tissues
Organs
Animals

Complexity evolves from simplicity most readily when the system is hierarchically structured, and nowhere is this more apparent than in the organization of biological systems. The formation and structural transformations of small molecules are fairly well understood phenomena in both biological and chemical systems, and though the detailed mechanisms may differ, the transformations are often quite similar. It is the ascension to the next two hierarchical levels, the formation of large molecules and molecular assemblies, where biological and current chemical systems differ. Because these levels represent the level of complexity and structural order we seek to obtain in chemical systems, we now turn our attention to the hierarchical design strategies employed so successfully in biological systems.

Hierarchical systems are usually composed of a few different kinds of subsystems in various combinations and arrangements. This high degree of redundancy allows the complex system to be generated from a basis set of elementary subsystems (Simon, 1981). The major
classes of molecules found in biological systems, proteins, nucleic acids, polysaccharides, and fatty acids, are composed of repeating sequences of modular subunits. Though each class represents a type of polymer, the synthesis of the chain molecule in each case is performed one step at a time and the sequential arrangement of components along the linear chain is rigorously controlled. This synthetic process is contrasted with polymer chemistry, where polymerization occurs randomly.

The sequence-specific stepwise synthesis of biomolecules is most clearly illustrated in the biosynthesis of proteins. Twenty amino acids form the basis set from which the proteins are generated. Alterations in protein structure and function are achieved by variations in amino acid sequence. The number of different proteins which may be formed is essentially limitless due to the sequence combinations which twenty amino acids provide. There are for example some 3 million combinations of twenty amino acids which can be formed in a five amino acid sequence. Moreover, a five amino acid sequence is hardly large enough to be considered a protein. Myoglobin, an oxygen binding protein present in mammalian muscle, consists of about 140 amino acids and is representative of moderately sized proteins. The synthesis of new proteins by variations in amino acid sequence is most cogently illustrated by the immune system, which through variations in amino acid sequence produces antibodies specifically directed against an enormous range of chemically different antigenic determinants (Edelman, 1969).

Modularity is thus the central theme in the biosynthesis of large molecules. The use of modular subunits minimizes the information needed to create complex molecules. New biochemical machinery and new chemical reactions are not required to produce new structures. This contrasts sharply with traditional organic chemistry, where new sequences of chemical reactions and sometimes the development of entirely new reactions are required to synthesize new compounds.

The synthesis of short segments of DNA and of proteins can be performed chemically without the aid of biological systems, so the concept of modularity is not at all restricted to biosystems. In fact automated synthesis machines have been constructed and are commercially available for the specific syntheses of DNA and proteins (Merrifield, 1966). (We shall return later to the issue of automation.) However, the concept of modularity and stepwise synthesis as a route to complex molecules has not been applied to molecules other than those which are naturally formed (nucleic acids, proteins, polysaccharides, and fatty acids).

Although these concepts have not been thoroughly exploited, the ascension of hierarchical
levels from modular subunits to large molecules via stepwise synthesis is reasonably well understood. The further ascension to molecular assemblies from large molecules involves the more complex phenomena of self-organization known as self-assembly. Because self-assembly occurs widely in diverse formats and involves complex molecules, a complete description of the chemical determinants of self-assembly processes is not yet available. Nonetheless, some general statements concerning common themes found in disparate self-assembly processes can still be made (Kushner, 1969; Bouck, 1976).

Self-assembly can be most broadly defined as the spontaneous yet selective formation of particular 3-dimensional ensembles of molecules from simpler precursor molecules. In biological self-assembly large structures are formed through repetitive use of smaller molecules, decreasing the amount of genetic information required to specify the molecular ensemble. Furthermore, synthesis errors are reduced since malformed components tend to be excluded during the assembly process. The 'robustness' of self-assembly accrues from the feedback control mechanism that reversibility of bond formation provides. In most cases in biology the complexes formed via self-assembly are held together via weak non-covalent interactions and thus fall under the rubric of supramolecular chemistry. Because association occurs through multiple bonds of relatively low energy, assembly and disassembly occur under gentle conditions and are easily controlled. Self-assembly processes are distinguished by their near-exclusive formation of a single complex product in high yield.

The quintessential example of a self-assembly process is the formation of double helical DNA. Two single strands of DNA complementary in sequence contact intermolecularly by diffusion. Non-covalent base pairing occurs, and upon recognition and proper digitation of complementary bases, the two single strands rapidly zipper together to form the double helical DNA molecule. The unifying theme of self-assembly processes at the microscopic level is that one or more intermolecular interactions gives rise to a succession of intramolecular interactions, each contributing to the overall stability of the molecular complex. The formation of precise 3-dimensional structures occurs because only those molecules having the correct positioning of functional groups and the proper structure can form the maximum number of binding interactions.

Other clear-cut examples of self-assembly as defined here include conversion of globular actin to fibrous actin, formation of microtubules and flagellin, assembly of protein coats of viruses such as tobacco mosaic virus and bacteriophage lambda, and certain aspects of protein folding. It is almost certain that full-fledged self-assembly processes in vivo occur subject to a considerable number of biological regulatory mechanisms. However, some self-assembly
processes can be performed at least in part in vitro, and the products formed accounted for by relatively straightforward thermodynamic models. Thus the phenomena of self-assembly should be exploitable as a means of creating complex ensembles of molecules.

Design of Chemical Systems:

The design and creation of artificial chemical systems is divided into several stages. Goal-setting, such as the acquisition of knowledge or the development of a product, is first performed based on a needs-analysis. In particular a set of target molecules are generated which are believed will provide a means to achieve the desired goal. Much of the work in synthetic organic chemistry involves synthesis of small molecules which occur naturally (natural products chemistry), hence the structures in these cases are given. For artificial systems the selection of the target molecule is of critical importance and necessitates extensive planning.

The next stage of design is process synthesis: developing a reaction path for each specific target molecule (Rudd, 1973). A reaction path is the prescription for the series of chemical reactions which convert feedstocks and commercially available precursors to the target molecule. The construction of the reaction path involves a complex search among the available arsenal of reactions, typically numbering in the thousands. Computer programs have been developed which can analyze some synthesis problems retrosynthetically, working back from the target to the wide assortment of available precursors (Corey, 1969). Many alternative routes are available for even the smallest organic molecules, and a large synthesis tree is thus constructed. Most chemists design synthesis trees without the aid of computers, using both retrosynthetic analysis and forward searching strategies based on an implicit knowledge of available precursors, the feasibility of many synthetic reactions, and the properties of the intermediate compounds.

