17. Computational Modalities Of Design Evaluation

Yehuda E. Kalay

Computer-Aided Design Research Center
State University of New York
Buffalo, New York, 14214, U.S.A.

Evaluation can be defined as measuring the fit between achieved (or expected) performances and stated criteria. It is complicated by the multi-criteria and multi-level modalities of design, where an overall balance of performances is preferred to maximizing the performance of a few characteristics, and where evaluation must be performed at different design phases, each characterized by a different informational profile. Each design modality requires a different approach to evaluation: the Multi-Criteria modality requires evaluation of a proposed solution at a particular design phase from multiple points of view, while the Multi-Level modality requires the evaluation of a particular performance characteristic at several different design phases. This paper discusses the multi-modal nature of evaluation and prediction in design, exemplified by some of the approaches that have been proposed to support them computationally. It then argues for the need to develop an integrated, multi-modal design evaluation paradigm.

Introduction

Design, as a discipline, formally recognizes our ability to influence the future, and our responsibility to do so in the "proper" way. The process of design, therefore, is the purposeful and conscious specification of actions that ought to be taken in order to achieve some desired conditions. Specifying the actions that will lead to achieving some desired human wants or needs by complex systems such as buildings is, however, difficult, because it involves political, psychological, social, economical, environmental, technical, and many other variables. There is seldom a formula that can be used to specify the desired performances of any one of these variables, let alone specify a desired combination of their effects and interactions. Rather, the process of design can be likened to *exploratory search*, where alternative courses of action are hypothesized, and their effects are predicted and evaluated by comparing them to the desired objectives. The evaluation may find that the predicted effects achieve the objectives, and thus validate the proposed course of action. More often, however, the predicted effects do not achieve the objectives, or they conflict with each other. In such cases the actions must be modified so they will achieve the objectives, or the objectives must be revised to match the predicted effects. The design process can thus be said to comprise three major tasks, performed iteratively: (a) defining a set of objectives that the proposed actions should achieve; (b) specifying actions that, in the opinion of the designer, will achieve the objectives; (c) predicting and evaluating the effects of the proposed actions to verify that they are consistent with each other, and that they will achieve the objectives.

This paper focuses on the third task, whose centrality to the process of design stems from the fact that prediction and evaluation are the "glue" that connects and binds the other two components of the triad: specifying the actions that achieve a set of objectives cannot be separated from predicting and evaluating the expected effects of these actions, because the evaluation guides the generative process toward achieving the stated objectives, uncovers opportunities to be explored, and indicates tradeoffs that must be made in order to improve the overall quality of the solution.

Prediction and evaluation are, therefore, important means for defining and clarifying the objectives of the design process, as well as guiding it toward achieving them. They rely on the designer's ability to predict the environmental, psychological, social, economic, and other effects that will ensue from executing the specified actions, evaluate the desirability of these effects, and derive operational conclusions from the evaluation. To evaluate the correspondence between the effects of the proposed actions and the specified objectives, the designer must translate between two different representational schemes: the *physical* representation that is used to describe the designed artifact, and the *functional* representation that is used to describe the design objectives. For example, to evaluate the coherence, or "readability" of a proposed floorplan, the designer must predict the psychological effects of a yet non-existing building, represented by geometrical and topological means, on the mental disposition of the "average" user, which is comprised of characteristics such as memory, prior experiences, and capacity for learning. This translation is error-prone, because the behavior of people is context-dependent, time-dependent, and experiential.

To alleviate these difficulties, the search for methods that can predict the effects of design decisions have began in the 1960, seeking a synthesis between the designer's creative and rationale faculties. These methods, comprising a series of logical and mechanical operations would, according to their early proponents, free the designer from the tedious evaluative aspects of his work, to the great advantage of his creative powers (Jones 1980). They would provide instant feedback on the effects and side effects of decisions made through his creative skills, and thus allow him to adjust these decisions to achieve the objectives. Furthermore, a rational and deductive approach to the processes of prediction and evaluation would make it possible to identify a certain number of operations that could be automated. Consequently, proponents of such evaluative and predictive methods tried to model them in a way that would also take full advantage of the potential offered by computers (Cross 1977).

Three decades of research have, nevertheless, failed to realize this aspiration. In fact, the considerable knowledge gained in predicting and evaluating specific aspects of design (e.g., energy, behavior in space, etc.), remains largely unutilized in everyday architectural design practice, and most building projects do not enjoy the fruits of the considerable expertise gained in similar design enterprises (Carrara et al 1991). This failure is due, to a large extent, to the isolation of the study of evaluation from the other components of the design process, and subsequently, our failure to integrate seamlessly the ensuing evaluation methods and tools with the other components of the design process. In fact, it is the position of this author that the centrality of evaluation to the design process actually *precludes* its study (and the implementation of computational methods that result from it) independently from the study of the design process as a whole.

