

COMPUTATIONAL BUILDING MODELS: THEME AND FOUR VARIATIONS

Ardeshir Mahdavi

Department of Building Physics and Human Ecology

Vienna University of Technology

A-1040, Vienna - Austria

ABSTRACT

Building design and operation processes can be supported by many kinds of models, from traditional architectural scale models to computer-generated virtual buildings. This paper presents four variations not so much "on", but rather "around" the theme of computational building models. As such, these variations address matters and ideas that are important for the range and effectiveness of model application towards supporting the design and operation of more habitable and sustainable built environments. Specifically, the variations deal with questions a propos: representational integration, performance-to-design mapping, design space exploration, and self-organizing building models.

0. THEME

The question is, if the reduced model isn't the quintessence of the artifact in the first place... In reduction, the totality of the object appears less intimidating.

(Claude Lévi-Strauss)

A word on models. Models are entities that represent other entities. While represented entities may be arbitrarily complex, models can be highly "reduced", i.e. they may focus only on a limited sub-set of the features of the represented entity while abstracting from other features. Despite – or perhaps because of – this reduction and abstraction, models can enable their users to effectively explore, document, understand, and predict certain properties and behavior of the modeled entity. Moreover, models not only facilitate communication amongst people, but also fulfill an "auto-communicative" function, supporting a kind of internal dialogue within one individual's mind. On occasions, architects' early sketches seem to function as a medium for such an internal dialogue. The effectiveness of a representational act depends not only on the intentionality that motivates its inception, but also on the consistency of the referential framework and the interpretative (or reconstructive) role of the user (Mahdavi 2003).

A road map. Building design and operation can be supported by many kinds of models, from traditional architectural scale models to computer-generated virtual buildings. This paper presents four variations not so much "on", but rather "around" the theme of computational building models. As such, these variations address matters and ideas that are particularly important for the range and effectiveness of model application towards supporting the design and operation of habitable and sustainable built environments:

- i) The first variation deals with the question of representational integration: To which extent can a building information schema accommodate and support multiple views and applications and their diverse informational requirements?
- ii) The second variation explores the potential for the reversal of the default mode of inference in building performance simulation (i.e. from design to performance): Can a computational system facilitate performance-to-design mapping?
- iii) The third variation carries the question of the second variation one step further: Is it possible to generate and effectively represent the entire (or a large chunk of the) corpus of principally possible designs together with their respective performance attributes so that they could be viewed and evaluated in the context of a "design-performance space"?
- iv) The fourth variation explores the possibility to apply computational building models beyond the building design phase and into the operation phase of buildings. It discusses self-organizing models and their role in model-based building operation methods and approaches.

The paper concludes with some thoughts on possible explanatory limits of computational models. Thereby, it specifically addresses the gap between the knowledge about the energetic attributes of exposure situations in the built environment and their subjectively perceived and assigned nature and meaning.

1. REPRESENTATIONAL INTEGRATION

The problems is that, to some extent, all of the sciences have their own "reality". And ... it is just not good enough for each scientific discipline to content itself with its own "reality"—even when this reality is actually irreconcilable with the reality of other sciences. To be decent scientists we must take one another's "realities" seriously enough to try to eliminate the contradictions.
(Jesper Hoffmeyer)

What is integration? In this contribution, the problem of representational integration is understood as follows. Building models may represent specific (disciplinary) views of the building. Moreover, they may represent information at different levels of resolution (abstraction). An integrated building model would thus imply a unified, structured, multi-resolutional, and multi-disciplinary building information repository. Occasionally, integrated building models have been thought of as "universal" (i.e. all-purpose) sources of information.

What is the use of integration? The diversity of views and levels of abstraction in building information can be a source of problems in communication. Professionals must obtain, interpret, transform, and exchange information. These processes typically involve redundancies, errors, and inefficiencies (CSI 1999). Representational standards could presumably add to the efficiency and fidelity of the information exchange processes. Moreover, so it has been argued, a combination of representational standards and computational mapping techniques would enable professionals to "automatically" derive from a central building model the specific disciplinary information they need at a properly scaled level of resolution. Such automated mapping would be specifically desirable from the building performance simulation perspective. Preparation of a building model for performance simulation purposes out of the conventional building information media is a time-intensive and error-prone process and has been thought to be one of the main hindrances against pervasive use of building performance simulation tools in the building design process.

Why is there a problem? The problem, as stated above, is some three decades old. Progress has been made, but conclusive solutions have not been found. The reasons are multi-fold. To better expose the problem, we will briefly discuss two general approaches toward its solutions.

The first, rather top-down approach relies on establishment of common representational standards. "Universal", generally agreed-upon descriptions of building information would presumably result in unambiguous and efficient communication. While

sounding logically sound, this program faces a number of challenges: *i)* Conceptually, there is no reason to believe there exist necessary and "natural" (quasi platonic) representations for built entities (Mahdavi 2000). Rather, one deals in this area with fluid conventions that can differ from discipline to discipline, culture to culture, and period to period; *ii)* Practically, the standardization effort in the building domain must address a large and diverse professional context. Representatives of multiple disciplines must be brought to table. Accordingly, the administrative and procedural overhead (time and cost) is high and classical problems of "design by committee" emerge. Given the long developmental cycles involved, standards may become technically obsolete by the time they are released (Behrman 2002). Moreover, without legislative backing or the promise of likely short-term investment return, completed standards need not be adapted by the relevant industries, developers, and potential users.

A second approach (or group of approaches) to the integration problem could be loosely characterized as bottom-up or perhaps "pragmatist". Thereby, the idea of a unified building model is either fully abandoned or substantially modified (down-sized). Once the heterogeneity of the nature, sources, and formats of design information is assumed and accepted, integration efforts gain a more local, strategic, and pragmatist flavor. Given diversity of data representations, data exchange processes are supported on a case by case (application by application) basis.

