The Program as Problem: Origins and Impact of CRS's Problem Seeking

This paper contributes to our understanding of the problematizing of architecture by focusing on how the task of defining the scope and nature of a future architectural project, commonly referred to as the "program" or "brief," came to be framed as defining a problem to be solved through design. This story takes us through post-World War II America, the design of the first schools to accommodate the baby boom, and the growth and management of Caudill Rowlett Scott, Architects, an architecture firm that pioneered an approach to architectural programming, that still influences the profession. The approach, known as "problem seeking," is now synonymous with managing overwhelming amounts of information associated with large and complex building projects. During the systems thinking era following World War II, in which management theorists struggled with understanding decision making under uncertainty, problem seeking emerged in response to the very real challenges met head on by an aggressive architecture firm that aspired to both design innovation and corporate growth. In covering the origins and impacts of problem seeking, this investigation helps us to understand how the problematizing of the program presents a challenge for contemporary education and practice.

BRIAN SCHERMER
University of Wisconsin–Milwaukee

ORIGINS

The formal establishment of the program as a "problem" for architecture occurred after World War II when social, economic, and technological factors transformed the design and construction of buildings into a complex endeavor in critical need of data management and consensus building.

With the advent of the large organization clients around the beginning of the 20th century, firms like Albert Kahn Associates of Detroit were organizing their office management structure to be able to cope with more complex projects of their clients such as Ford and General Motors. Larger organization clients, both in the private and public sector, delegated responsibilities for decision-making and approval to committees, rather than a single powerful individual.[1] Bigger and more complex buildings as well as a growing cast of specialists and consultants required a more comprehensive and systematic approach that would not only account for technical complexity, but also ensure the means to achieve agreement.

In addition, societal consensus about basic functional building types was evolving. Impulses toward democratization and social reform meant that it was no longer possible to accept taken-for-granted assumptions how building should be organized. Museums, for example, during the 1920s were the subject of intense discourse about their role in the United States. Museum curators, board directors, and architects debated about whether museums should maintain their role as palaces for wealthy patrons and the well-to-do, or whether they should they could be democratized by catering to the lower and middle classes, even to the point of being modeled after department stores.[2]

Finally, after World War II, economic expansion, the population boom, the suburbanization of communities, and the growing technological complexity of buildings brought the "problem" of programming to a head. Assumptions about education and the nature of community were challenged. There was pent-up demand to catch up with the booming economy and the phenomenal growth of "war babies" to absorb into public schools. These created new opportunities and challenges for architecture firms. Among the challenges was the need to efficiently and effectively redefine school buildings. The formal framing of architectural programming as a "problem" emerged in the context of school design in the post-war U.S. And, there is one firm, through its early beginnings to its maturity as one of the largest firms in the world, which has come to be most associated with this shift.

CRS AND EARLY PROGRAMMING

The architecture firm most associated with, and indeed credited with formalizing a process for programming—and equating programming with defining a problem to be solved through design—was Caudill Rowlett Scott, Architects (later known as CRS and derivations thereof).[3] The firm was established in 1946 by a group of young professors at Texas A&M University.[4] To position the firm for public schools, one of the founders, William Caudill had written a book called Space for Learning.[5]

This book critiqued the poor heating, lighting, and general inadequacy of traditional school building models. For Caudill's part, traditional school design was anathema to what children needed.

"Kids aren't interested in fancy architecture. All they know is light, air, comfort, and color…. Let's give them logical schools, with scientific lighting, heating, and air movement. Let's not make them go to school in monuments."

Traditional school buildings were based on multi-story, double-loaded corridor design for urban neighborhoods. While this model held through the early part of the century, it was becoming obsolesced by the new reality of automobile-based suburban communities. The book piqued the interest of potential clients, including the school board of the Blackwell, Oklahoma public schools. The project was also highlighted in the national. A September 1950 article in *Collier's Weekly* magazine, captured the sense of urgency in Blackwell and the rest of America:

"Blackwell had to meet an emergency which faces us all. The problem posed by the horde of war babies now coming of school age and rushing with a thunderous patter of little feet into our already crowded elementary schools. In the next 10 years these youngsters will raise the nation's public-school enrollment from 21,000,000 to 28,000,000."[6]

The *Collier's* article was largely complimentary towards Caudill and lauded him and Blackwell for pioneering the most advanced elementary schools in the country. It reported that Caudill called for schools with lots of natural light and broad overhangs to shelter them from sun. The building was positioned to optimize solar orientation and to shelter play areas from prevailing winter winds. Caudill argued that teaching methods had changed and that classrooms had to become more flexible and homelike. And, he claimed that school buildings must be located according to demographic needs and situated to accommodate outdoor play space and room for expansion if needed.

