

A LOOK AT PROGRAMMING

Every building has come into existence only after a need for it has been articulated. From the beginning of human civilization, architects have prepared rudimentary or complex instructions on how to create buildings for shelter, leisure, worship, and commerce. Beginning in the eighteenth century, buildings became more complex but it was not until the Industrial Revolution at the beginning the nineteenth century that specialized building types came into existence. This growth of specialized building types contributed significantly to the increased need for comprehensive programming activities.

Contemporary interest in programming began in 1966 when the American Institute of Architects published its first booklet on the subject of programming entitled, *Emerging Techniques of Architectural Practice*. While AIA has declared programming as a separate and important activity within the architectural delivery process, AIA has not formally adopted a position that would elevate programming to the level of a standard basic service on the B-141 agreement. (A forthcoming article will explore the impact of AIA's position. For example, did you know that architecture firms are losing millions of dollars a year in potential services fees to accounting and business consulting firms who offer strategy and

planning services to facility executives?)

The definition of architectural programming varies widely among scholars and practitioners. Some view programming as a necessary and prerequisite step whereby the programmer uses a systematic process of gathering and analyzing information about a building or setting, and then uses that information to create guidelines for the performance of that setting. Others view programming as separate from design and one that should be performed by the building's owner. Still others perform cursory programming activities and rely on drawings, matrices, and diagrams to show basic departmental adjacencies, space requirements and circulation patterns. The American Institute of Architects has defined programming as the process of leading to a statement of an architectural problem and the requirements to be met in formulating a solution. Generally, the programming process involves steps of data collection, data analysis, data organization, communication of data, and evaluation. Each program is a customized statement of needs and aspirations relating to the building. Robert Hershberger, FAIA, professor at the University of Arizona College of Architecture, Planning, and Landscape Architecture, has defined programming as the first stage of

the architectural design process in which the relevant values of the client, user, architect, and society are identified; important project goals are articulated; facts about the project are uncovered; and facility needs are made explicit.

According to Hershberger there are four basic programming approaches that architects use today: the Design-Based Architectural Model, the Knowledge-Based Architectural Model, the Agreement-Based Architectural Model, and the Value-Based Architectural Model.

The Design-Based Architectural Model assumes that programming and design occur simultaneously. Usually the architect meets briefly with the client to gain a general understanding of the project's needs and programming proceeds as the architect creates varying design concepts for the building. The client selects the design concept that best suits his needs and the designer modifies the design as the design process continues toward the construction documentation phase.

During the late 1960s, a second programming model emerged as the Civil Rights Movement enveloped our country. The Knowledge-Based Architectural Model assumes that the ensuing design solution should support the needs of the inhabitants of the building. This design movement arose to correct misalignments between people and

the built environment especially for large complex buildings such as prisons, hospitals, and governmental office buildings. This approach has never gained much momentum, perhaps due to its emphasis on the social sciences. The third model described by Hershberger is the Agreement-Based Architectural Model. This approach to programming assumes that the programmer is a facilitator of knowledge. He or she meets with key individuals such as CEOs, senior executives, or building committee members and directs them to acquire the necessary programmatic information needed for the design phase of the project. The key to the success of this approach is in the programmer's ability to gather accurate information from the client and relay this information to the designer of the building. The most notable example of this programming approach is seen in the problem-seeking approach of William Peña. The problem-seeking model emphasizes the separation of programming and design: programming being analysis and design being synthesis. The model involves over 140 considerations categorized into five major programming steps: goals, facts, concepts, needs, and problem. The fifth step, which states the problem, is the transition point where programming meets design. The five steps are each filtered through four major considerations: function, form, economy, and time. Hershberger outlines one final approach to programming and he

refers to it as the Value-Based Architectural Programming Model. Value-based programming is simply the accumulated strengths of the first three approaches discussed above. Hershberger refers to his model as the "HECTTEAS" model, which is the acronym for the value areas he tests during the programming phase of design. The value areas are:

1. Human: Functional, social, physical (ADA), and psychology.
2. Environmental: Site, climate, context, resources, and waste.
3. Cultural: Historical, institutional, political, and legal.
4. Technological: Materials, systems, and processes.
5. Temporal: Growth, change, and permanence.
6. Economic: Finance, construction, operations, maintenance, and energy.
7. Aesthetic: Form, space, color, and meaning.
8. Safety: Structural, fire, chemical, personal, and criminal.

Regardless of the differences among definitions of programming or differences among theoretical approaches to programming, one common *assumption* does exist among the majority of architectural programming approaches such as those described in the preceding paragraphs and that is: *The primary purpose of programming is to determine the needs and requirements for the design of the*

proposed building. Applying this assumption further, the assumption that underlies a large percentage of architectural design theories and corresponding design activities is that the purpose of architecture is to create a "great" piece of artwork - a beautiful object. In other words, many design professionals approach building design from the vantage point of the building itself. Supporting this perspective, architect Phillip Johnson has said, "The job of the architect is to create beautiful buildings. That's all."

In a forthcoming article, we shall explore an alternative view of architecture and one that assumes that the building is an object *and a subject*. This alternative view also assumes that the building is an active participant in the lives of social actors and that it can act as an agent for organizational, communal, environmental, societal, and generational transformation. And, perhaps most fundamentally, this alternative view assumes the architect engages in the practice of "place-making."

About the authors:

Valerie Bugni is an organizational and social researcher with Lucchesi, Galati Architects, Inc. Comments or inquiries may be sent directly to her by email at vbugni@lgainc.com or she may be reached by telephone at 702-263-7111.

Dr. Ronald Smith is the Chair of UNLV Department of Sociology and may be reached by email at smith@unlv.edu or by telephone at 702-895-3322.