To illustrate this point, consider the "classical" problem of input preparation for performance simulation applications. This input must include, amongst others, the building geometry information. In absence of universal representational standards for building geometry, one bottom-up approach suggests to work with conventional design documents, but process those via AI-based techniques (such as automated computational geometry interpretation) toward a format suitable for simulation applications.

This second group of approaches faces challenges of its own: *i)* Without standardized shared building models, the number of mapping routines for data exchange can rapidly increase as the number of applications increase. The overhead involved in the conception, implementation, and maintenance of such mappings may become prohibitive; *ii)* Depending on the nature of required data post-processing, AI-techniques may be brittle and not always reliable. To return to the example of geometry interpretation, it is not a trivial AI task to enable a computational application to "read" traditional design documents. There have been a number of efforts to develop methods and routines for automatic geometric

recognition of drawings (Negroponte 1975, Do 1996, Pohl and Reps 1988, Pelletret and Keilholz 1999). However, robust interpretative routines for the geometric-topological interpretation of complex architectural drawings have been difficult to develop. To match human agents' capacity to make sense of an underdetermined set of partial representations (such as plans and sections of a building), the interpretative computational routine must not only recognize architectural entities, but also creatively patch over possible inconsistencies (cp. Figure 1 for a humorous take on this question). Moreover, a drawing is as such frequently ambiguous. Rival valid architectural interpretations of drawn entities are possible. To choose the appropriate option amongst multiple possible interpretations of a drawing represents yet another formidable AI challenge.

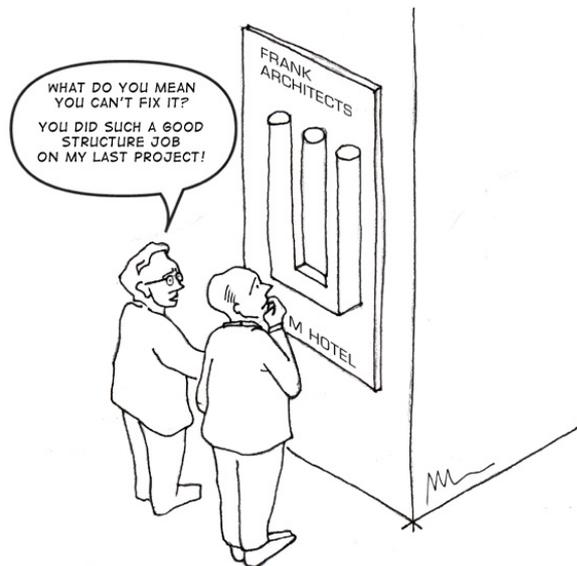


Figure 1. The spatial (3-D) interpretation of freely generated 2-D drawings can be a bit difficult (cartoon by the author)

The middle-path. Somewhere between the "platonian" and "pragmatist" approaches, the SEMPER effort (Mahdavi 1999, Mahdavi et al. 1999a) pursued a "middle path" (Mahdavi 2000). We showed, in principle, that design information from a properly structured shared building model can be seamlessly mapped into the domain models of a number of technical building analysis applications for energy simulation, thermal comfort prediction, building HVAC (heating, ventilating, and air-conditioning), air-flow, lighting model, room acoustics model, and life-cycle assessment. Two suppositions were particularly important for the SEMPER effort. First, it was believed that at some basic level some shared

notation of the constitutive building entities and their topological interrelations was *conditio sine qua non*. Second, it was assumed this notation could be hardly conceived as a necessity to be derived via meditations on the nature of the building as such. Rather, it had to evolve and had to be tested in the context of requirements of the "down-the-line" manipulators of the entities encapsulated by such a notation system. This ensuing work resulted in SEMPER's perhaps most essential features, namely a space-based shared building model and a "homology-based" mapping technique (Mahdavi et al 2001, 1997a, Mahdavi and Wong 1998).

The notion of space was an important part of the way SEMPER's multiple applications related to a shared scheme of constitutive building components. The space-based representation in SEMPER provided the necessary condition to cater for the informational needs of a number of analytical applications from a shared building model. This shared object model is a hierarchically structured template to capture the features and elements of a building and their properties, to the extent required by the simulation applications considered in SEMPER. We concluded that a shared building model could be arrived at for a number of technical analysis applications and for performance inquiries of a certain range of informational resolution. In other words, SEMPER's integration functionality was never thought as universal. Incidentally, the shared model in itself did not contain the entire building information. Rather, it contained an abstract representation of constitutive building elements, with pointers to (addresses for) the detailed information on such elements in separate data repositories. Moreover, while this shared object model enabled the SEMPER applications to retrieve the necessary building geometry, material, and context information, it was not sufficient on its own for a building performance simulation application to function. For each disciplinary domain, the simulation application's representation, or the "Domain Object Model", was generated upon filtration and modification of information in the shared model according to the specific view of the building in that domain. Furthermore, domain specific entities (e.g. finite control volumes in numeric heat and mass transfer computation) had to be added to what is extracted from the shared model.

While domain representations in SEMPER used different internal spatial representations for their computations (e.g. a thermal zone, an airflow control volume, or an acoustical space), they were nonetheless homologous (configurationally isomorphic) to the shared building model. This homology was exploited to a certain extent for automated and non-ambiguous mapping operations from the shared building model to the domain models

of the applications incorporated in the SEMPER environment (Mahdavi and Wong 1998, Mahdavi and Mathew 1995). It was not claimed, however, that this mapping method would work for all domains and independent of the informational resolution of the pertinent inquiries.