Planning of a reaction path is not performed in a vacuum, but is interactively analyzed for feasibility and attractiveness. Analysis of reaction paths involves heuristics at two levels: a global or primary set of heuristics concerning the strategic selection and ordering of the reactions to achieve the most efficient formation of product, and a secondary set of heuristics for evaluating the judiciousness of any particular reaction step. This 'generate and test' strategy allows the number of alternative reaction paths to be greatly narrowed (Govind, 1977).

The designer of the artificial system is interested in achieving the final goal and is not wedded to any particular target molecule. A means–ends analysis can be performed, re-evaluating the target molecule(s) which will satisfy the desired goal in light of alternative synthesis plans. In this manner the designer has the flexibility to tailor the target molecule within the context of the goal, and yet profit from especially powerful synthetic design strategies. Powerful design strategies are those which convert simple precursors to complex
molecules in a small number of steps with a minimum level of difficulty.

Consideration of the hierarchical organization of biological systems leads to a set of heuristics for formation of complex molecules from simple precursors:

1. Use a small library of modular components.
2. Couple the modular components in a stepwise manner.
3. Employ a limited set of coupling reactions.
4. Maintain control of molecular structure at all times.
5. Use self-assembly when possible.

A logic based program for synthesis design recently described by Hendrickson (1985) utilizes symmetry and topological considerations in preparing a synthetic approach. Some of the heuristics employed are:

a) Economy is achieved by using the fewest steps or operations in the most convergent order.

b) Performing several constructions at once is a very powerful synthesis concept which provides a much greater efficiency.

c) Large starting materials are especially favored.

Though statement b) has rarely been exploited in synthetic chemistry, it is identical to the concept of self-assembly (heuristic 5) derived from examination of the hierarchical organization of biological systems. Emphasis on convergent sequential strategies results from considerations of reaction yields. A linear sequence of a dozen reactions, each of low to moderate (less than 50%) yield, necessitates the use of unwieldy amounts of starting materials to achieve acceptable levels of product. A chain is no stronger than its weakest link, and one low yield step toward the end of a sequence of reactions severely crimps the overall process. Using convergent pathways and arranging to perform the lowest yield reactions first can result in a more efficient synthetic process.

Although we recognize the importance of convergence, the considerations in opposition to serial syntheses may not be strictly applicable to the stepwise synthetic coupling of modular components. In a modular approach the small set of coupling reactions employed can be chosen for high yield from among the thousands of available reactions, and then further optimized and fine-tuned for the application at hand. The up-front investment in yield
optimization generates dividends each time the reaction is repeated. Rarely can the chemist afford to thoroughly optimize a particular step in a one-shot serial reaction path.

The comparison of potential reaction paths requires a detailed understanding of each reaction step, and synthetic chemists implicitly evaluate many parameters in addition to reaction yield. These include factors such as: a) duration of reaction, b) hazardous conditions, c) cost of reagents, catalysts, and solvents, d) exotic reaction conditions such as extremes of temperature or pressure, e) necessary equipment, f) conditions for product workup, g) possible purification problems, and h) total man–hours required. Some estimate is generally made concerning the tolerable margin of error (flexibility) and the track record (reliability) of a chosen reaction. Knowledge of these factors in conjunction with expected yield data allows the chemist to assess the feasibility and efficiency of the reaction step.

Technology which now appears on the horizon suggests that "possibility of automation" is a parameter that may soon enter the evaluation function. Pursuing this line of thought, it is clear that reaction steps and reaction paths which can be automated are much more attractive than those performed manually. Given that the designer of artificial chemical systems has great flexibility, target molecules can be tailored for synthesis via automated technology. The heuristic of "Incorporate the capacity for automation when possible" then becomes a powerful design strategy in the creation of complex chemical systems.

Automated synthesis in chemistry is essential for two different reasons. Theories of molecular design are not sufficiently advanced to permit the a priori design of target molecules for the performance of a particular chemical function. A variety of molecules must typically be examined by trial and error until the design requirements are satisfied. Automated synthesis reduces the human effort involved in the preparation of the trial compounds, and the construction of the target compounds from modular components minimizes the new chemistry which must be developed during this exploration and selection process. A second attraction of automation occurs at a later stage when a target molecule with desirable features has been identified. The pre-established automated synthesis can then be readily exploited for its production.

We are developing an automated synthetic work station in conjunction with chemical engineers and computer scientists. The major task of this work station is to carry out a limited set of synthetic organic reactions with minimal human supervision. A prototype currently under construction consists of a reaction vessel, solvent dispensing system, an array of sensors including on-line analytical instruments, and limited capabilities for product purification.
A small robot at the heart of the work station is used to manipulate solvents and synthetic reagents in the preparation and analysis of the compounds.

To achieve the goal of automated operation, at least a skeleton of an expert system must be developed. Though expert systems cannot yet be written to encompass the whole of synthetic chemistry, a limited expert system can definitely be constructed for the type of modular chemistry envisioned. The physical and chemical properties of a small family of components can be tabulated, and detailed models of the synthetic reactions can be included in the database. Furthermore, at least some of the chemical properties of the target molecules can be predicted on the basis of the properties of the modular components.

Some Examples:

A powerful approach to the design of a complex structure is to "discover viable ways of decomposing it into independent or semi-independent components corresponding to its many functional parts. The design of each component can then be carried out with some degree of independence of the design of others, since each will affect the others largely through its function and independently of the details of the mechanisms that accomplish the function." (Simon, 1981). In this spirit, we believe complex target molecules must be designed with facility for hierarchical decomposition. Specifically, a synthetic plan which utilizes modular components, self-assembly, and automation is likely to offer superior features of economy and overall attractiveness. The only example to point to which has exploited the trilogy of automated synthesis of modular components followed by self-assembly is the chemosynthetic recapitulation of the biosynthesis of proteins and DNA. In general, automation of synthesis has hardly entered the chemical laboratory, and modular approaches only rarely have been demonstrated. Though the most profound examples of self-assembly occur in biological systems, some rather simple cases are known in chemistry.

Chelation of a metal atom by a polydentate ligand is probably the most representative example of self-assembly in synthetic chemistry. The metal has several (often 4-6) ligation sites, as does the complementary organic ligand. The first contact of ligand with metal is intermolecular, the remaining sites are held close together, and a succession of intramolecular contacts readily occur. The metal chelate is rapidly assembled by the ligand wrapping around the metal. The final structure consists of the metal encircled and in some cases encapsulated by the organic ligand (Figure 2).