Recently, some promising research avenues have been developed that recognize the complexity of evaluation, and its dependency on other components of the design process. To understand the motivation behind these new methods of design evaluation, and how they can be integrated into a comprehensive, computational design framework, we must better understand the role and the process of evaluation itself, and how it is used in the design process. This understanding will further not only design evaluation research, but also computer-aided design in general.

What is evaluation?

Evaluation is a term that has many meanings, and is subject to many interpretations, which range from measuring the performance of buildings (or other artifacts), to measuring the performance of the people who design them (Goldschmidt 1991). They also range from interpreting evaluation as *retrospective diagnosis*, in the sense of post-occupancy evaluation of completed projects (Jockusch 1991), to interpreting evaluation as a process of *dynamic transformation*, in the sense of mapping between different representational schemes (Manning and Mattar 1991). In between these extremes we find the "usual" meanings and interpretations of evaluation, as the counterpart of generating design solutions, and as critique of the decisions that lead to their completion.

Evaluation can thus be understood as a process that compares what *has been* achieved (or is *projected* to have been achieved), to what *ought to be* achieved. (What ought to be achieved, known as *objectives* or *goals*, is itself subject to evaluation. In the context of this paper we shall only deal with the evaluation of design solutions.) Evaluation, therefore, can be defined as *measuring the fit between achieved or expected performances, to stated objectives*.

Evaluation is a proactive, operational term, not only because its accomplishment requires considerable effort, but also because it is a dynamic component of the design process itself (as well as of other processes, such as medical diagnosis). Evaluation is called for when a given state of affairs is known to be, or is suspected to be deficient in some non-obvious way (e.g., the patient is sick, but the cause for his illness is not known), and while there is still hope and desire to improve it. The purpose of evaluation, like the purpose of medical diagnosis, is to establish what could and should be done to improve the current state of the designed artifact, and possibly to determine what steps ought to be taken. Conversely, if the solution is known to be optimal (because it was, perhaps, generated by an algorithm that guarantees optimum), or if we are resigned to the fact that it cannot be improved (e.g., in the case of a terminally ill patient), then there is no need to evaluate.

The process of evaluation can, however, only be applied to a given, specific set of performance characteristics, such as the form, composition, and location of a building, much like medical diagnosis can only be applied to the physical condition of a patient. When evaluating hypothetical design alternatives, where performances are not yet in evidence and cannot, therefore, be measured directly, evaluation must be preceded by *prediction*. Prediction is the process whereby the expected performance characteristics of buildings (or other artifacts) are simulated, hypothesized, imagined, or otherwise made tangible, and hence can be subjected to evaluation. For example, the rate of heat loss through a given building envelope must be predicted, often by way of simulation, before an evaluative procedure can determine whether this rate is acceptable. Likewise, a building model must be subjected to earthquake simulation before its non-elastic behavior can be evaluated.

In practice, prediction and evaluation are so inseparably tied to each other that they are often considered to be one and the same. Energy analysis by means of heat loss simulation has been, for decades, considered the undisputed method of "evaluating" the energy performance of buildings, as has been cost and structural evaluations. Moreover, implicit evaluations are often confused with prediction, such as in the case of appraising the suitability of a building to support different stages in a family's life-cycle. Yet, it is the opinion of this author that evaluation and prediction should not be confused, so they can be studied separately, for each has its own characteristic methods, and distinctive knowledge-bases.

Prediction, in most cases, is domain-specific. Although different predictive simulations may rely on similar techniques (e.g., queuing models, stochastic methods, and various forms of linear programming), predicting the loss of energy through the building envelop (Jog, 1991, Kalisperis 1991), for example, is not similar to predicting way-finding in a building (Gross 1991, O'Neill 1991). Evaluation, on the other hand, is much less context-dependent. Since it basically measures the difference between *achieved* performance and *desired* performance, evaluation can be regarded as a neutral process, and general methods to perform evaluation can be developed (Simon 1969).

Nevertheless, to draw operational conclusions from a specific evaluation it must be combined with domain-specific information. Hence, a method that will mediate between different competing or conflicting design objectives must, by definition, be domain-specific, and incorporate both evaluative and predictive measures. A discussion of evaluation and prediction in design, therefore, cannot be separated from a discussion of the design process itself. In particular, the partitioning of the design process into discrete "phases," or levels of abstraction on one hand, and the domain-specifity of design decisions on the other, define a framework for studying evaluation in design: different modalities of design require different modalities of evaluation and prediction.