Despite its limitations, the SEMPER project did demonstrate that, for a certain set of applications, a certain set of queries, and a certain level of building information resolution, a well-balanced representational labor division between a reasonably detailed shared building model and a number of behavioral domain models (for building performance simulation) is possible, and that the latter can autonomously infer their informational requirements from the former via mapping operations. A likely question is, of course, if and to which extent this integrative framework could be expanded to accommodate other applications and other levels of building information resolution. Given what we have learned about the problems and perils of integration, perhaps it would be best to approach this question on an empirical, case-by-case basis.

2. PERFORMANCE-TO-DESIGN MAPPING

Form follows function.
(Louis Sullivan)

The direction of inference. The main role of behavioral models of buildings in the design process is generally believed to be the prediction and evaluation of the performance implications of changes in design. This implies a direction of inference from design to performance. It could be argued, however, that what designers usually do (or expect) would actually imply (or require) the opposite inference direction (Augenbroe and Winkelmann 1991). After all, shouldn't a design evolve in response to a set of (broadly understood) performance criteria? And if performance-to-design mapping is necessary and desirable, shouldn't there be more effective computational support for it?

In practice, computational environments for the support of performance-to-design inferences are even less widespread than the ordinary simulation applications for design-to-performance mapping. In this context, two problems are especially important, namely the ambiguous nature of performance-to-design inferences, and the iterative nature of the convergence processes in design.

The ambiguity problem. Given a clearly specified building in a clearly defined context, and given a reliable and robust simulation tool, one can consider design-to-performance mapping to be a non-

ambiguous operation. In other words, repeated simulation runs for the same building in the same context are expected to yield the same performance results. The same, however, does not hold for a reverse mapping operation. Since very different designs can yield identical performance indicator values, it follows that the performance-to-design inferences must be ambiguous by nature (a fortunate circumstance, from the design freedom and creativity point of view). Thus, computational engines for performance-to-design mapping must work around the ambiguity problem (Mahdavi 1993).

Optimization, generation, and the iterative nature of design. As such, "classical" optimization applications in design may be considered to represent clear instances of performance-to-design mapping: Based on an explicit definition of applicable objective functions (typically specified in terms of performance criteria), desirable (optimal) values for one or more design variables can be derived using appropriate mathematical algorithms (Radford and Gero 1988).

A typical "one-shot" optimization application, however, does not seem to be amenable to certain important features of the design process. First, identification, explication, and operationalization of pertinent design variables are not trivial matters. Interestingly, the (as such positive) trend away from prescriptive standards toward performance-based standards reduces the need for (and the significance of) formal definitions of design variables. Second, it has been argued that designs evolve gradually in an iterative manner (involving multiple rather unpredictable stages with numerous instances of problem-restructuring along the way). It follows that such inherently iterative processes could be hardly supported (let alone substituted) by "one-shot" optimization applications.

Another general class of performance-to-design inference applications may be recognized in so-called generative systems. Floor plan layout generators represent a typical example for such systems. Here, a set of constraints (e.g. adjacency requirements concerning rooms in a floor) are the starting point for the generation of a (potentially large) number of candidate solutions. Underlying computational approaches may range from rule and constraint-based methods (Flemming et al. 1992) to genetic algorithms (Elezkurtaj and Franck 2001). While "chip-packing" and lay-out problems represent typical instances of generative systems, such systems may also be realized in domains directly relevant to building performance. For instance, the distribution of terminal units for heating and cooling of buildings and the associated duct layout may be derived via specialized generative applications (Mahdavi et al. 2001a).

As with the optimization methods, a generative system could be hardly qualified as a design support tool if its use-scenario would imply a "design-machine" for the automated production of quasi perfect designs. Note that this skepticism is not about the designers' fear of replacement by some sort of "intelligent design robots". Broadly speaking, there are good reasons to believe that designing is not about generation of unique and perfect artifacts based on unique (and explicit) requirements. In reality, design requirements are seldom expressed explicitly, nor are they homogeneous. Next to fairly concrete minimum standard requirements (including general performance criteria), there are many other aspects of design and design decision making that are less tangible. As such, designs do not result from performance requirements in the way that effects are thought to result from causes.

Bi-direction inference environments. Moving away from the early rather rigid instances of optimization applications and generative systems, recent implementations seem to have adopted a more responsive stance to the specific traits of the design process. This does not indicate a fundamental change in the basic computational techniques underlying optimization approaches and generative systems, but rather an intention to incorporate such techniques in the overall context of a design support environment (Mahdavi et al. 1997b).

At a general level, there is less interest in postulating and finding "optimal" design solutions in the strict sense of the word. For instance, instead of trying to establish a framework for "global" optimization of designs, performance-based design support environments can embody optimization routines such that they could be used in a flexible, dynamic, and iterative manner (Mahdavi and Mahattanatawe 2003). Thus, partial and local optimization functionalities could be deployed in tandem with other means and methods of design decision support, whereby the entire step-wise convergence process would be guided by the considerations and concerns of the designer (or the design team). It is of course true that optimization procedures are typically guided by constraints. However, in flexible implementations, such constraints need not be cast in concrete over the entire length of a design session. Rather, they can be redefined or modified dynamically, depending on the specific features and contingencies of a design exploration trajectory.

Likewise, in use scenarios involving generative systems, designs need not be produced exclusively by the generative system, but can be manipulated on-the-fly by the designer (Harada et al. 1995, Elezkurtaj and Franck 2001), leading to a kind of creative ping-pong between human designer and the generative

algorithm. Moreover, in a flexible generative system, the selection criteria for the automated refinement of successive generations of solutions may be freely revised from generation to generation.