The same principle underlying metal chelation was used to design the synthesis of a macropolycyclic porphyrin-quinone cage molecule (Figure 2). The quinone has four aldehyde groups and the porphyrin has four complementary amino groups. One aldehyde and one amine
react intermolecularly to form a Schiff base, in turn bringing the three aldehyde and three amino groups in close proximity. These react readily to give the cage molecule in over 85% yield at thermodynamic equilibrium (Lindsey, 1982). Both metal chelation and formation of the porphyrin-quinone cage molecule are cooperative processes since the first interaction facilitates the subsequent ones. These cooperative processes can be described by simple thermodynamic models involving products of association constants for the initial 'nucleation' step and the subsequent 'growth' steps in the self-assembly process.

Self-assembly also can be achieved without cooperativity, as shown in the synthesis of porphyrins (Lindsey, 1986). In the synthesis of tetraphenylporphyrin, four molecules of benzaldehyde and four of pyrrole self-assemble to form the tetraphenylporphyrinogen in 60% yield at thermodynamic equilibrium (Figure 3). The remaining 40% consists of a broad spectrum of acyclic polymeric components also formed in the reaction. The desired porphyrin is obtained by allowing equilibrium to be attained, followed by oxidative quenching to convert the porphyrinogen to the porphyrin.

In spite of these successes of exploiting self-assembly as a rapid and simple means of synthesizing complex molecules, our knowledge of the chemical determinants of self-assembly processes remains distressingly primitive. We know very little, for example, about how 3-dimensional shape influences molecular assembly processes. Related questions we would like to answer are "How much conformational flexibility can be tolerated without substantially altering selectivity in assembly? To what extent can our knowledge of molecular recognition among non-covalent species (host-guest, receptor-ligand, etc.) be used to design covalent self-assembly processes? Can computational tools be developed to calculate condensed phase equilibrium formation constants of complex multifunctional products when cyclization, polymerization, and macropolycyclization are all possible?" The existence of good computational tools might go a long way toward rendering the design process more systematic and less reliant on sheer intuition. At present our only explicit tool for assessing the possibility of self-assembly is model building: constructing space-filling molecular models to determine by visual inspection if the desired molecules fit together in a complementary fashion. By current standards such structures have large molecular dimensions and high numbers of functional groups. Self-assembly has remained a largely unexplored frontier because in general the syntheses of the necessary precursors stretches the present synthetic methodologies.

Of greatest need is a small family of molecules which can be used as a vehicle to systematically probe the self-assembly question. Toward this end we are currently working to design a variety of potential self-assembling target molecules composed of modular components.
Adherence to the heuristic of modular chemistry should permit both rapid construction and systematic variation of the target molecules as well as automated synthesis.

One potential target molecule has a structure resembling that of a ladder. The ladder molecules simultaneously satisfy our design requirements for studying the self-assembly process in complex 3-dimensional structures, in addition to the capacity for hierarchical decomposition. Furthermore, applications of this type of structure appear to be numerous if the nature and juxtaposition of the molecules in the rung positions can be varied. These include but are not restricted to studies of photochemical energy transfer, electron delocalization, and ion conductivity.

A hierarchical synthetic plan for ladder molecules conceptually mirrors the chemical synthesis of DNA, though the molecules and reaction steps employed are different. The legs of the ladder are synthesized in a stepwise manner from modular subunits. Each subunit has two coupling sites for formation of the ladder leg, and one tertiary functional group for formation of the rung of the ladder. Two different ladder legs are prepared containing complementary functional groups (i.e., amino and aldehyde groups together give a Schiff's base adduct). The final step is then a self-assembly process, where the complementary legs of the ladder interdigitate, yielding the desired target molecule.

To prepare the linear leg of the ladder, the concept of geometric progression (monomer - dimer - tetramer) is a more powerful approach than the purely sequential (monomer - dimer - trimer - tetramer) process (Figure 4). The stepwise synthesis of linear molecules in conjunction with the principle of geometric progression results in a highly convergent approach. Geometric progression also provides an extraordinarily powerful means of ascending into the high molecular weight range with only a few reactions.

In summary, the problem of designing artificial chemical systems presents a paramount challenge in synthetic chemistry. A significant assault can be mounted by merging the best of the incredible wealth of current synthetic methodologies with biologically derived hierarchical design strategies. Modular self-assembly approaches are parsimonious in reaction steps, parlaying a minimum number of components into rather complex structures. In concert with automation technology, a powerful new approach can be taken for the creation of artificial systems at the molecular level.
Acknowledgement:

I thank Gary Powers for illuminating discussions.
Bibliography:


Figure 1. General relation between functional complexity and 3-dimensional order. Biological systems are the epitome of complex functions arising from highly ordered systems. Artificial systems hardly begin to approach biological systems in their complexity. Note the absence of functionally complex systems which are randomly organized.

Figure 2. The self-assembly of a metal chelate compound (left) and a porphyrin-quinone cage molecule (right) both involve cooperative processes. The first intermolecular interaction facilitates the subsequent intramolecular interactions, and all contribute to selective formation of the cyclic product in high yield.

Figure 3. Benzaldehyde and pyrrole self-assemble to form tetraphenylporphyrinogen 1 under equilibrium conditions. The addition of an oxidant (Q) affords the tetraphenylporphyrin 2.

Figure 4. Geometric progression represents an efficient strategy for preparing high molecular weight (MW) structures without the randomness that polymerization entails. If the monomer has a molecular weight of 300, after only 6 cycles the product has a MW of 19,200. If a purely arithmetic sequential process was employed, some 63 cycles would be required to obtain the same product.
<table>
<thead>
<tr>
<th></th>
<th>Synthetic Chemistry</th>
<th>Polymer Chemistry</th>
<th>Biology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>typical MW</strong></td>
<td>≤10³</td>
<td>10⁴ to &gt; 10⁶</td>
<td>10³ - 10⁵</td>
</tr>
<tr>
<td><strong>MW distribution</strong></td>
<td>unique</td>
<td>polydisperse</td>
<td>unique</td>
</tr>
<tr>
<td><strong>structure</strong></td>
<td>well-defined</td>
<td>many conformations</td>
<td>well-defined</td>
</tr>
<tr>
<td><strong>synthesis mode</strong></td>
<td>sequential reactions</td>
<td>monomers form polymer</td>
<td>monomers form polymer via sequential reactions</td>
</tr>
<tr>
<td><strong>self-assembly</strong></td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong># compounds in precursor library</strong></td>
<td>&gt;20,000</td>
<td>several hundred</td>
<td>20 amino acids</td>
</tr>
<tr>
<td><strong># of steps</strong></td>
<td>usually 5 - 15</td>
<td>1</td>
<td>xn in an n-mer</td>
</tr>
<tr>
<td><strong># of repeating units</strong></td>
<td>none</td>
<td>n in an n-mer</td>
<td>n in an n-mer</td>
</tr>
<tr>
<td><strong>average dimensions</strong></td>
<td>≤20 A</td>
<td>100 - 10,000 A or more</td>
<td>50 - 1000 A or more</td>
</tr>
</tbody>
</table>
12. DESIGN AND IMPLEMENTATION OF COMPUTER SYSTEMS

Preface

These are the thoughts of a practicing computer system designer about the process of designing and implementing computer systems. Introspecting about one's rather specialized trade is difficult and presenting the conclusions to a lay audience is even more so. However, there are enough parallels among the various design disciplines, including computer system design, so at least other designers can easily understand the generic synthetic and analytic activities. The problem lies in conveying the computer-specific knowledge that is so vivid to the practitioner? These details comprise the bulk of the skills that are directly taught.