The modalities of design

It was stated earlier that design can be viewed as a process that results in specifications whose physical or organizational implementation will produce a solution that meets certain pre-defined objectives. Architectural design usually must meet a wide range of design objectives. Each objective has its own technological, environmental, social, economical, and other requirements, and each has been the subject of intensive study, and even specialization, over the years. For example, the objective of making buildings energy-efficient has enjoyed great popularity since the late 1970s, due to the worldwide energy crisis. Likewise, much work has been done in the area of earthquake damage mitigation in structures, which has become the focus of considerable research efforts.

These individual objectives, however, are not independent of each other: when they are combined in the context of the built environment, design decisions that are intended to meet one objective may support or interfere with the achievement of another objective. For example, to achieve optimal energy efficiency, buildings have been earth-sheltered. But this design solution interferes with accessibility objectives, creates many technological problems, and may have psychological drawbacks. The designer must, therefore, achieve the stated individual objectives while reconciling their conflicting effects and side-effects, to the benefit of the design solution as a whole.

The difficulties induced by these conflicting demands are further exacerbated by the hierarchical nature of the design decision-making process: broad and general solutions are contemplated first, and are gradually refined as the design progresses. Alternatively, designers combine desired details into new wholes. In both cases, constraints are propagated "up" and "down" different levels of the abstraction hierarchy: high-level organizational decisions constrain lower-level details, and decisions that concern details limit the designer's freedom in selecting high-level organizational schemes (Liggett et al 1991). Design theorists refer to different levels in the design abstraction hierarchy as "design phases." Such phases typically include feasibility studies, conceptual design, design development, detailing, and design documentation. While the exact definition of each phase varies among researchers, their existence has been widely accepted since the early 1960s (Asimow 1962, Gregory 1966, Broadbent and Ward 1969).

It is but a small step now to recognize that hierarchical stratification is an attribute that applies not only to the design process as a whole, but also to its individual components. For example, a circulation system in a building is typically first planned schematically (using such terms as *circular*, *linear*, *radial*, etc.), then refined and articulated gradually into a hierarchy of major and minor traffic zones, which are further refined into a hierarchy of public and private circulation spaces. These are later detailed by adding doorways, fire zones, means of egress, and so on.

Taken together, the composition of design processes can be said to consist of two separate *modalities*:

1. The modality in which the designer moves between different levels of the abstraction hierarchy, searching for a composite solution that achieves multiple design objectives.

2. The modality in which the designer considers separately particular design objectives.

Figure 1 depicts (schematically) this dual modality: each horizontal "band" represents a particular design phase, and each vertical "band" represents a particular design objective. The bands are overlaid, to show the close coupling between the two modalities.

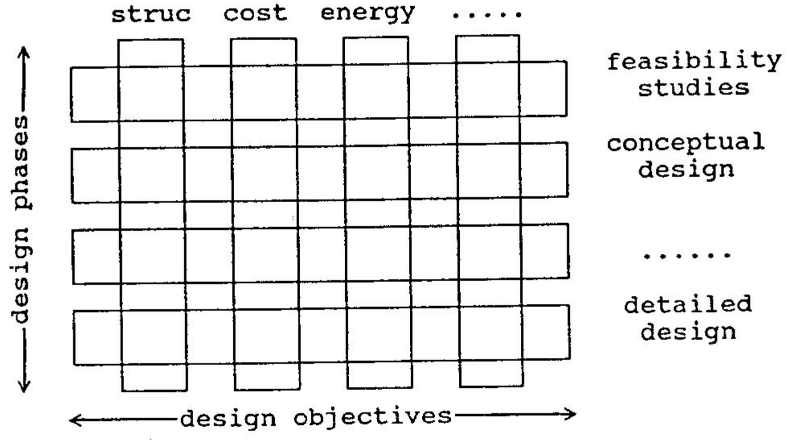

Figure 1. The dual modality of design.

Designers engage both modalities alternately and in parallel: it is often difficult to engage one modality without, at the very least, be cognizant of the other (although the experts whom the designer consults are typically oblivious to design objectives other than their owns). Nevertheless, for the purposes of understanding the design process and its evaluation, the distinction ought to be maintained. Furthermore, as it will be demonstrated later, the distinction between the two modalities is acknowledged, at least implicitly, in most software systems that were developed to support computationally the design process.

The modalities of evaluation

Given the different modalities of design, and given the centrality of evaluation to the design process, it should come as no surprise that the *evaluation* of design solutions follows closely the modalities of the design process itself. Hence, we distinguish between *Multi-Criteria* evaluation, and *Multi-Level* evaluation.

The Multi-Criteria evaluation modality examines a given design solution from *several different* points of view (e.g., energy, cost, structural stability, etc.). Because it evaluates a *particular* design solution, this modality is limited to a single design phase. Hence, it corresponds to a horizontal band in Figure 1.

The Multi-level evaluation modality examines how the design solution, or a succession of design solutions, satisfy a *particular design objective* (e.g., energy). Given its narrow focus, the same single-criteria evaluation must be applied at *different phases* of the design process. It corresponds, therefore, to a vertical band in Figure 1.