The case of GESTALT. Perhaps the promise and certain inherent (probably systemic) problems of performance-to-design mapping operations could be further illustrated using the example of an actual implementation effort, namely the GESTALT project (Mahdavi and Berberidou 1994).

The peculiarity of this prototype lied in the special way in which the performance to design mapping inference engine was realized and perceived as such by the user. Specifically, the computational core of the design-to-performance mapping shell was actually a typical simulation routine (i.e. a design-to-performance mapping algorithm) which was recursively applied. In a nutshell, the system worked as follows:

- i) the user entered the initial design into the system;
- ii) the system computed and displayed the performance of this initial design via one or more performance indicators (e.g., daylight factor, glare index);
- iii) the user indicated a desire to increase or decrease the value of a performance indicator;
- iv) internally (transparent to the user), the system generated alternative values of a number of design variables, computed the performance of the resulting design alternatives, selected and displayed the one with the best performance improvement in the desired direction.

To function properly, a system such as this must possess a number of critical features:

- a) Performance indicators must be aggregated over time and space. For example, indoor daylight levels change dynamically depending on the external condition. Furthermore, at any point in time, daylight levels are different in different points in a space.
- b) Design variables must be explicitly defined in the system in manner that allows for feasible modifications of their values (physical dimensions or proportions of spaces or apertures represent geometric examples, reflection coefficients represent semantic examples of design variables). Specifically, the overall numeric range and the size of discrete steps of such variables need to be defined.

- c) If improvement in the values of more than one performance indicator is desired, an aggregate performance indicator would be necessary. This, in turn, requires the definition of priorities (and associated weights for aggregation).
- d) If there are more than one design variables that could "react" to a desired performance improvement, then the system must know which ones, in which order, and to which extent could be modified. This implies the need for the explication of preferences for certain ranges of values for design variables.
- e) Potentially complex mechanisms are necessary to ensure that alternative designs generated by the system (via changes in interrelated design variables) possess integrity. Designs are typically hierarchical systems of many entities and features. Maintaining integrity, while propagating changes in such complex systems is a difficult problem. Proposed solutions require the systematic definition and management of nested constraints (explicated topological and numeric interdependencies) in the behavior of design variables and represent thus a formidable implementation overhead (Mahdavi and Suter 1997, Suter and Mahdavi 1999).
- f) If new designs are generated exclusively via incremental changes in design variable values (the "greedy" approach), it is quite likely that the exploration would lead to local performance minima or maxima. To reduce this risk, appropriate methods (e.g. random re-starts) must be implemented in the system.
- g) To be attractive from the usability point of view, a real-time system behavior is desirable. This would require however, either extremely powerful computing resources for performance simulation, or alternatives to detailed numeric simulation (e.g. neural network copies of simulation applications).

Despite all these problems and limitations, the GESTALT project provided a prototypical instance of an environment for the vivid, dynamic, and bi-directional exploration of the interrelationships between a number of performance indicators and design variables. Even though GESTALT's unusual performance-to-design mapping functionality was perhaps not scalable to the level of topologically and hierarchically complex designs, it did turn out to be an effective and attractive feature in educational settings. Using GESTALT for daylight evaluation, even users with little familiarity with building performance issues could rapidly develop a "feel" for the basic performance implications of design actions.

3. DESIGN SPACE EXPLORATION

Dimension is a geometric way of referring to a variable.

Time is a nonspatial variable, so it provides a fourth dimension, but the same goes for temperature, wind-speed, or the number of termites in Tangentia. ...In fact, any complex system is multidimensional.

(Vikki's diary – Ian Stewart)

Alice in the design-performance land. Performance-based design may be supported by exploring the realm of possibilities in the "design-performance space". Performance simulation applications can support the generation of this realm of possibilities, namely a virtual space defined by multiple design and performance dimensions. In such a space, each design dimension accommodates the range of possible values of a design variable and each performance dimension accommodates the range of the values of a specific performance indicator. Once a design-performance space is constructed around an initial design, it can be visualized and used by the designer to explore the relationship between a specific constellation of design variables and the resulting performance attributes (Mahdavi and Gurtekin 2002, 2001).

It is true, of course, that in the current building delivery process extended excursions in the design-performance-space are typically not paid for and thus rarely undertaken. But assuming proper boundary condition (availability of time and computational resources), such excursions could arguably contribute to lively and creative dealings with performance issues in design. Two of the many challenges toward implementation of environments for design-performance space explorations are the identification of pertinent and preferably continuous design and performance variables and the provision of computationally efficient performance modeling engines for real-time generation of the design-performance space around an initial design.

Defining design variables. Building design variables capture either geometric or non-geometric (semantic) information. Most performance-relevant semantic design variables can be defined in terms of numeric values (thermal conductivity, visible transmittance, etc.). Geometric design information is more difficult to express in terms of scalar values. Examples of some common building geometry indicators are plan aspect ratio, ratio of a space's height to its depth, and ratio of glazing area to the facade (or floor) area. Given the inherent complexity of building shapes, better aggregate descriptors of building geometry are needed. Ideally, such indicators must not only numerically represent building geometry, but should also be perceptually relevant, i.e. conform to the designers' perception of the buildings' geometry. Only then would the performance implications of

parametric changes in the value of a geometry indicator provide the designer with a sense of desirable concrete geometric manipulations of a design.