Introduction

Computer systems solve problems in a large domain—sometimes completely, sometimes only partially. Problems may be simple such as controlling a light based on the time of day, or calculating an amortization schedule for a mortgage. Problems may also be complex such as transcribing human speech or providing a central repository (database) of the scheduling and reservation data of an international airline. For problems that are partially solved, humans complete the solution. When a spelling checker locates a word it is does not recognize, it asks a human whether or not the word is acceptable. The spelling checker knows that its own vocabulary is limited.

Surprisingly, the computer (or hardware) that solves this wide range of problems is reasonably simple. It comprises a memory, one or more computational units, and some input/output devices.

A computer's memory is made up of a collection of cells that are referenced by number. Each cell stores data, which on many computers are represented by 32 bits of information. Thirty-two bits are enough to represent 4 characters, a wide range of integers, and many other values. Ideally, cells do not change unless new data is stored into them. Reading the value of a cell leaves the data in it unchanged. In practice, memories are divided into a few constituent parts, which vary principally in the number of cells they contain, the speed at which they operate, and the conditions under which they reliably retain data.

Computers that perform very simple calculations suffice for all applications, provided that the operations are performed very fast. That is to say, a computer with a large enough memory and the ability to carry out a few simple operations quickly (e.g., adding two numbers, or repeating a sequence of operations) is a satisfactory basis for all the applications that can
use computers. In practice, computational units perform slightly more complicated calculations to make up for their limited processing speed.

Input/output devices permit computational units to read from and write data to the real world. Video terminals, thermocouples, and robotic arms are typical input/output devices. Another example is a clock, which permits the computer to know the time.

The process of constructing a computer system to solve a problem requires precisely identifying the problem and then finding a sequence of computer operations that will solve it. That sequence of instructions is called a program, or software. In many ways, writing a program is a typical engineering problem; a program is a synthetic entity, created out of simple components. However, most synthetic activities create material from material. A computer program is a logical entity that describes a sequence of operations that solve a problem.

I believe the primary difficulties in identifying problems and finding their computer solutions arise from two topics I have previously discussed. The first is the great diversity of problems that are to be solved. Frequently, there is no clear conception of what problem the computer should solve or how to solve it. The second is the simplicity of the computer. How can a computer system transcribe speech if it only knows how to carry out a few very simple operations?

Sections 2 through 4 should help to answer these questions. They discuss topics that are fundamental to producing useful computer systems: extending the computer to make it more useful, choosing the right problem to solve, and creating a computer system that solves it. Section 5 describes what is meant by the design and implementation of computer systems.

The Section titled Practical View takes a more directed view of the design and implementation of computer systems and discusses some of the computer-specific knowledge that is needed. The design and implementation of a complex computer system is then exemplified via a case study of the space shuttle's Primary Avionics Software System. The last section contrasts computer system design and implementation with other design disciplines, particularly bridge design and construction, and summarizes the basic topics that pertain to computer system design.

Dealing with Simple Machines

A novice might be surprised to learn that computers are simple and that their simplicity contributes to the difficulty of building computer systems. However, if computers had a wider repertoire of operations, it is logical to believe that it would be easier to solve problems with
them. For example, if computers had a language dictionary and understood the rules by which we form new words by adding prefixes and suffixes to old ones, there would be an easier place to start building spelling checkers. It would take longer to learn what the computer could do, but the reward would be the knowledge of a more useful machine.

This notion of an augmented computer is a good one. It is even better, because it does not require additional silicon chips or copper wire to construct it. Software can extend the basic functions of the computer. Such software effectively provides a virtual computer, which is implemented partially in electronic circuits and partially in software. For example, there are no computer operations for computing trigonometric functions. But they have simple mathematical formulae that use only arithmetic functions, like addition and multiplication. Software that computes the trigonometric functions augments the bare machine and extends its capabilities.

More complex problems can then be solved using the facilities of this software and the underlying basic operation repertoire. The solutions to these more complex problems can be viewed as forming yet another level of virtual computer and so on. The program that is solving the ultimate problem is thus resting on top of a hierarchy of software levels, all of which ultimately rely only on the bare machine.

[To be added]

Figure 12-1: A Hierarchy of Software Levels

Perhaps, a computer could be built that would have enough lower levels of software to make all computer applications expressible as a simple combination of complex operations. If a computer could quickly transcribe text, translate text from one natural language to another, and read text aloud, then a simultaneous translator could be easily built by merely composing the three facilities. The problem is the diversity of computer uses; there are so many uses of computer systems that it would be impossible to determine a universal set of primitive operations for directly expressing all problems.

This is not to say there cannot be some universal augmentation of a computer. In fact, computer scientists have been quite successful in developing widely useful libraries of software to extend its function. For example, computers arrive from the manufacturer with an operating system, a program that increases the functions of the basic machine. Additional software that performs sorting, optimization, text manipulation, and many other functions can also be obtained for nearly all computers.
In addition to software that augments the machine, there are also computer programs that translate (or compile) easier to use programming languages, such as Fortran, Lisp, or Pascal into computer operations. Programming languages also augment the computer just as do operating systems and software libraries. In fact, programming languages do more than just translate; they also automatically utilize software libraries to perform some commonly used functions. For example, nearly all programming languages contain a print operation. The print operation translates into some basic computer operations, but it also uses some low-level software to format numbers. The combination of programming languages and general purpose software make computers much more useful, and together implement what I termed virtual computers. Indeed, the history of computer science can largely be understood by studying the history of general purpose software and programming languages.

Specifying Complex Problems

Having a virtual computer is only the basis for solving problems. But what precisely is the problem to be solved? Broadly, excess paperwork in an office might be the problem. What is the solution? There are many ranging from closing the office to staffing it with robots that have been designed to relish paperwork.