Multi-Criteria evaluation

Proposed design solutions, as well as realized design projects, exhibit many performance characteristics and must meet many objectives. The individual performance characteristics rely, nevertheless, on common physical attributes (e.g., layout, materials, budget). Attempts to maximize the performance of one characteristic, therefore, influence the performance of others. The designer must consider all (or at least many) of the building's performance characteristics together, in order to maximize the overall performance of the solution, and to avoid inadvertently diminishing the performance of some characteristics due to over-achieving others.

When the evaluation shows that all design objectives have not been equally-well satisfied, the designer must establish the relative importance of competing or conflicting objectives. Such prioritization will allow him to determine the overall merit of the design solution, and compare it to other candidate solutions. The designer must also determine the marginal returns of relaxing certain criteria in order to gain in others. For example, when the budget established for the purpose of constructing a single family house requires choosing between a third bedroom and a family room, which one of the two will the client prefer to have?

Multi-Level evaluation

The difficulty in designing large and complex artifacts such as buildings necessitates the use of hierarchical abstraction and stepwise refinement methods as means of partitioning the design process into discrete, manageable "chunks," which are called *design phases*: typically, a design solution is first conceived schematically, before it is further developed and detailed in subsequent design phases. The emerging design solution must, therefore, be evaluated at different levels of resolution: early in the design process, it may not be sufficiently well articulated to allow the application of detailed evaluation procedures, nor would it be meaningful to do so if it were possible. Rather, a generalized appraisal of the designed solution may be in order, to verify that the approach chosen by the designer is at all correct. Conversely, in later phases of the design process, it may be possible and desirable to perform detailed evaluation of the performances of the proposed solution. In intermediate design phases, where the design solution is only partially developed, it may be prudent to use evaluation methods that can operate with whatever information is available to them, and provide results that are meaningful for the particular design phase. In other words, different evaluation methods are needed to perform evaluation of specific criteria at different design phases.

Additionally, partitioning the design process into discrete phases also means that some decisions must be made prior to others, hence before all the details which may effect them, and which are effected by them, are known. It may be very useful to know in advance what are the implications of early design decisions on the solution in later phases of the design process, before much effort is spent on their adaptation to the particular conditions of the project, and before other decisions that rely on them have been made. However, while it is possible to consider multiple different aspects of the designed artifact at a particular phase of the design process, it is virtually impossible to consider *all* aspects at *all* phases. It is possible, however, and even desirable, to consider *individual* design characteristics in depth, from top to bottom, anytime in the design process. In early design phases such consideration will help select approaches that are likely to prove productive later on, while later design phases will benefit from reflection upon the reasons for adopting a particular approach rather than another, thereby helping the designer to select details that better support the chosen approach.

Software implementations of different evaluation modalities

Software systems that perform Multi-Criteria or Multi-Level evaluation of design alternatives have been under development for many years. Recently, some evaluation programs that demonstrate a range of possible approaches to modality-conscious computational evaluation have been developed. In the following, we shall discuss a few examples of each.

Multi-Criteria evaluation tools

When designing a software system that operates in the Multi-Criteria evaluation modality, one is often faced with the choice (and the desire) to use existing evaluation tools that were developed by experts in their respective disciplines. Typically, each evaluation program requires different input parameters, presented in a particular format, and produces output in its own unique way, which makes comparing its results to those of other evaluations difficult. This makes the use of existing evaluation tools difficult, often to the extent where it is deemed easier to redesign each and every evaluation to fit the integrated system. Obviously, while this approach solves the integration problem, it necessitates rewriting each evaluation program from scratch.

Another problem faced by designers of Multi-Criteria evaluation software is weighing the performance of different characteristics in order to derive the overall performance measure of the design solution. While it can be argued that the overall value of a design solution is a subjective measure, it is, after all, what designers (and their clients) are expected to do when faced with the need to choose among multiple options.

Wiezel and Becker have chosen an integration approach with relatively-weighted individual performance criteria, utilizing a database generated by a commercial CAD system (ARC+^{TM}) to describe the building at some given design phase. They developed their own set of evaluation procedures, which provide feedback on the thermal, acoustical, fire safety, and lighting performance of the building (Wiezel and Becker 1991).

In contrast to this approach, a group of Carnegie Mellon University researchers preferred to rely on a host of existing evaluation programs, and developed a common architecture that provides the necessary data for the application of these evaluators (Fenves et al 1991). Evaluations are performed by sending data to each evaluation module, and collecting the results from the modules centrally, for use in subsequent design operations. The integration is based on a blackboard model, which provides a shared memory for both the evaluation and the synthesis modules. Each module is a self-contained knowledge-based system, which performs both evaluative and generative tasks in its particular domain of expertise. For example, a module called ARCHPLAN, an architectural planning expert (Schmitt 1988), assists in the development of a design concept. A module called CORE (Flemming et al 1988), evaluates and generates layouts of elements in the service core of the building. The overall architecture of the system allows each process to maintain its own identity, and to follow its unique approach, while providing mechanisms for integration that steer these processes towards a common goal.