There have been some attempts to describe the compactness of building shapes in terms of the relation between a building's volume and total surface area (Mahdavi et al. 1996, Markus and Morris 1980). For example, the "Relative Compactness" (RC) of a shape (Mahdavi and Gurtekin 2002, 2001) is derived by comparing its volume (V) to surface area (A) ratio to that of the most compact shape with the same volume. The most compact shape in geometry is the sphere. However, it is perhaps not the ideal reference, as most buildings have orthogonal polyhedral shapes. Using the cube (the most compact polyhedron) as the reference shape, we obtain:

$$RC = 6 V^{2/3} A^{-1}$$

In order to explore the degree to which RC correlates with the subjective assessments of the compactness of building shapes, we performed an empirical pilot study (Mahdavi and Gurtekin 2001). A sample of 14 building shapes was established with RC values ranging from 0.49 and to 0.98. A group of 48 senior architecture students participated in the subjective evaluation of the compactness of the shapes. They were asked to evaluate each shape (presented in a random sequence in terms of axonometric projections) separately based on a semantic differential, whereby they were to assign a numeric value (from 1 to 7) to each shape according to its (subjectively perceived) level of compactness. Figure 1 shows for these shapes the quite remarkable relationship between the subjective evaluation marks (averaged over all test participants) and the corresponding numeric values of the Relative Compactness.

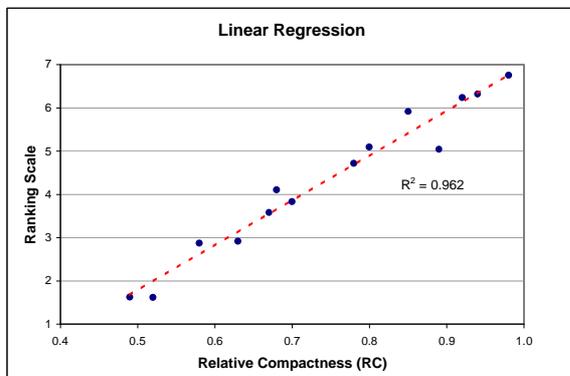


Figure 2. Relative compactness of 14 shapes versus their subjective ranking

Real-time performance modeling. To generate the design-performance space, design variables must be parametrically changed along all design dimensions and the resulting performance attributes must be expressed in terms of corresponding numeric ranges along the performance dimensions. Simple examples for thermal performance indicators are annual and peak heating and cooling loads and thermal comfort indices such as PMV and PPD (Fanger 1970).

In a prototypical implementation of an environment for the exploration of the design-performance space, we used neural network copies of simulation programs for performance modeling. Neural networks offer two advantages. First, once generated, they can provide simulation results very fast. This is a decisive point, if a real-time exploration of the design-performance space is to be supported. Second, as opposed to detailed simulation routines, neural networks allow to use scalarized input variables for geometry. Since the prototype environment focused on the thermal performance of typical residential building designs, a sample of such buildings was selected and used for extensive parametric analysis. The data thus obtained was then used to for neural network training, resulting in a mathematical model that represents the relationship between design variables and the performance attributes. Using this model, the entire design-performance space can be constructed. Thus, a design can be analyzed by viewing it among other possible alternatives. The exploration environment helps the user to identify possible solutions that will yield better performing designs. Given the ranges of the design variable values, 2D, 3D and 4D plots of the design-performance associations can be generated and updated on the fly (see, for example, Figure 3).

Frontiers. The design-performance space formalization is conceptually and educationally attractive. But it has practical limitations. First, the production of generally applicable neural network copies of simulation programs can become a highly tedious undertaking. Besides from the issue of non-continuous design variables, generation of required samples and the training process can represent a formidable challenge in terms of overhead. Second, the identification of neat numeric design variables (particularly in case of geometry) is not trivial either. For instance, an indicator such as RC can only capture just one of the many aspects of a building's geometry. Perhaps there is a limit as to how far a design variable can be simultaneously abstract (e.g. numeric, continuous) and concrete (relevant for the perceptual evaluation and intuitive manipulation). Third, visualization tools can support in principle the exploration of higher-dimensional "information-scapes", but only so much. They can aid our

imagination, cognition, and memory, but they cannot transcend their limitations.

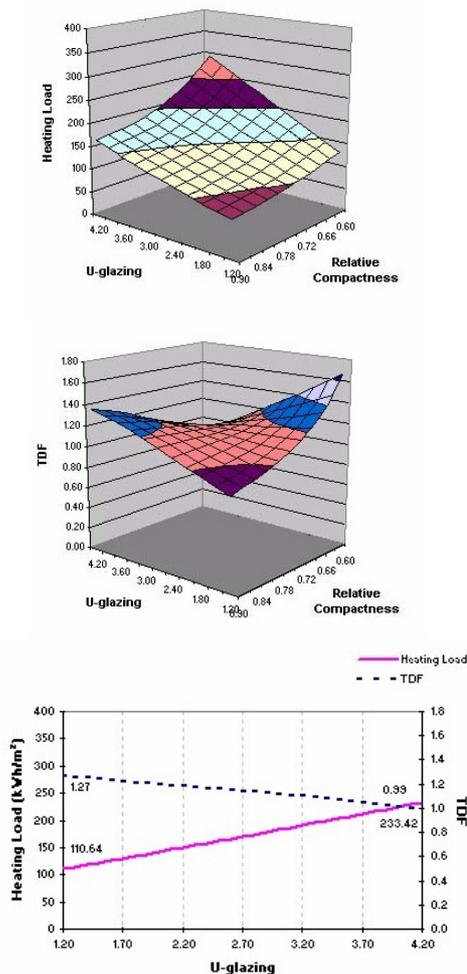


Figure 3. Data visualization allows exploring the relationships between multiple variables at once

4. SELF-ORGANIZING BUILDING MODELS FOR SELF-AWARE BUILDINGS

Does a thermostat act, or only 'react'?
(Ralph Ellis)

Evolution of models. Computational building models have evolved considerably in the past decades. Without attempting to adhere to a strict thematic or chronological order, certain milestones may be loosely identified:

i) Simple geometric models: Most commercial computer-aided drafting systems relied (some still rely) on representations, which are mainly built

upon geometric primitives (points, lines, polygons, etc.). Such elementary geometric representations are typically devoid of semantic attributes and explicitly embodied topological information.