Computer scientists use the term specification to mean a precise description of a problem and the nature of its solution. But, a specification is not a solution, for it is not constructive in the mathematical sense. A specification defines the scope of the problem, and exactly what constitutes a correct solution. It describes the allowable commands to a program and the actions that are to be taken in response. These actions include commands to output devices and changes to the computer's memory.

Specifications may be written entirely in mathematical notation, but they are usually written in a combination of both mathematics and English. For example, part of a specification of an airline system must describe the commands that reservations agents type, the responses to each command, and the changes that occur in the database. The specification does not say how the computer system should achieve these effects, but nonetheless, specifications can be very long and difficult to write.

Determining the specification of a problem to be solved requires thoroughly understanding the environment in which the computer is to be used. This is called "needs analysis" by Freeman and "system requirements analysis" by Boehm. This process is much more complex for computer systems than most other disciplines, since there are many possible problems a computer could solve. (After all, there are only so many alternative specifications for the "toast" problem). Furthermore, computers are used in very complex environments.
As we gain more experience with computers, there will be portions of specifications that are standardized. This is beginning today. For example, many programs that use a Macintosh will be required to have a particular user interface so users do not have to continually relearn different modes of interaction. Presently, some specifications also mandate certain levels of reliability, but standardized specifications are still a rarity. On the other hand, bridge engineers have a standard specification that has been written by a governmental body, the American Association of State Highway and Transportation Officials (ASHTO).

Trade-offs must be made between the quality of a solution, and the cost of producing it and using it on a computer. If the cost of producing a solution is greater than the cost of the problem, it is not worthwhile. Similarly, the cost of purchasing a sufficiently powerful computer and using it may be prohibitive. Because of the necessity of looking at these costs, there must often be iteration between the processes of specification and problem solving. At least, a person or group writing a specification must have some intuition as to the costs of solving it.

Before concluding this section, I should note that determining a specification for a problem is itself a type of problem. As such, the process of finding a specification is similar to designing a program or highway bridge. All of these problems require the use of problem-solving skills to decompose them into more tractable subproblems. Finding a problem specification requires analyzing the enterprise and finding simpler subproblems that can be solved by a computer. Writing a computer program requires decomposing a problem into basic operations that the virtual computer can execute. Analogously, designing a bridge requires determining how to assemble a collection of simple steel or concrete subcomponents. What varies are the simplest subcomponents into which problems are decomposed; these subcomponents are simpler problems, computer operations and steel/concrete members for specifications, programs, and bridges, respectively.

Solving Complex Problems on Simple Machines

I have suggested that a hierarchical decomposition must be performed to solve hard problems, be they problems in specification or in developing solutions. That is, problems must be decomposed into subproblems that are easier to solve, and the subproblems decomposed, and so forth until the remaining subproblems are directly solvable. There are a few approaches to decomposing problems, but all approaches start out with analysis or a thorough understanding of the problem. Based on this analysis, an artificial structure for the problem may be synthesized.

For the simultaneous translator problem I mentioned in Section titled Simple Machines, I
proposed a synthetic structure containing three sub-problems: transcription, translation, and vocalization. These subproblems can be further decomposed, again using analysis and synthesis. For example, transcription has sub-problems that include dividing the sound into phonemes, using a dictionary, and analyzing syntax; but it is such a difficult problem that good solutions are only now being found. Analysis of the airline database problem leads to a decomposition into the subproblems of user interaction, network communication, long term data storage and retrieval, reservations, billing, scheduling, etc. The subproblem of network communication has been decomposed into 7 distinct levels by the International Standards Organization (OSI)\textsuperscript{137}.

The decomposition of a difficult problem does not always proceed easily. Many decompositions are possible, and some do not result in good solutions. In some cases, no decomposition seems tractable. In the early 1960's, noted linguists and computer scientists thought they knew a decomposition for the natural language translation problem, but twenty-five years later, natural language translation is still an open problem. Certainly, even when decompositions are evident, it is necessary to analyze the decompositions, not only to permit further decomposition, but to verify the decompositions properly solve the problem at hand. In the case of computer systems, the decomposition also must be shown not to exceed reasonable computer memory or time bounds. Such analysis may lead to improved decompositions.

Though it is clear that we need to decompose a problem and ultimately express it in terms of simple operations, there are many possible methods we could use. We could start with the problem, create a decomposition and accurately describe each subproblem. In our example, we could begin by accurately describing the transcription subproblem and then doing the same for the translation and vocalization sub-problems.

Then, we have our choice as to what to do next. We could proceed depth-first and finish the solution to transcription. We could also work breadth-first and produce the next level of decomposition for the other subproblems before continuing. Both the depth-first and breadth-first solutions are instances of top-down design, since we start at the highest level problem and proceed downward. Advocates of top-down design usually favor the breadth-first approach, since this postpones the design of details until the entire structure has been thought out.

Another approach to the design process is called bottom-up design. Motivating bottom-up design is that we can determine primitive operations that support many different decompositions. In this case, we might analyze a problem but not produce a precise decomposition. Instead, we might determine some low-level operations that we expect to be
useful. For example, we may know a priori that the Fourier transformation will be a useful tool in recognizing speech, and we can begin by solving this problem first. Once primitive operations are made explicit, the higher level design will be more constrained and it may be easier to think concretely about it. Bottom-up design is really the basis for the standard software and programming languages that are written for a bare computer.

Designs usually combine bottom-up and top-down design techniques. The decision as to how to proceed is influenced by the goals of properly utilizing available personnel and reducing the completion time of the project.

Many authors, including Simon and Freeman, have written in detail about the particular analytic and synthetic activities used in solving problems. For example, they describe search, generalization, and restriction techniques that are part of the synthesis process. Search locates known problems whose solutions shed light on the problem in question. Generalization and restriction adapt known solutions by making them more general or more restrictive, respectively. These authors also describe the analysis techniques such as transformation, measurement, and validation. Transformation is the modification of a problem into a form that is more easily solved. Measurement provides insight into the behavior of a problem or its solution, and validation shows the correctness of a solution.

Here are some simple examples of where these techniques are used in the design of computer systems.

- **Synthesis activities**
  1. **Search.** Consciously or subconsciously, a designer of an airline database system considers other similar systems to find hints as to how to construct the new one.

  2. **Generalization.** A designer wants to develop a program that counts the number of occurrences of individual words in text. He has seen a program that counts the number of words in a document, and realizes it is straightforward to keep counts by individual words and to print the individual totals at the end. Or, an implementor may extrapolate the performance of a system in production use from its performance on relatively simple test cases.