A hybrid approach, which combines the advantages of both integrated evaluators with stand-alone ones, was developed by Hacfoort and Veldhuisen (1991). Their system, called COSMOS, provides quick, sketchy evaluation of a building using built-in, integrated evaluation tools. It also allows the user to request in-depth evaluation of the same criteria using stand-alone, external evaluation modules, developed elsewhere.

Most of these (and other) Multi-Criteria evaluation systems strive to provide the designer with an overall measure of the performance of the design solution. While recognizing that such a singular measure is meaningful only for comparative purposes, it is useful, nevertheless, to show the designer how improving certain aspects of the solution may hurt other performance aspects.

Wiezel and Becker have assigned a weight factor to each design criterion (Table 1), which is multiplied by the grade, or degree of satisfaction of that criterion as reported by the evaluators. The weighted grades of all the evaluated criteria are summed up, providing an overall measure of the performance of the particular design solution.

Hacfoort and Veldhuisen chose to establish the binary relationships between each pair of design criteria, using one of two methods to describe the relationships between the different performances. The first method is based on linear least-square equations to describe the relationships, generating a scattergram which shows the criteria that are most seriously affected (both positively and negatively) by the proposed design actions (Figure 2a). The second method is based on a contingencies table, which shows the degree to which each design criterion is effect by the design parameters (Figure 2b). The use of these two kinds of measurements allows them to express different kinds of relationships between the design parameters (ratio, interval, ordinal, and nominal). For example, the relationship between costs per cubic meter and the mean thermal insulation of the building can be expressed through the least-squares method, while the relationship between the quality of the view and the acoustic insulation (in Db(a)) can be expressed by means of the contingencies table method.

Another solution to the problem of comparing the results of different evaluations, without actually computing an overall measure of performance, was proposed by Schmitt (1987), who used a pie-chart like visual presentation method, based on dividing a circle into a number of equal sectors that correspond to the evaluation criteria (Figure 3a). The circle's radius represents the normalized threshold for acceptability of the performances of the different criteria: for criteria such as first cost, heating, cooling, and the use of electricity, it represents the preset budget. For criteria such as structural stability, circulation, visual and contextual orientation, it represents the minimum standard of acceptability. If the radius of a particular sector is larger than the radius of the circle, then the criteria it represents performs better than the acceptability threshold (e.g., the energy consumption is lower than the established energy budget). If, on the other hand, the radius of the sector is smaller than the circle's radius, then the criteria it represents performs more poorly than the acceptability threshold (e.g., the circulation area is excessive). When the performance of some criterion is not quantifyable, Schmitt uses a rule-based system to determine the circle's and the sector's radii. This method of visualization also allows for representing weighted differentiation between the criteria, by varying the degree sizes of the sectors that represent them (Figure 4b).

	Building parameter	Enhancing criterion (in terms of "Prevention of")	wn(%)
1	Acoustic insulation of the external envelope	noise penetration from the outside	8.5
2	Thermal insulation, and resist. to fungi of int. coverings	mould growth due to condensation in the dwelling	8.5
3	Acoustic insulation of inter-dwelling partitions and floors	noise penetration from neighboring dwellings	7.6
4	Balance between thermal insulation and TTC of envelope	excessive expenditures for cooling during summer	7.4
5	Fire endurance of external envelope	fire penetration into the dwelling from outside	7.4
6	Medium TTC of external envelope	excessive heat at noon during summer	7.4
7	Low fire spread of interior coverings	fire spread in the dwelling	7.2
8	Location of thermal insulation close to the inside	excessive cold in the evening during winter	7.2
9	Low TTC of the envelope	excessive heat in the evening during summer	7.0
10	Thermal insulation of external envelope	excessive cold at night during winter	7.0
11	Thermal insulation, and high TTC of envelope	excessive expenditures for heating during winter	6.6
12	Location of thermal insulation close to the inside	excessive cold in the morning during winter	6.3
13	Acoustic insulation of intradwelling partitions	noise penetration between rooms within dwellings	6.1
14	Acoustic insulation of intradwelling partitions	being heard in other rooms (privacy)	5.9

Figure 2. List of building parameters and their weighing factors (from (Wiezel and Becker 1991).