According to *Collier's*, an inquisitive school board member learned about Caudill from a faculty member at the University of Oklahoma. Caudill and the school board commissioned a master plan to account for population growth over the next 20 years and analyzed appropriate sites based on growth trends. Funds for new construction would be limited. With a small budget, the firm responded with a rather spare, modernist module of classrooms with a shed roof, clerestory windows with fixed louvers, sloped ceiling with acoustical tile, floors with radiant heating, and green chalkboards instead of black ones. To save money, there were outdoor corridors and a large outdoor covered play shed. There was also a lobby and an auditorium with mechanical ventilation. *Collier's* reported that the townspeople were shocked by the "cow shed" appearance. But, after an initial failed referendum, a second attempt passed by a ten-to-one margin. The Blackwell schools, along with other examples in California and Illinois represented a new model of education for the baby boom generation.

As triumphant as the story is for the development of schools, it is also represents a watershed in the way firms interacted with clients. In his 1971 book, *Architecture by Team*, Caudill recounts the story of how the firm's commission for two elementary schools for Blackwell, Oklahoma necessitated rethinking of their process for client engagement.

"It came early. We were working on our first school project—two elementary schools for Blackwell, Oklahoma, 525 miles away from our office over the grocery store in College Station, Texas. We were having a most difficult time getting the preliminary plans approved. It seemed that we made at least four round trips trying to get the board to say "yes." It was always "no." Patience, enthusiasm, and money were running short. Finally I said (at least I'll take credit for it) to Wallie Scott, "Wallie, we are going to lose our shirts if we don't do something quick. How about you and me loading the drafting boards in your car (my car was so old it wouldn't stand the trip), driving to Blackwell, and squatting like Steinbeck's Okies in the board room until we get the damn plans approved?" So we did…. The main thing that happened was this: In trying to find a way to lick the distance problem, we happened upon a truth that should have been obvious to us all the time—the client/users want to get into the act of planning, and when they do there is no reason to get approval because that is automatic. The communication problem is solved."[7]

This simple truth evolved into an elaborate approach for client and user engagement that they referred to as "squatters." CRS continued to organize its practice around the squatters process in order need to gather information and create consensus as quickly and as systematically as possible. According to an interview

1

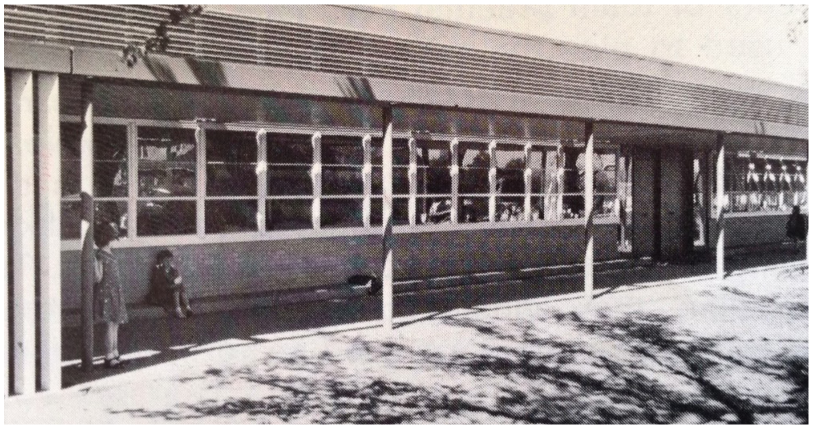


Figure 1: The first school designed for Blackwell, Oklahoma. Design began in 1948.

of William Peña in the book, The CRS Team and the Business of Architecture," Caudill had learned the technique of intense planning and design while working for the U.S. Army Corps of Engineers.