  3. **Restriction.** An implementor of a program to diagram sentences uses the program that counts total number of words in a document only for its ability to divide a stream of characters into words. Realistically, he may have to generalize that portion of the program, so it will not discard punctuation.

- **Analysis Activities**
  1. **Transformation.** An implementor restates a problem in a new way that is more tractable. For example, an implementor may transform an English statement of
a problem’s solution into a particular programming language. Programming languages are mechanical transformers.

2. Measurement. An implementor might measure a physical system to determine its properties so as to learn more about it. For example, the implementor of the airline system may measure the number of reservations that are made per second to understand system throughput requirements. In addition, an implementor may have to measure the computer system to determine how well it is functioning.

3. Validation. An implementor must validate solutions to show they meet the requirements. Validation may be partially performed by testing, and partially by analytic procedures in which programs are verified to show that they work properly.

This is only a partial list of the techniques used to solve problems but it is illustrative. Simon notes that these techniques are shared with other disciplines and are common to many endeavors of man.

Design and Implementation

Once a requirements analysis has been satisfactorily made, it is possible to determine a specification, and begin decomposing the problem. In the decomposition, the subproblems at each level will also be specified and they have their own decompositions, unless they are problems that are solved directly by the computer.

Consider an incomplete top-down decomposition—one in which only simple subproblems are not decomposed. That decomposition could be termed a component-level design because it identifies the problem to be solved and describes the design of the major components and their interrelation. I consider the computer system design process to be the determination of problem specifications for a real world situation, and the construction of a component-level design.

The implementation process incorporates the remaining activities necessary to produce a working computer system. The first of these activities is programming. This is the process of filling in the remaining details in component-level design, expressing the design in a particular programming language, and properly utilizing the basic software that augments the computer. The second is the process of debugging and testing—finding and correcting errors either in programming or design. This process usually has many stages and involves designers, programmers, independent test personnel, and ultimate users. The third is tuning. Tuning is done shortly after a system has been made to work and results in small changes to make it work better.
Requirements Analysis

Problem Specification

Component-Level Design

Programming

Debugging/Testing

Tuning

Design Process

Implementation Process

Table 12-1: Major Steps in the Design and Implementation Process

The steps on the left are divided into the design and implementation process. Though the process generally progresses downward, there is iteration among the steps. For example, more work on problem specification may be required if it proves too hard to design an implementation. It is particularly painful when testing reveals a deficiency in a specification. There may also be parallelism in the process; for example, debugging of some portions of a solution may begin while other parts are still being designed.

Producing usable computer systems require activities other than design and implementation. These include documentation, training, and maintenance. Some documentation describes the systems to users, while others is intended for programmers who must maintain the system in the future. Good programmer documentation is critical to ensure low life-cycle cost for software, much as corrosion protection is needed for a bridge. Training goes beyond documentation and actively teaches an organization to use the system.

The discussion of designing and implementing computer systems has been limited to software, but hardware design or selection could be included. Computers vary in their execution speed, memory, and the input/output devices that they use. Permitting the designer the flexibility of choosing the best computer for a task provides an extra degree of freedom, but it does not change the nature of the design process. For example, if the designer is given the freedom to develop a computer, the primitive operations might be boolean operations on binary bits. These operations are even simpler than typical computer operations, and they would now form the lowest level of the problem decomposition.
Computer-Specific Knowledge

The computer system design and implementation process that I have described is fairly applicable to the design and implementation of many other types of systems. There may be slight differences, but many of the steps remain the same. For example debugging and testing might be replaced by client consultation in the case of architecture design\textsuperscript{139}. However, good problem solving skills are only a part of what is required to design computer systems. Good designers must not only be expert in problem solving, but also expert in a particular problem domain and the use of computers. *Flexible* designers must be adept at understanding new problems quickly.

It is not possible to list the domain-specific knowledge required for each domain that uses computers, but it is possible to list some of the computer-related topics that a computer system designer should know:

- **Available computers and software that augments them.** Designers should know the available hardware and the software that provides the virtual machines that can serve as the basis for their design. These include programming languages, operating systems, database management systems, etc.

- **Known abstractions.** Many broadly useful components of solutions have been developed. For example, queues and directories are data objects useful in many applications, as are certain mathematical procedures. These abstractions serve the role of vocabulary in language; they provide convenient names and concepts with which to design. Furthermore, abstractions have known component designs and there are frequently complete implementations available.

- **Mathematics.** Mathematical analysis plays an important role in both the analysis and synthesis components of computer problem solving.

- **Management.** Computer systems frequently require enormous manpower to design and implement. Experience is needed in project organization to utilize it effectively.

- **The Future.** Many computer systems that are designed today will be in use many years hence. However, they will be continually modified and upgraded. It is helpful to know the future (e.g., of technological advances) so that systems we write today will work in the tomorrow's environment.

Certainly, there are many other relevant topics. Many of these are described in a recently developed compute science curriculum developed at Carnegie-Mellon\textsuperscript{140}

A Case Study: The Space Shuttle Primary Onboard System

The space shuttle Primary Avionics Software System is the primary software system that runs on the space shuttle\textsuperscript{141}. It completely controls the vehicle at certain critical mission points and aids the crew at other times. It is designed to tolerate many failures, and there have been no catastrophes. I will use this large system, which required thousands of man-years to design,
Problem Specification

Launching men and machines into orbit in a reusable vehicle creates many problems that require computer solutions. Among many others, they include the computation of trajectories, the sequencing of the thousands of events that must occur prior to a launch, the ground monitoring of vehicle telemetry, and the control of the flight surfaces that permit the vehicle to fly.

The top-level specification divides these functions into three sets to be solved by three separate computer complexes. Broadly, the complex at the Kennedy Space Center in Florida controls all launch-related activities up until 25 seconds before launch. The complex at the Mission Control Center in Houston performs trajectory, and monitors the vehicle after launch. The onboard systems do all the fine grain control of the vehicle from 25 seconds before launch until landing.

The specification calls for two separate onboard systems. The first, or Primary Avionics Software System (PASS) normally handles all the functions of the shuttle, and must automatically tolerate a large number of failures. The specifications require that no single hardware failure should cut short a mission, and that no double hardware failure should jeopardize the crew. However, the PASS is a software system and as such, it is susceptible to logic errors. For this reason, another system was mandated. It is called the Backup Flight System (BFS). The specification called for it to be developed by a completely separate group, to minimize the likelihood of the same error being committed twice.