Multi-Level evaluation tools: The major problems in developing software that can assist designers in performing Multi-Level evaluation of individual design characteristics stem from the different informational characteristics of the design phases they must span. They include the propagation of constraints "up" and "down" the design abstraction hierarchy, and the selection of software tools that are most suitable for performing the evaluation at each design phase.

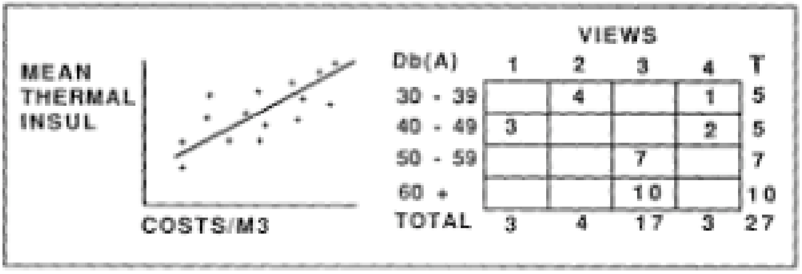

Figure 3. (a) Scattergram; (b) Contingencies table (from (Hacfoort and Veldhuisen 1991)).

The propagation of constraints across different levels of the abstraction hierarchy involves *translations* of design constraints between the different representational languages that are used at each abstraction level. Each abstraction level typically uses a particular symbolism to represent its informational content. Schematic design phases, for example, tend to use single line drawings and "block" symbols to denote such information as walls, rooms, or even the style of a column. Detailed design phases, on the other hand, use multiple lines to denote walls, rooms are typically furnished, and columns are fully articulated. The transition from one level of abstraction to another is not merely syntactic, but rather it is a semantic process, since the symbols convey different amounts, and even different kinds of information: while the line representing the wall in the schematic phase

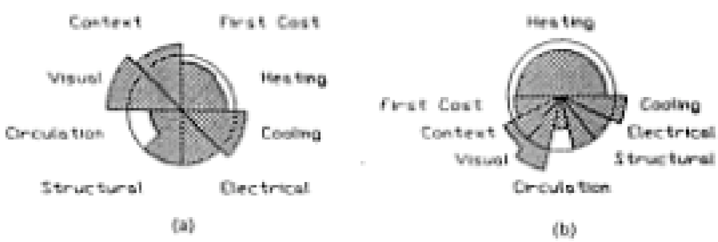

Figure 4. Building criteria visualization: (a) unweighted; (b) weighted (from (Schmitt 1991)).

may convey only the intent of partitioning the space, the fully articulated wall in a detailed design phase conveys both the commitment to partitioning the space as well as the means to be employed for that purpose (dimensions, materials, finishes, etc.). An evaluation tool that can operate across multiple abstraction levels must be capable, therefore, of interpreting the semantic information conveyed by each level, and to apply it properly.

The different informational characteristics of different abstraction levels also determine the kind of software tools that can be used to perform the evaluation: Early design phases may not contain sufficient information to support energy or structural simulation of the designed artifact, because they lack information concerning materials, finishes, or even exact dimensions. The use of defaults to fill-in missing information may actually be misleading, if the building does not conform to the averages from which the defaults were derived (Shaviv and Kalay 1991). Nor is it meaningful to perform such evaluation, and to provide the designer with "exact" heat loss information or precise sizing of structural elements, even if it were possible to calculate them precisely. Rather, the designer may be interested only in the effects which the orientation of the building will have on its general energy consumption, and whether the chosen structural schema is plausible. On the other hand, if the information needed to perform detailed evaluation is available, chances are that a detailed answer is expected.

Several systems were developed that perform Multi-Level, single criterion evaluation of design solutions. A system developed by Liggett, Mitchell, and Tan, called TOPDOWN, deals with the functional interpretation of architectonic shapes, such as columns and floorplans (Liggett et al 1991). It allows the designer to check the degree to which the evaluated element fulfills its desired function (e.g., floor area, structural support, etc.), in any phase of the design process. TOPDOWN deals with the problem of translating the semantics of the design solution between different levels of the abstraction hierarchy by employing shape grammars that tell the system how to refine an abstract, high-level subsystem into a specific arrangement of detailed, lower-level subsystems, and how to assemble a set of detailed, lower-level subsystems into higher-level ones (e.g., how to detail a schematic column into a combination of base, shaft, and capital, as depicted in Figure 5).

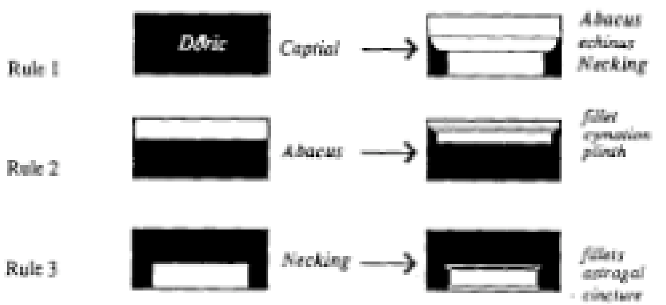

Figure 5. Rules for top-down refinement of a schematic design (from (Liggett et al 1991)).

