ROUTLEDGE REVIVALS

Integrating Programming, Evaluation and Participation in Design

A Theory Z Approach

Henry Sanoff

Routledge Revivals

Integrating Programming, Evaluation and Participation in Design

First published in 1992, this book is about making connections that may lead towards a new professionalism, since the past several decades have given rise mainly to new kinds of specialists in the areas of programming, evaluation, and participation. The implications for such integration are far reaching, with profound future effects on the physical environment, the design professions, and the education of designers. The book is split into four sections dealing with facility programming, several forms of evaluation, participatory design, and the application of Theory Z principles. This book will be of interest to students of architecture and design.

Page Intentionally Left Blank

Integrating Programming, Evaluation and Participation in Design

A Theory Z Approach

Henry Sanoff

Routledge
Taylor & Francis Group

First published in 1992
by Routledge

This edition first published in 2016 by Routledge
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN
and by Routledge
711 Third Avenue, New York, NY 10017

Routledge is an imprint of the Taylor & Francis Group, an informa business

A Library of Congress record exists under LC control number: 92029044

ISBN 13: 978-1-138-20338-9 (hbk)
ISBN 13: 978-1-315-47173-0 (ebk)

Integrating Programming, Evaluation and Participation in Design

A theory Z approach

HENRY SANOFF
North Carolina State University

Avebury
Aldershot · Brookfield USA · Hong Kong · Singapore · Sydney

Published by
Avebury
Ashgate Publishing Limited
Gower House, Croft Road
Aldershot, Hants GU11 3HR
England

Ashgate Publishing Company
Old Post Road
Brookfield, Vermont 05036
USA

A CIP catalogue record for this book is avaialble from the British Library and the US Library of Congress.

ISBN 1 85628 338 0

Printed and Bound in Great Britain by
Athenaeum Press Ltd., Newcastle upon Tyne.

Contents

Foreword

Professionals are not known by what thay do, but by the way they do it. Professionalism lies in expertise and expertise relies on skill, method, and knowledge. While many areas of design such as architecture, landscape architecture, and interior design are established professions, much of what professionals do, have been and will continue to be done by lay people. Experience and knowledge resides with lay people as much as with experts. In fact, humanity has done without design professionals for a long time, and probably would continue to survive in their absence.

While designers have thought of the built environment as dependent upon them, it is becoming increasingly evident that their decision domain is shrinking. This is partially explained by the designers neglect in being sufficiently responsive to human needs. It is clear that designers cannot be responsible for everything, nor can they control everything. Yet, it is apparent that the role of the professional is continually being questioned, as is the issue of human accountability as it has been practiced. Creating environments that are more responsive to human needs will require changes in traditional practice. Practice in the future promises to offer new challenges, since the role of the professional will need to be multi-faceted. Designers will require new skills and knowledge as an enabler, technical advisor, social worker, and bureaucratic trouble-shooter. Learning how to listen, not only to the paying client, but to people who use and are effected by the environment, within the social and historic context, can produce a professional with an expanded capacity for shaping the future. Habraken (1986) envisions this new role, not only to study the health and well-being of the physical environment, but to help it become better through design intervention.

This book is about making connections that may lead towards a new professionalism, since the past several decades have given rise mainly to new kinds of specialists in the areas of programming, evaluation, and participation. The implications for such an integration are far reaching, with profound future effects on the physical environment, the design professions, and the education of designers.

1 Facility programming

Programming is generally viewed as an information processing system setting out design directions that will accommodate the needs of the user, the client, the designer, or the developer. The nature and scope of design decisions and design information have changed rapidly, and the role of programming has thus changed as well. The uses of programming have been extended from primarily acquiring and organizing information (Heimsath, 1977) to investigating and developing information, analyzing client and user needs, and evaluating projects after construction and occupancy (Freidman, Zimring, & Zube, 1978; Wener, 1989).

Programming has been seen as a valuable resource for a systematized process that provides a structured framework for accumulating and classifying data. As an analytical process, it encourages decision making through objective procedures rather than on individual assumptions or personal prejudices. A report by the Building Research Board of the National Research Ciouncil (1986) on programming practices states that 'programming services may not always result in new construction or changes to the physical building, but in organizational or managerial changes that achieve the same objectives' (p.1).

There is considerable diversity in the use of the terms *space programming, facility programming, and functional programming* and in their meaning within the design professions. In addition to the differing terms that identify programming, there are also philosophical differences regarding the

An earlier version of this paper was first published in Zube, E.H. & Moore, G.T. (eds.), 1989, *Advances in Environment, Behavior, and Design*, Plenum, New York.

of *programming*. Expressions such as 'programming is design, programming is not design,' and 'programming is getting ready for design,' underlie the diversity of purposes and places of programming in the design process.