One of the key bottlenecks in the process was the need to define the scope of the project. Caudill enlisted the help of a fourth partner, William Peña, to develop a program foe another school project. Peña recounts the assignment in *The CRS Team and the Business of Architecture*:

"Bill said, 'Willie, I want you to go to Elk City and come back with a program,' so I felt it was my responsibility. But I really didn't know what I was going to have to do to come back with a program. There were no rules, no criteria. But he said, "You know: How many classrooms? How much money? Site survey, et cetera." So, I went, [and] sure enough, I enjoyed the hell out of it. I interviewed the teachers, superintendent, and some of the school board members. I got the information I needed, came back, and gave the information to Bill. Two or three weeks later, he went up there to design the school."[8]

Over the coming assignments, the programming process continued to evolve. In December of 1950, the CRS team was working on a high school for Norman, Oklahoma. Working in a motel room, Caudill asked Peña to display the information from his interviews on the wall. Later in the week, Caudill wanted to show Peña what he done with the information:

"Look what I've got," [he said]. He had six salient considerations for the design of the Norman High School. We didn't recognize it at the time, but it was a statement of the problem. He asked, "What do you think of this?" He had these six statements not related to the solution, not design solutions. They stated a condition. For example: "Since the students in a high school spend a lot more time in the halls than they do in an elementary school, if you spend ten minutes of every period there and have six periods, you have sixty minutes in the halls. Therefore, the halls should be made part of the teaching process and exciting.' He had six of this kind of statement. Later—much later—we looked back on them and said, "That was the first statement of the problem; Bill had avoided the solution."[9]

Peña retrospectively points to this as the revelatory moment wherein intensive data gathering and analysis led to the formulation of a problem statement, one that dealt not only with the quantitative problem of fitting square footage onto a site, but the qualitative issue to address through design.

"ARCHITECTURAL ANALYSIS"

After several more years of practice, their approach to programming had become

2

more codified. In 1959, Peña and Caudill co-authored an article in *Architectural Record*.[10] In it, they argued that a mere list of spaces was insufficient to the task of solving architectural programs. An architect, they argued need to be an analyst or diagnostician in order to clearly define the problem to be solved through design. Analysis, they wrote, must be "complemented with the creative before a really good architecture can be produced."

Peña and Caudill were first and foremost concerned with distinguishing client's wants from needs. Like a doctor, they argued, the architect had to address, not what the client says they want, but what they actually need. The author's admitted that they could not articulate how to distinguish wants from needs, other than it required sound judgment and analytical ability. But, they did offer a series of steps: (1) make your client part of the planning team, (2) acquire background research, (3) and develop the art of interrogation, by which they meant meeting with clients often, focusing on problems and not solutions, and visiting projects to help identify problems, not to mine ideas for solutions.

They also expressed the importance of writing down ideas on paper. "It has been our experience that you can get the fuzz off a fuzzy idea by writing about it. We also believe that our design ability improves when we learn to organize our thoughts in simple, clear words."

Peña and Caudill were searching for the proper terms to describe the process. "Architectural analysis," captured some of the essence for the moment. They also explained that programming was akin to something with which architects were already familiar: writing specifications. But, they were also attempting to elevate the analogy by articulating the notion that it was specification writing for "an architecture."

"What are the specifications for an architecture? The designer whether he is involved in the analysis phase or not (we thinking of course he should be), receives actually little guidance and most certainly no inspiration from a listing of "spaces required by the program." His job is to create an environment, beautiful and workable—not just shuffling around some oversized dominoes. People live and work in those dominoes. The designer therefore should be as interested in qualitative space as in quantitative space. Writing specifications for an architecture is putting down on paper in words the requirements of qualitative space."

Creating "an architecture" meant addressing both the quantitative and qualitative aspects of design. Note also that here they advocate the involvement of the designer in the programming process. In later developments of their process,

Figure 2: Schoolchildren participating in a squatter's session in Columbus, Indiana, 1971.

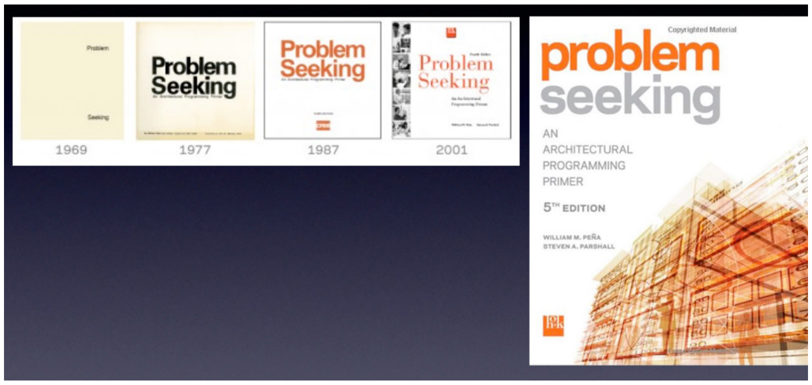
they began to see the value of greater separation between stating the problem (programming), and problem solving (design).