ii) Component-based models: Emerged in the context of computer-aided architectural drafting systems, such models involve explicit references to "architectural" entities. This is achieved, in that elementary geometric features are bundled to graphically represent standard architectural elements and components such as walls, roofs, windows, doors, stairs, etc.

iii) Semantically enriched models: These models include, beyond simple labels for geometrically represented architectural entities, associated non-geometric attributes (i.e. semantic information such as material properties).

iv) Topologically enhanced models: These models embody information on spatial entity relationships amongst the elements of the geometric object model. For instance, the model "knows" not only of the existence of multiple rooms in a building representation, but also on their adjacency relationships.

v) Integrated models: As already discussed in this paper at some length, these models are meant to embody building information in a comprehensive and multi-view (multi-disciplinary) fashion. Such models can emerge when information from multiple disciplinary sub-models are brought together within the framework of a shared building representation (Mahdavi 1999, Mahdavi et al. 2002, 1999a).

vi) Behavioral models: Representational systems for buildings are often static, i.e. they are limited to a structured – often hierarchical – collection of descriptions for a set of constitutive building components. It is possible, however, to enrich a primarily static building model with dynamic behavioral features via rules and simulations. Performance areas such as energy, acoustics, illumination, and structures are amongst domains relevant for the application of behavioral simulation.

A cursory look at these representational instances in the context of the building delivery process reveals that their evolution has been predominantly guided by the concerns and requirements of the design phase. A shift of focus to the operational phase of the building delivery process could arguably open new vistas in the building modeling realm.

Toward this end, we introduce the concept of a self-organizing building model as one with the built-in potency of real-time and predominantly independent evolution and adaptation with regard to changes in building context, structure, systems, status, processes, and occupancy. Note that the term "self-organizing" is often used to elaborate on the distinction between "the living, agent-like, active, and purposeful on the one hand, and the non-living, passive, merely reactive or mechanical on the other" (Ellis 2002). Accordingly, a self-organizing process is defined "as one whose organization creates a strong tendency to maintain itself across various alternative causal mechanisms at the level of the components making up the system." (Ellis 2002). In the context of this contribution, "self-organizing" is used in a "weaker" (i.e., "as-if") sense and is not meant to imply ontological identity with salient features of biological systems. Rather, it is intended to denote the implementation of certain self-regulating and self-adapting functionalities in representations that are geared toward the operation phase of the building life-cycle.

Self-aware buildings. Aside from general sources in cybernetics, information theory, and dynamic system theory (Bertalanffy 1962, Brillouin 1956), the concept of self-organizing building models was inspired and informed by the role of behavioral models in the so-called "self-aware" buildings (Mahdavi 2001a, Mahdavi et al. 2001b). Behavioral models (typically realized as simulation applications) have been used mostly to predict and evaluate the performance of designs. But since they allow to capture the dynamic interactions between a built entity and its surrounding context, they can be used also to predict the implications of changes within or around existing buildings. As such, they make it possible to perform virtual experiments with building systems. Hence, the central impetus for envisioning self-organizing models was the emergence of proactive building control strategies that rely on the predictive capacity of embedded knowledge sources (particularly simulation applications). Behavioral models could be integrated within the building automation system of a "self-aware" building, and thus facilitate the virtual exploration of the control state space of the building's environmental systems (e.g. for heating, cooling, ventilation). A self-aware building possesses thus an internal (dynamic) representation of its own systems and can use this representation toward self-regulatory determination of its status. As such, the functionality of a self-aware building requires the realization of a self-organizing building model and model-based control strategies.

Model-based building control. The control approach in a self-aware building requires:

- i) A modular, distributed, flexible, and scalable data monitoring and processing infrastructure to collect information (on building components and elements and their current status and properties, micro-climate, indoor environmental conditions, status of technical building support systems such as heating, cooling, lighting, status of building components such as windows, partitions, furniture) and to actuate environmental control and modification devices and systems.
- ii) The capability to generate and maintain representations and behavioral models of buildings and to use those – among other things – for building systems control purposes. Such models address buildings' actual (current) operational status, buildings' past behavior/status (building memory system), and buildings' future operation (generation and predictive evaluation of alternative building control schemes and sequences).

The multitude of controllers in a complex building controls scheme must be coupled appropriately to facilitate an efficient and user-responsive building operation regime. The nodes in the network of such couplings represent points of information processing and decision making. An important challenge for any building control methodology is to find effective methods of knowledge encapsulation and decision making in such nodes. There are various ways of doing this (Mahdavi 2001a). The simulation-based control method is discussed below. This method is particularly relevant to the concept and realization of self-aware buildings.

Modern buildings allow, in theory, for multiple ways to achieve desired environmental conditions. For example, to provide a certain illuminance level in an office, daylight, electrical light, or a combination thereof can be used. The choice of the system(s) and the associated control strategies represent – from a theoretical point of view – a non-trivial problem with implications for the objective function of the control strategy (e.g., desirable environmental conditions for the inhabitants, energy and cost-effectiveness of the operation, minimization of environmental impact). The reason for this non-triviality is that there is no deterministic procedure for deriving a necessary (unique) state of the building's control systems from a given set of objective functions.