Everyone, however, would accept the view that a program is the organized collection of specific information that involves developing, managing, and communicating. Most will also agree that programming is the process of identifying and defining the needs of a facility. Although general features have emerged, programming should be recognized as a dynamic and interactive process.

The basic references for programming lie in the published volumes of Palmer (1981), Pena (1977), Preiser (1978, 1985, 1992), Sanoff (1978), and White (1972). The principal objective of Preiser's edited collection *Facility Programming* (1978) and *Programming the Built Environment* (1985) is to provide an authoritative overview of the user-oriented programming approaches that are currently to be found at work in architectural and environmental design. Both books describe the professional programming activity conducted by architectural and programming firms. The topics covered range from problem definition, cross cultural programming, and post-occupancy evaluation to adaptive reuse and other more specific examples of programming. Each chapter is largely self-contained and represents various attitudes about programming and about the breadth, scope, and prospects of the field.

As a complement to the Preiser volumes, Sanoff's *Methods of Architectural Programming* (1978) is about the technical aspects of programming. Here the material moves from data-gathering techniques, through methods of synthesizing and organizing data, to a field application of programming techniques that makes use of user expertise. The volume stresses a general flexibility of approach, in which the techniques may be combined and merged, depending on the situation on hand.

Palmer's *Architects Guide to Facility Programming* (1981) covers much of the same ground as the Preiser and Sanoff volumes, though attempts to integrate information-gathering techniques with case studies. Although it was written 3 years latter than Sanoff's book, it fails to show the development of facility evaluation studies and their relationship to facility programming.

By contrast, Pena's *Problem Seeking* (1977) is a presentation of one approach, the CRS (Caudill, Rowlett, and Scott, Architects and Planners) method of programming. The five-step process is presented in lock-step fashion with partially worked examples. Rather than adopting a flexible attitude toward the organization of information, the book views the role of the programmer as being conformity to a predetermined format.

In an effort to bridge the theory-practice gap, White (1982) conducted interviews with architects to assess their attitudes toward programming practice. In an open-ended telephone survey of 73 architects in the United States, programming strengths were described by the respondents as including 'thorough rigorous analytic process, strong client/user participation, programming tailored to each project, strong integration with design and successful projects, and happy clients.' The recurring programming problems were reported to be 'finding the true needs of the client, getting clients to make decisions, and clients don't appreciate programming at program-design connection' (p.37).

According to White (1983), clients are frequently unaware of their needs and sometimes are impatient or do not understand why the information is needed. Some clients will not permit the staff to participate in the programming process. There are also occasions when the client says one thing and the programmer hears another; the client often faces difficult choices and is not prepared to work as quickly as the process requires. These difficulties cited by practitioners clearly suggest the need for considerable improvement in communication with the client, as well as collaborative involvement in the programming process.

When asked about the reasons for offering programming services, beyond providing a better building, the respondents in the White study said that such a service 'facilitates the design process, marketing, project management tool, client confidence in project and firm, and saves client and firm time and money' (p.52). The majority of architects agreed that programming provided the client a way to participate in the project's planning process. They also indicated that programming leads to improvements in the client's operation; therefore, programming firms sometimes serve also as management consultants. Hellmuth, Obata & Kassabaum (Silver & Klein, 1985) described the programming process as being able to 'promote confidence, teamwork and consensus; convey a sense of care and attention to users; minimize space-allocation abuses; and leave a clear trail of decisions and approvals for superiors to review' (p.3).

Comparing programming models

Architectural programming is not a rigidly defined process. Each programmer has his or her style and emphasis, and each project requires a certain amount of custom fitting of any model that a programmer may have. Because a number of programmers have described their approach, the purpose of this section is to examine seven of these approaches, to compare their similarities and differences, and to combine them into a composite model.

The seven models

The models reviewed are by Davis & Szigetti (1979), Farbstein (in Palmer, 1981, p.33), Kurtz(in Preiser, 1978, p.41), McLaughlin (1976, p.42), Moleski (in Palmer, 1981, p.37), Pena (1977, p.43), and White (1972, p.44). These authors were selected because their programming work is documented in the literature and they follow a well-delineated process.

Gerald Davis. Davis's programming model (Davis & Szigetti, 1979) is a 21 step process (Figure1.1) that begins with preprogramming and moves through evaluating the facility in use. It is directed toward the planning of corporate facilities. The first part of the process, as with Moleski and Kurtz (below), is to become familier with the client's business organization, operation, activities, and needs, both present and long range. The programming steps include gathering data on the operating facilities; on physiological needs, such as those affecting health, safety, and performance; and on behavioral require-