The specification of the PASS can be subdivided into the three areas. The first area contains the guidance, navigation, flight control specifications. There is a set for ascent, another set for on-orbit operations, and another for entry (return to earth). Another area is called "systems management". It is used when the vehicle is orbiting, primarily for opening and closing the payload doors, controlling the manipulator arm, monitoring and controlling the payload, and annunciating error conditions to the crew. The third set is for vehicle checkout and is used before launch to check out the various shuttle subsystems.

The specification for each of the areas can be decomposed. In the case of the shuttle, NASA wrote very detailed specifications. In fact, they are so detailed that they sometimes contain some system design. For example, the specifications for navigation contain equations that can be used to compute the vehicle's location. In all, the detailed specifications require 20 volumes.
Basic Components

The basic components that the designers had to work with included a collection of memories, 4 computational units, and a variety of input/output devices. A programming language, called HAL/S was also mandated. The memories were quite limited, and, there was almost no additional software that constrained—or aided—the design. A major hardware feature is that hardware components fail independently; that is, the failure of one component is unlikely to cause the failure of another.

Design

The structure of the design is similar to the hierarchical decomposition of the specification, but there are differences. For example, there is a special purpose operating system that underlies all of the activities of the PASS. As another example, there were screen management functions designed to format the displays; these, too, are used by many separate activities.

Implementation and Testing

The implementation generally followed the initial design, but there were times when implementation constraints forced redesign. The limitation on the amount of memory resulted in a much less elegant design that is much more difficult to implement correctly. In 1976, a prototype system was developed for flying the shuttle off the back of a 747. The performance of the prototype also resulted in design changes.

About 100 individuals were involved in designing and programming the PASS. Some worked for the seven years that predated the first flight in 1981. At some stages, an equal number were involved in analyzing and testing the software to ensure that it met specifications. These independent test personnel use a collection of computer tools to analyze the code and simulate missions.

Other activities

There are substantial software manuals\textsuperscript{142} that are the basis for major training efforts. The computer system is specified for use by experts and they must receive adequate training.

With monthly shuttle flights that each require slightly different software, software maintenance has become an expensive project. The difficulty is compounded by the need to carefully test each change. At least six separate software configurations are under development at any time.
Reflections on Design

There is no question that there are standard modes of thought used in designing complex systems. The processes of problem specification and design are similar in many disciplines, though different knowledge is used, and decompositions are expressed in terms of different primitives. For example, a recent case study of bridges performed by my colleague David Gifford and myself shows that the bridge design and implementation process has very clear problem specification, design, and implementation phases.143

In bridge design, standard specifications are typically provided by the ASHTO body mentioned above. A set of alternative bridge–specific specifications are presented to the client after a preliminary design144 has been completed. The design phase begins after the client has decided on the desired solution; this phase results in a complete bridge design. The resulting plans and construction guidelines are the basis for the implementation phase, which is performed by the lowest-bidding contractor. The analytic and synthetic methods, the tools, and the primitives of the decompositions, of course, differ substantially from those used by computer system designers.

Bridge design and computer system design are both engineering–oriented. A good question is whether the design techniques used in the fine arts have as much similarity. I would guess there is analysis and synthesis, but that many techniques are more intuitive. Artists presumably use less formal aesthetic considerations, whereas computer scientists use formal methods such as type–checking to analyze programs, and bridge designers use finite element methods to analyze bridges.

The physical sciences, which once aimed exclusively at understanding the natural laws, are becoming more synthetic as they increasingly seek to harness nature to solve more complex problems. Certainly, design activities play a central role there. However, even in the classical physical sciences, design is needed for hypothesis formulation, proof, and experiment formulation. Certainly, the synthetic and analytic activities on which design relies are used in all sciences and mathematics.

I conclude that the process of design, properly augmented by the details of design in certain domains, provides intellectually viable and interesting material for engineering, scientific, and liberal arts curricula. The process of design in computer systems is particularly interesting because of its complexity and diversity.

Acknowledgments
I wish to thank Jeffrey Eppinger, Bruce Horn, and Debra Lynn for helpful comments as I was writing this essay.
NOTES

135FreemanFundamentals, Boehm

136Of course, when we embed a computer in the toaster, the options become greater, for the toaster becomes a type of computer; we can conceive of it entertaining us while we await breakfast.

137ISO Paper

138Simon Artificial, Freeman Fundamentals

139Saarinen

140CMUCSC Curriculum

141Shuttle Case Study

142Space Shuttle Programs

143Bridge Case Study

144In the preliminary design phase, a requirements analysis is performed and various alternatives are considered.
13. APPENDIX

Mary Kay Shaw-Johnson

One can follow the tradition of the educational philosophy of the institution in its publications, position papers, committee minutes, and publicity.

I. Major publications of the philosophy

II. Annual report versions of policy statements.

III. Preparatory discussions in position papers and committee minutes.

IV. Further comments and reports in the Focus, Tartan, and the Carnegie Review.

V. Other topics relevant to the philosophy would be the Social Relations Program, the development of CHSS, and CMU's recent Strategic Planning program.

I. MAJOR PUBLICATIONS

CARNEGIE EDUCATIONAL PAPERS. 1945-1958

***** Contents:

• Plan of Development of the "Social Relations Program" of the College of Engineering and Science at Carnegie Institute of Technology.

• Professional Education in Engineering and Science.

• Doherty, Robert E., The Educational Task: The Challenge to Education: Reconstruction of Professional Education

• Education for Social Responsibility

• Education in Values.

• Smith, Elliot Dunlap, Fundamentals of Professional Education

• Professional Education in a Free Society

• The Education of Professional Students for Citizenship

• Carnegie's Attitude Toward Teachers and Teaching

• The Development of Humanistic-Social Education at Carnegie Institute of Technology

• The Place of the Teacher in Professional Education

• Teachers and Teaching.

• Warner, John C., Scholarly Growth.

• Teare, B. Richard, Jr., The Use of Problems and Instances to Make Education Professional.
Verplanck, D.W., Technological Education at Carnegie.

Wright, Austin. The Interplay Between Liberal and Professional Courses in a College of Engineering and Science.

Teare, B. Richard Jr., The Bond Between Humanistic-Social and Scientific-Technological Sources.

Smith, Elliot Dunlap, Slack, Robert C. and Ward, Paul L., The Role of Humanistic-Social Education in Making Professional Education Liberal at Carnegie Institute of Technology.

Steinberg, Erwin R., English Composition in the College of Engineering and Science.

Slack, Robert C., Introduction to Literature.


Verplank, D.W., Description of a Course in Engineering Analysis. Professional Education in Engineering and Science.