"PROBLEM SEEKING"

Ten years after the publication "Architectural Analysis," the CRS strategy of linking of programming to systematic data gathering, objective analysis, and consensus based decision making was formalized and disseminated with the publication of the first edition of the book, *Problem Seeking*.[11] First published in 1969 and now in its fifth edition, Problem Seeking represents a kind of orthodoxy of the systems-based reasoning approach to architectural programming.

Before diving into the details of the book, it is worth exploring the origins of the phrase "problem seeking" itself. A Google NGRAM search suggests that the phrase was relatively obscure until well into the 20th century.[12] The use of "problem seeking" as a compound verb, first came to be associated with process of defining a problem and as a healthy habit of an inquiring mind was popularized in literature pertaining to high school science teaching in the 1950s.[13] The phrase was also used by Abraham Maslow's characterization of one of the key functions of scientific inquiry: "problem-seeking, question-asking, hunch-encouraging, hypothesis-producing.[14] Sometime between 1959 and 1969, the rather dry term, "architectural analysis," was supplanted by the more evocative "problem seeking."

3



Perhaps more important than the origins of the phrase is the intellectual atmosphere from which the problem seeking approach evolved. Postwar economists, management theorists, and psychologists since the end of the war set about the task of understanding rational decision making under uncertainty. By the 1950s, the growing consensus was that the classical economic model of man as rational actor had lost credibility, especially when analyzing the ability of entrepreneurs and managers to make decisions. People are not omniscient. Decision making ability was seen as limited by what Herbert Simon called "bounded rationality."[15] Problem seeking also attempted to cope with the growing realization that design problems are "wicked" by nature.[16] Wicked problems have a high degree of ambiguity, and they tend to reveal unforeseen factors as one attempts to solve them. *Problem Seeking* was a bold attempt to tame what could be tamed about them. Against this intellectual backdrop, as well as the post war boom, the process of problem seeking emerged.

Figure 3: All five editions of Problem Seeking

THE LOGICAL REASONING OF PROBLEM SEEKING

The foundation of the problem seeking approach to programming is systematic, logical, and nuanced reasoning. The assumption was that architectural projects are so information-laden and complex that one must apply systematic means of data gathering and analysis in order to manage it all.[17] By being systematic, one is assured that one is able to gain a measure of accountability over the many disparate factors that must be addressed in the design, avoid missing critical information that without which the project would not be deemed a success, and to avoid the dreaded information overload condition, which Problem Seeking labeled "data clog."

In addition, the problem seeking approach sought to create a "dialogic space" to enable key stakeholders to participate.[18] Key decision makers met in intensive "squatter' sessions in order to hammer out the basic scope and nature of the project. Information was comprehensively gathered and systematically allotted into the cells of a matrix, which are displayed on large pieces of brown wrapping paper for all to see. The columns of the matrix were labeled: "facts, goals, needs," and the rows were labeled: "form, function, economy, and time." Key concepts were labeled and adroitly diagrammed on white "snow cards" and arranged logically on the matrix.

Problem Seeking describes programming as analysis: as a first and necessary stage of problem definition. Design, on the other hand, involves a synthesis of programming facts in order to solve that problem. From a problem seeking perspective, one might say that one creates architecture by formulating an objective problem statement and then solving that problem through the intuitive and subjective processes of design. After all, the authors argue with seemingly airtight logic: How can one solve a problem unless one has stated it first?[19]