Simulation-based control can potentially provide a remedy for this problem (Mahdavi 2001b, 1997, Mahdavi et al. 2000, 1999b). Instead of a direct mapping attempt from the desirable value of an objective function to a control systems state, the simulation-based control adopts an "if-then" query approach. Consider, as a simple example, the lighting

control problem in a space, involving both a moveable louver (for shading and light re-direction) and electric lighting (luminaries). In the simulation-based control scenario, first the control state must be parameterized. For example, for louvers, discrete positions may be defined. Likewise, the light output of the luminaires in the space may be parameterized in terms of discrete dimming states. This results in a "control state space". Subsequently, multiple simulations are performed to map this control state space to a corresponding building performance space. The simulation results are then ordered in a matrix, which is used to rank and select the most desirable control scenario based on the applicable objective functions.

There are two main representational requirements for a self-aware building that makes use of a simulation-based control strategy. First, the underlying model must coherently incorporate both a "traditional" building product model (with its rather static view of the building) and an inherently dynamic building systems control model. Second, to meaningfully and efficiently support real-time building operation processes (e.g. via simulation-based control methods), the model must detect and consider dynamic changes in context (micro-climate), building components, building systems, indoor environment, and occupancy patterns fairly autonomously, i.e. without (or with a minimum of) user intervention. These twofold model attributes constitute a self-organizing model for self-aware buildings.

Self-organizing models for self-aware buildings. It was already argued that a building model, if it is to be useful for the self-aware building functionality, must unite both static and dynamic features of and process in the building. In contrast to the abundant efforts to develop building product models ("static" representational schemes for constitutive building components), there is a lack of representational systems that specifically address the building operation phase and its "dynamic" actors and processes. A coherent representational scheme for building controls is *conditio sine qua non* for the realization of the self-aware building concept. A terminology to capture the essential elements and concepts pertaining to building control systems and processes must contain definitions of controllers, control objectives, control devices and systems, actuators, controlled entities and zones, sensors, control actions, and control loops (Mahdavi 2001a). A typical control process involves a controller, a control device (or systems), and a controlled entity. In buildings, controllers may be realized at different levels of control systems hierarchy (within different control loops). They may regulate the behavior of a single valve, or they may act as the executive control agent for a whole building. Control loops may be

nested into larger organizations (control loop hierarchies). Controllers typically use trade-off methods and rules in case of conflict amongst multiple objectives. Moreover, they may have access to external information such as weather conditions, utility prices, and other sources of information and knowledge (sets of rules, behavioral models, etc.).

As such, the complexity of building systems control could be substantially reduced, if distinct processes could be assigned to distinct (and autonomous) control loops. In practice, however, controllers for various systems and components are often interdependent. A controller may need the information from another controller in order to devise and execute control decisions. For example, the building lighting system may need information on the buildings thermal status (e.g. heating versus cooling mode) in order to identify the most desirable combination of natural and electrical lighting options. Moreover, two different controllers may affect the same control variable of the same impact zone. For example, the operation of the window and the operation of the heating system can both affect the temperature in a room. In such cases, controllers of individual systems cannot identify the preferable course of action. Instead, they must rely on a higher-level controller instance (a meta-controller, as it were), which can process information from both systems toward a properly integrated control response.

Once a building model is available with instances for building context, structure, systems, status, processes, and occupancy, it can be used to support the real-time building operation (building systems control, facility management, etc.). However, given the complexity of such a model, it seems clear that it needs to be self-organizing, i.e. it must maintain and update itself fairly autonomously.

Depending on the type and the nature of the entity, system, or process to be monitored, various sensing technologies can be applied to continuously update the status of the building model:

- i) Information about critical attributes of external micro-climate (e.g. outdoor air temperature, relative humidity, wind speed and direction, global and diffuse irradiance and illuminance) can be gained via a number of already existing sensor technologies (Mahdavi 1999c). A compact and well-equipped weather station is to be regarded as a requisite for every self-aware building.
- ii) The success of indoor environmental control strategies can be measured only when actual values of target performance variables are monitored and evaluated. Also in this case there exists a multitude of sensor-based technologies to

capture factors such as indoor air temperature, mean radiant temperature, relative humidity, air movement, CO₂ concentration, and illuminance. Further advances in this area are desirable, particularly in view of more cost-effective solutions for embodied high-resolution data monitoring and processing infrastructures.

- iii) Knowledge of the presence and activities of building occupants is important for the proper functionality of building operation systems. Motion detection technologies (based on ultrasound or infrared sensing) as well as machine vision (generation of explicit geometric and semantic models of an environment based on image sequences) provide possibilities for continuous occupancy monitoring.
- iv) The status of moveable building control components (windows, doors, openings, shading devices, etc.) and systems (e.g. actuators of the building's environmental systems for heating, cooling, ventilation, and lighting) can be monitored based on different techniques (contact sensing, position sensing, machine vision) and used to update the central building model.
- v) Certain semantic properties (such as light reflection or transmission) of building elements can change over time. Such changes may be dynamically monitored and reflected in the building model via appropriate (e.g. optical) sensors.
- vi) Changes in the location and orientation of building components such as partitions and furniture (due, for example, to building renovation or layout reconfiguration) may be monitored via component sensors that could rely on wireless ultrasound location detection or utilize radio frequency identification (RFID) technology (Finkenzeller 2002). Gaps in the scanning resolution and placement of such sensors could be compensated in part based on, geometric reasoning approaches (possibly enhanced through artificial intelligence methods). Moreover, methods and routines for the recognition of the geometric (and semantic) features of complex built environments can be applied toward automated generation and continuous updating of as-is building models (Broz et al. 1999, Eggert et al. 1998, Faugeras et al. 1998).