Professional Education in the College of Fine Arts.

Professional Education in Margaret Morrison Carnegie College.

Wright, Austin. Two-way communication in improving engineering education.

Slack, Robert C., Introduction to literature: course aims and teaching method in the required course in literature in the College of Engineering and Science.


Doherty, Robert E. The Development of professional education: the principles which have guided the reconstruction of education at Carnegie Institute of Technology as stated by its president Robert E. Doherty, 1936-1950. 31 p. Bibliography of the educational writings in social relations. December, 1937. Booklet, quoted from a memorandum submitted to the Maurice and Laura Falk Foundation. (85/SR/Do)

• Interpretation of Social Relations Program by Willard E. Hotchkiss.
• Pertinent concepts—Historical resume by F. Curtis Swanson. (85/SR/Te)


Elliot Dunlap Smith, Provost during the Carnegie Plan years, compiled notebooks on various grants, cataloged as *Educational development funds, 1948-1957*. 14 notebooks (76/Ed)

John C. Warner, assuming the presidency of Tech, carried on the Carnegie Plan for Professional Education into his term. His collected papers include "Carnegie and the future", address on the occasion of his inauguration as president of Carnegie Institute of Technology, 1950; "The American university and education for professional responsibility." Reprinted from *Chemical and engineering news* V. 27, June 27, 1949.—"Objectives of the curriculum for the professional training of chemists." Reprinted from *Journal of chemical education* V. 18. No. 12, December 1941. Speeches and articles kept in the Warner collection. (10/War)


Margaret Morrison Carnegie College. *Educational Objectives*, 1957. (82/Ed)


Conference on English Education. <English education for today's special concerns.> March 31–April 2, 1966. (85.1/Eng)


Center for Special Studies. *Plan for activation, October 15, 1971.* (includes initial faculty.) (88.5)


II. ANNUAL REPORTS OF THE PRESIDENT

Selected ones with pertinent passages on the educational philosophy include the reports for the years 1936/37, 1946/47, 1948/49, and 1949/50.

Annual reports of the deans of the colleges to the president for these years will have supporting data and arguments.

III. POSITION PAPERS AND OTHER PROPOSALS.

Chronologically:

Carnegie Technical Schools. Plan and scope of the proposed Carnegie School of Technology at Pittsburgh, Pennsylvania: preliminary report. Printed by order of the Board of Trustees of the Carnegie Institute, March, 1903. (01/Pl)


IV. SOCIAL RELATIONS PROGRAM

Social Relations Program. Basic course and social relations notebooks [by] Elliot Dunlap Smith. 1944–1947. 2 vols. Includes catalog statements, articles, reports, correspondence, annual reports of the Provost, index of file of materials on education, course outlines, minutes of Social Relations Program Committee. (85/SR/Smith/note)

Social Relations Program. Glen U. Cleeton, director. Materials concerning the Social Relations Program. Outlines, questionnaires, correspondence, reports, 1944–45. (85/SR/Cle)

Social Relations Program. Collection of communications, articles, reports, curricula. 1944–1950. 3 v. Analytical indexes precede text of each volume. (85/SR/Com)

Social Relations Program. Outline of social relations based on objectives of the program. First draft. January 20, 1943. (85/SR/Onj/1943)

Social Relations Program. Selected readings in problems in natural resources. (Social Relations C-712) 1942. (85/SR/C-712/1942)

Social Relations Program. Senior elective course outlines. 1940–1966. (85/SR/Cos)

Social Relations Program. College of Engineering. 1944. Description of content and
purpose of program. Course outlines, including topics covered and bibliographies. (85/SR/Des)

Social Relations Program Committee. Selected readings in human and natural resources. 1941. 3 v. Readings for Social Relations 712. (85/SR/C-712 1941)

Social Relations Program Committee. Selected readings in the study of intellectual tools. c1941. (85/SR/C-702/1941)

Social Relations Program Committee. Selected readings in the study of technology and society. c1940. (85/SR/C-701/1940)

V. HUMANITIES AND SOCIAL STUDIES/SCIENCES

Division of Humanistic and Social Studies. Annual Reports. (85/An)

Division of Humanistic and Social Studies. Core courses taught in H&SS by senior faculty members, Survey, 1963–64. (85/Cou)

Long and short-range plans for the Humanities and Social Sciences. 1964. (85/Plan)

H&SS Review Committee Reports. 1963. (85/Rev/1963)

Division of Humanistic and Social Studies. Proposed plan for reorganization into the College of Social Science and Industrial Administration. 1947. (85/Prop/1947)


VI. CURRICULA STUDIES

School of Applied Design. Minutes of the Curriculum Committee, 1913–1932. (86/Cur)


College of Fine Arts. Minutes of the Curriculum Committee. 1932–37; 1946–48. Continues the minutes when called School of Applied Design. (86/Cur)

College of Engineering and Science. Curriculum Committee. Minutes of the committee
on curriculum and scholarship requirements: on education. (81/Cur/Min)


VII. FACULTY SENATE MINUTES
Draft of a position paper for a new college of H&SS. Erwin R. Steinberg. FS Jan. 6, 1967

Goals Committee Report on H&SS. [Steinberg] FS Feb. 6, 1967

Committee on Improved Teaching Effectiveness. The goal: teaching excellence; a proposal for the faculty of CIT, CMU. FS Nov. 12, 1968


Schoenwald, Richard L. Teaching what we do: a goal for this university. FS May 21, 1969.


Faculty Senate. Educational Policy Committee. Minutes. 1942–45 through 1969–70. (25/Ed/Min)

VIII. SPEECHES, ADDRESSES, ARTICLES
Through the years as ideas have developed, faculty members and administrators have used speeches and articles to clarify the ideas and purposes of the institution’s educational philosophy for a broader audience of alumni and the wider academic community.
Blenko, Walter J. "The second education." Carnegie Institute of Technology Commencement address, February 3, 1946. (55/Ad/1946)


Prentis, H. W. "The r's of higher education" Commencement address, April 26, 1942. (55/Ad/1942)

Report of a committee on engineering education after the war. Reprntinted from the *Journal of Engineering Education* V. 34, no. 9, May 1944. (70/Re)


Doherty, Robert E. "Forfeited values." Carnegie Day Address, November 25, 1941. (51/Ad/1941)
Doherty, Robert E. "Where are we going?" Carnegie Day Address, November 26, 1946. (51/Ad/1946)


Hotchkiss, Willard E. "Pioneering new frontiers." Carnegie Day Address, November 28, 1939. (51/Ad/1939)