IMPACT

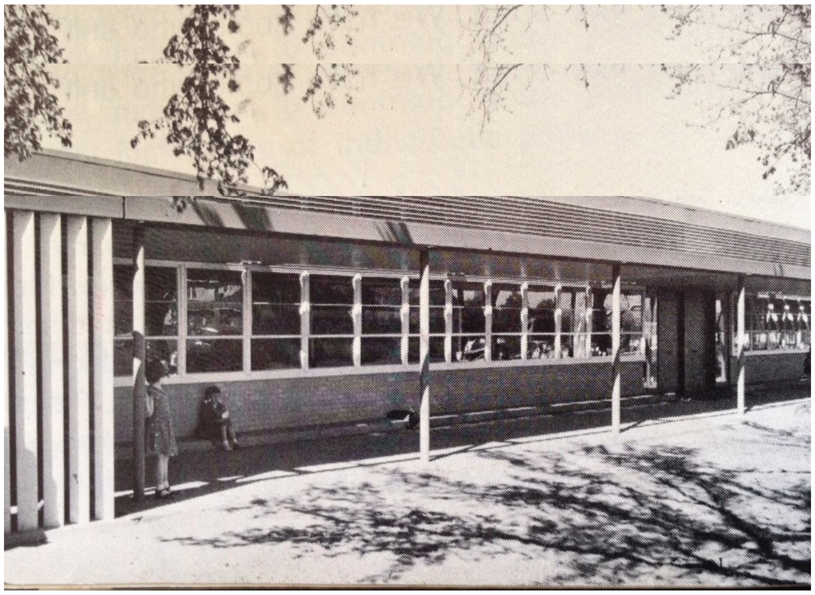
Problem Seeking has had a profound impact on the profession in terms of both professional licensure and pedagogy. In 1973, the National Council of Architectural Registration Boards (NCARB) incorporated its precepts into the professional licensing exam. NAAB required the teaching of programming as a condition for school accreditation (a requirement which is still retain in the newly adopted 2014 Conditions). In 1970, CRS ranked 21st in size among U.S. firms. In 1974, it became the first architecture firm to sell public stock. By 1980, it had become the country's largest firm. Via mergers and acquisitions it had so diversified its operations that in 1994 the architecture business, along with the copyright to *Problem Seeking*, was sold to HOK. The book is now in its 5th printing.

TEXTBOOKS

Problem Seeking inspired several textbooks on architectural programming that attempt to improve on its formula. Edith Cherry's (1999) *Programming for Design: From Theory to Practice* expands problem seeking beyond the analysis-synthesis framework discussed in the Problem Seeking primer. She urges programmers to try to gain a fuller understanding of a future architectural project by employing what she calls "versatile thinking." Versatile thinking is characterized by deliberate shifts between various modes of sense making: analysis and synthesis, induction and deduction, Eastern and Western thought, and so on. Such nuanced thinking can be useful in breaking through conceptual logjams and confirming hunches to support decision-making. Versatile thinking also helps to triangulate our knowledge and thus avoid pitfalls such as tunnel vision and "groupthink."[21]

Relying on a singular mode of thinking, Cherry argues, is a threat to reliability because it may filter out crucial information and discourage innovation. Programming through versatile thinking, she states focuses our attention on seeking what is unique about a project and gathering information in order design

4



for specific needs and contextual factors. One must be wary of falling into the trap of merely accepting what has worked before as necessarily the best solution. Reason-based programming is especially helpful to determine what is unique about the project at hand that may make past solutions invalid.

Another well-known textbook on programming, *Architectural Programming and Program Manager*, by Robert Hershberger expands Problem Seeking in a different way.[22] Hershberger's critique of the problem seeking approach is that while it works well as an agreement-based strategy, the emphasis on form, function, economy, and time, is too limited. Hershberger advocates what he terms a "values-based" approach to programming, in which the knowledge to be generated from programming is filtered through a designer's perspective. Hershberger writes:

In some cases, the interest in being systematic in developing knowledge about users may tend to obscure issues of importance to the design architect. Similarly, the fact that oftentimes the design architect has yet to be hired prevents the designer's expertise and values from influencing the program.

Far from tainting the program, Hershberger argues that omitting values from the equation can result in over-emphasis on the technical aspects of the project and leave little for designers to go on once design begins. This potential imbalance requires some means of redress in the programming process. Defining the architectural future by interpreting it through a filter of what Hershberger calls the "enduring values of architecture" will better inform designers in expressing and perpetuating the "essential purpose of the human institutions" that sponsor building projects. Hershberger' approach, which incidentally has also been summarized in the chapter on programming in a recent edition of the AIA's Architects Handook of Professional Practice, provides leverage for interpretive knowing by highlighting the symbolic, institutional, functional, technical, environmental, temporal, and financial aspects of a project.[23] Thus, interpretive programming

Figure 4: Diagram from Problem Seeking.

Programming is analysis. Design is synthesis.

ENDNOTES

1. Gutman, Robert. *Architectural Practice: A Critical View*. New York, NY: Princeton Architectural, 1988.

2. DiMaggio, Paul. "Constructing an Organizational Field as a Professional Project: U.S. Art Museums, 1920-1940." In Powell, Walter W., and Paul DiMaggio, (Eds). *The New Institutionalism in Organizational Analysis*. Chicago: U of Chicago, 1991.

3. Palmer, Mickey A. *The Architect's Guide to Facility Programming*. Washington, D.C.: Institute, 1981.

4. King, Jonathan, and Philip Langdon. *The CRS Team and the Business of Architecture.* College Station: Texas A & M UP, 2002

5. Caudill, William W. Space for Teaching. No. 9. Vol. 12. *Bulletin of the Agricultural and Mechanical College of Texas*, 1941.