A default-based approach to multi-criteria evaluation has been proposed by Johnson (1991) to calculate the cost of a building at different phases of its design process. His method handles the different informational profiles of the design solution at different phases of the process by arranging the defaults in an hierarchical object database, where lower levels of the hierarchy correspond to more detailed design solutions. The system can thus apply general floorprint area calculation in early design phases to compute gross cost per square foot, and refine this estimate by adding more accurate dimensioning and material costs as the design solution becomes more detailed over time.

The reliance on defaults for evaluating design solutions in early design phases was criticized by Shaviv and Kalay (1991), as potentially misleading when the design solution does not conform to the standardized cases from which the defaults were derived. Instead, they proposed to use a combination of rule-based expert systems and simulation modules for evaluating the thermal properties of a building. Expert systems can be used in early design phases to provide a gross, "rule of thumb" evaluation of the proposed solution's thermal properties. In intermediate design phases, expert systems can help select appropriate default values for use in simplified thermal properties simulation models (LCR and SLR). In later design phases, sophisticated thermal analysis software can be used to calculate precise thermal performance, and to provide for accurate evaluation (figure 5).

The system deals with the different informational levels of the design solution in different design phases by relying on a knowledge-base where the effect of each building parameter on the thermal behavior of the building as a whole have been stored, according to the design phase in which it is first considered, and the phases in which it is reconsidered (Figure 6). The knowledge-base also contains rules constraining the selection of subsequent building parameters, given the decisions that were made earlier. These rules, which operate bi-directionally (i.e., top-down and bottom-up), can be used to inform the designer of the consequences (if any) that certain design choices will have on the thermal performance of the building. The simulations models, in turn, validate the defaults suggested by the rules, and help derive new rules for non-standard design cases.

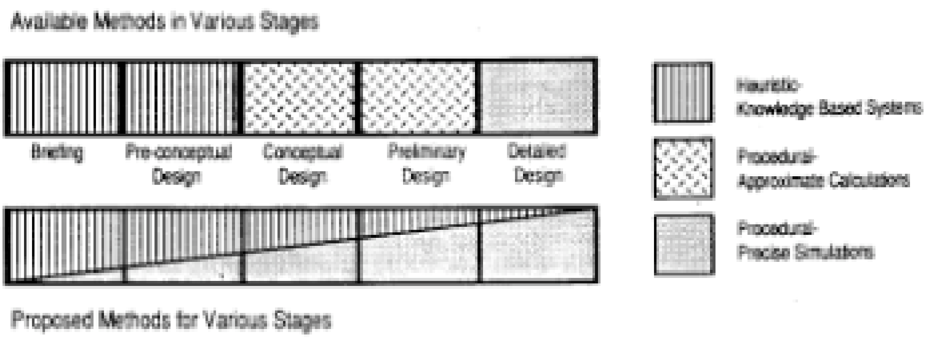

Figure 6. Schematic comparison of the use of available vs. proposed thermal properties evaluation tools in different design phases (from (Shaviv and Kalay 1991)).

Conclusion

In this paper we have discussed and established the multi-modal nature of evaluation, and the ensuing differences in software tools that support each evaluation modality. These differences stem not only from the different needs that each modality fulfills, but also from the kinds of knowledge it relies upon: The Multi-Criteria modality relies on broad-based knowledge and upon generalized methods for its application, whereas the Multi-Level modality relies upon deep knowledge and specialized methods.

Designers, however, engage both evaluation modalities alternately and in parallel throughout the design process. Furthermore, they shift within each modality from top-down to bottom-up methods, and within the same design phase from one design objective to another. Therefore, they ought to be able to use different design and evaluation methods interchangeably.

DESIGN PHASE	a.	b.	c.	d.	e.	f.
TOTAL FLOOR AREA	*		.	.		
VOLUME OF BUILDING		*	.	.		
NO.OF EXTERNAL WALLS			*	.		
AZIMUTH OF REFERENCE WALL			*	.		
AREA OF INTERNAL MASS				*	.	
HEAT CAPACITY OF INTERNAL MASS					*	
INITIAL TEMP. DISTRIBUTION		*				
HEAT GAINS: CONSTANT	*					
HEAT GAINS: SCHEDULE	*					
HEATER: SCHEDULE; TEMP.SET		*			.	
COOLER: SCHEDULE; TEMP.SET		*			.	
VENT: TYPE, SCHEDULE, ACH: DAY; NIGHT		*			.	
WALL NO 1						
NUMBER OF LAYERS				*	.	
MATERIALS(d,λ,ρ)				*	.	
* * *						
* * *						
WALL: AZIMUTH, INCLINATION			*	.		
AREA OF WALL			*	.		
WALL:ALBEDO,EMISSIVITY					*	
WALL:SC-SUMMER,WINTER				*	.	
NO OF WINDOWS AND SOLAR ELEMENTS			*	-		
WINDOW: AREA			*	.		
WINDOW: TYPE				*	.	
WINDOW: SC-SUMMER, WINTER		*			.	
U WINDOW: DAY; NIGHT				*	.	
U WINDOW: SCHEDULE		*			.	