6. McQuade, Walter. "The Little Red Schoolhouse Goes Modern." *Collier's Weekly* (September 9, 1950): 42-43, 65-67.

7. Caudill, William Wayne. *Architecture by Team; a New Concept for the Practice of Architecture*. New York: Van Nostrand Reinhold, 1971. 311-12.

8. King, Jonathan, and Philip Langdon. *The CRS Team and the Business of Architecture.* College Station: Texas A & M UP, 2002

requires examining the full "architectural potential" of the design problem, not just its functional and technical requirements aspects.

CONCLUSION

The critiques of the problem seeking approach are many. By positing architectural design as a solution to a consensus-drive programming problem, it undermines the romantic image of the architect as an autonomous artist. Problem seeking is said to artificially divide programming and design. Its internal logic obscures the notion that the process of defining a problem simultaneously solves it.[24] Problem seeking is too technocratic.[25] It privileges the analytical and the objective over the intuitive and subjective. Its narrow focus on form, function, economy, and time obscures other, more "architectural values," such as aesthetics, meaning, and sustainability. It is said to restrict design creativity. By promoting programming as a specialization, the problem seeking approach has contributed to the splintering and diluting of the profession. And, problem seeking leads to emphasis on narrow, wishful futures rather than long-term strategies.[26]

Despite the wariness with which architects have come to regard it, *Problem Seeking* continues to shape the profession's ideas about programming. It highlights the importance of organizing for consensus in order to get buildings built. Most importantly, programming, or the potential insights that derive from it, offers a means to add real value to clients and provides an avenue for architects to differentiate themselves in an era in which design services are increasingly commodified. The conundrum for teaching programming is similar to the one that Peña and Caudill described about "architectural analysis" and that many ascribe to the teaching design itself. It is possible to outline the steps, but more difficult to teach real insight.

9. King, Jonathan, and Philip Langdon. *The CRS Team and the Business of Architecture*. College Station: Texas A & M UP, 2002
10. Pena, William, and William W. Caudill. "Architectural Analysis: Prelude To Good Design." *Architectural Record* (May 1959): 178-182.
11. Peña, William, and John Focke. *Problem Seeking: New Directions in Architectural Programming*. Houston: Caudill Rowlett Scott, 1969.
12. "Problem Seeking." Google Ngram Viewer. Web. 10 Sept. 2014.
13. The High School Journal, School of Education of the University of North Carolina, 1951; *Educational Leadership*, Volume 15. Department of Supervision and Curriculum Development, N.E.A., 1957; Brandwein, Paul F. *Teaching High School Science: A Book of Methods*. Harcourt, Brace, 1958.
14. Maslow, Abraham H. *Motivation and Personality*. New York: Harper, 1954.
15. Simon, Herbert A. *Models of Man: Social and Rational; Mathematical Essays on Rational Human Behavior in Society Setting*. New York: Wiley, 1957.
16. Rittel, Horst W. J., and Melvin M. Webber. "Dilemmas in a General Theory of Planning." *Policy Sciences 4* (1973): 155-69. Churchman (1967) "Guest editorial: Wicked Problems". *Management Sci*. 14(4), p. 141-142.
17. Palmer, Mickey A. *The Architect's Guide to Facility Programming*. Washington, D.C.: Institute, 1981.
18. Schneekloth, Lynda H., and Robert G. Shibley. Placemaking: *The Art and Practice of Building Communities*. New York: Wiley, 1995.
19. Peña, William, and John Focke. *Problem Seeking: New Directions in Architectural Programming*. Houston: Caudill Rowlett Scott, 1969.
20. Cherry, Edith. *Programming for Design: From Theory to Practice*. New York: John Wiley, 1999.
21. Janis, Irving L. *Groupthink: Psychological Studies of Policy Decisions and Fiascoes.* Boston: Houghton Mifflin, 1982.
22. Hershberger, Robert G. *Architectural Programming and Predesign Manager*. New York: McGraw-Hill, 1999.
23. Haviland, David S. *The Architect's Handbook of Professional Practice*. Washington, D.C.: American Institute of Architects, 1994.
24. A critique attributed to Herbert McLaughlin, founding principal of KMD, San Francisco.
25. Personal communication with Frank Duffy, founding principal of DEGW, London.
26. Brand, Steward. *How Buildings Learn: What Happens after They're Built*. Harmondsworth: Penguin, 1995.