Figure 7. Phase-specific classification of energy-related design parameters (from (Shaviv and Kalay 1991)).

The fundamental differences between the computational paradigms that support each evaluation modality make it unlikely, however, that a single evaluation tool could be constructed to support both modalities simultaneously. Such a tool would, by definition, have to include both broad and deep knowledge-bases, concerning all the relevant performance criteria at each phase of the design process. Such a mammoth knowledge-base could hardly be conceived as a monolithic entity, much less be programmed as one.

Rather, computational frameworks for availing different evaluation tools to the designer, and facilitating dynamic shifts between the modalities they support, are needed. Such frameworks will include a plethora of specific evaluation tools, many of which have already been developed. They will differ from the integrated models discussed earlier in that they will also include a knowledge-base which can select and invoke the right tool at the right time, triggered by both the design phase and the design objective under consideration. Such frameworks will resemble more closely the knowledge of the architect himself, rather than the knowledge of the specialists he relies upon, without sacrificing the specialists' particular knowledge. Hopefully, the discussion presented here will stimulate the research toward developing integrated computational frameworks of this kind.

References

Asimow, M. 1962. *Introduction to Design*. Englewood Cliffs, NJ: Prentice Hall.

Broadbent, G. and A Ward (eds.). 1969. *Design Methods in Architecture*. London: Lund Humphries.

Carrara, G., Y.E. Kalay, and G. Novembri. 1991. "A Computational Framework for Supporting Creative Architectural Design." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Cross, N. 1977. *The Automated Architect*. London: Pion Ltd.

Fenves, S.J., Flemming U., Henderson C., and M.L. Maher. 1991. "Performance Evaluation in an Integrated Software Environment for Building Design and Construction." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Flemming, U., R Coyne, T. Glavin, and M. Rychener. 1988. "A Generative Expert System for the Design of Building Layouts - Version 2." In J.S. Gero (ed.), *Artificial Intelligence in Engineering: Design*. New York: Elsevier.

Goldschmidt, G. 1991. "Criteria For Design Evaluation: A Process Oriented Paradigm," In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Gregory, S.A. (ed.). 1966. *The Design Method*. London: Butterworth.

Gross, M.D. and C. Zimring. 1991. "Predicting Wayfinding Behavior in Buildings - a Schema-Based approach." In Y,.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Hacfoort, E.J. and J. Veldhuisen. 1991. "A Building Design and Evaluation System." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Jockusch, P.R.A. 1991. "How Can We Achieve A Good Building?" In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Jog, B. 1991. "Evaluation of Design Using CAD Systems." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Johns, C. 1980. *Design Methods*. New York: John Wiley & Sons.

Johnson, R.E. 1991. "Design Inquiry and Resource Allocation." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Kalisperis, L.N., M. Steinman, and L.H. Summers. 1991. "A Human Thermal Comfort Response Design Model for Non-Orthogonal Surfaces." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Liggett, R.S., W.J. Mitchell, and M. Tan. 1991. "Multi-Level Analysis and Optimization of Design." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Manning, P. and S. Mattar. 1991. "Strategy and Tactics, Prediction and Evaluation: A Preliminary to Development of Expert Systems for Total Design of Entire Buildings." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

O'Neill, M. 1991. "A Neural Network Simulation as a Computer-Aided Design Tool for Evaluating Building Legibility." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Schmitt, G. 1987. "Expert Systems in Design Abstraction and Evaluation." In Y.E. Kalay (ed.), *Computability of Design*. New York: John Wiley & Sons.

Schmitt, G. 1988. "ARCHPLAN - An Architectural Planning Front End to Engineering Design Expert Systems." In M.D. Rychener (ed.), *Expert Systems for Engineering Design*. New York: Academic Press.

Shaviv, E. and Y.E. Kalay. 1991. "Combined Procedural and Heuristic Method to Energy Conscious Building Design and Evaluation." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.

Simon, H.A. 1969. *The Sciences of the Artificial*. Cambridge Mass.: MIT Press.

Wiezel, A. and R. Becker. 1991. "Integration of Performance Evaluation in Computer-Aided Design." In Y.E. Kalay (ed.), *Evaluating and Predicting Design Performance*. New York: John Wiley & Sons.