Outlook. Some preliminary instances of self-organizing models for self-aware buildings have been prototypically realized. For instance, previous studies demonstrated that an integrated daylight and electrical lighting control system may be realized using a simulation-based approach (Mahdavi 2001a). In this case, information on outdoor conditions was

collected real-time through an external daylight monitoring station and was used to establish (and continuously update) the contextual component of the building representation (sky model). Additionally, the building model was continuously updated in view of the position of the building enclosure's light redirection louvers. An internal illuminance sensor was used to dynamically calibrate the predictions of the control system's underlying light simulation application. Moreover, machine learning methods were applied to generate fast-response calibrated neural network copies of the lighting simulation engine so that a larger portion of the control state space could be explored toward identification of preferable control options (Chang and Mahdavi 2002).

These implementations provided a proof of concept for the principal feasibility of self-organizing models as the internal representational core of self-aware buildings. However, as of now, we cannot make theoretically or empirically founded statements about the scalability and robustness of such models in the context of spatially more complex buildings and technically more demanding systems.

5. CODA

"Nature always retains behind her something problematic which it is impossible to fathom with our inadequate human faculties." (Goethe)

Simulation and phenomenal experience. We treated simulation as a form of dynamic behavioral representation. Performance simulation provides clues regarding the behavior of a building. Such clues typically consist of a bunch of numbers. The numbers are then evaluated, meaning that they are compared with some reference numbers considered appropriate or desirable. So far, so good. A lingering question remains, though: Can representational simulations assist evaluative processes in cases where phenomenal (subjective) issues are involved? Let us reiterate this question: If the process of occupancy evaluation of a space or a building is influenced by phenomenal experiences, which are thought to be essentially non-representational, are the essentially representational simulations relevant and applicable at all?

Circumvention of abstraction? To avoid long-winded philosophical discussions, we shall approach the question with a number of simple – and hopefully coherent – propositions:

- i) Many instances of performance simulation results such as predicted annual energy demand (and associated fuel consumption and energy cost) are irrelevant phenomenally.

- ii) Certain simulation results (such as illuminance level or CO₂ concentration levels in space) do not have direct phenomenal correlates, but they may be linked to other indicators that do.
- iii) Simulation results pertaining to indicators such as luminance, air temperature, sound pressure level, and reverberation time are phenomenally relevant. Their evaluative functionality is based on empirically-based correlations that link the indicator values with (the statistically compressed version of) people's report on their phenomenal experiences (e.g. thermal and acoustical sensations).
- iv) Simulation results could presumably circumvent the representational mediation of abstract formal (typically numeric) indicators, in that they offer to the sensory channels a virtual version of the simulated entity, which is, ideally, indistinguishable from the entity itself.

The last point above requires some qualification and explanation. Simulation applications have indeed made remarkable progress in the production of virtual environments. For instance, it has been argued that, since computer-generated images are now factually indistinguishable from photographic images, they could complement (if not substitute) traditional (mostly numeric) methods in the evaluation of the visual quality of the architectural spaces. Likewise, computational "auralization" tools allow for the placement of recorded acoustical events in virtual spaces and thus could presumably facilitate the immediate evaluation of room acoustics.

Image and reality. The postulated evaluative equivalency of virtual and actual spaces in view of their phenomenal implications is perhaps best approached empirically. For instance, a recent study (Mahdavi and Eissa 2002) attempted to empirically establish if and to what extent subjective lighting evaluation of architectural spaces can be reproduced using computationally rendered images of such spaces. A metric (7-point semantic differential) was used to capture certain subjective light quality dimensions. Five actual lighting situations were selected involving different spaces and lighting schemes. These situations were evaluated by a first group of test participants using the subjective lighting metric. High-quality renderings of the above-mentioned situations were generated using an advanced visualization tool. The rendered versions of the lighting situations were evaluated by a second group of test participants. Subjective lighting assessments of the real spaces were compared with those of the computational visualizations to empirically determine the degree to which such visualizations can represent real spaces toward

subjective lighting evaluation of architectural designs. To establish an overall understanding of the degree of agreement between the results of the two tests, a regression analysis was performed (cp. figure 4). The correlation (Multiple r) was found to be 0.91; the corresponding r^2 of 0.83 indicates the variance accounted for by regression.

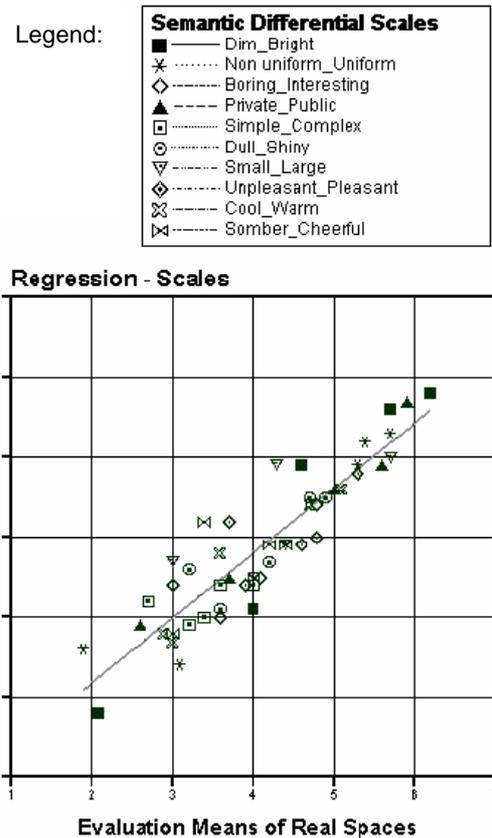


Figure 4. Comparison of subjective evaluative responses based on real and rendered spaces (Mahdavi and Eissa 2002)

These results imply a high level of congruence between impressions of the lighting gained from rendered images as compared to impressions gained from actual spaces. Thus, for the sample of participants and lighting scenes tested in the study, the results suggested that such images can reliably represent certain aspects of the lighting conditions in real spaces.

Digital surrogates of real buildings and spaces are rapidly improving, and immersive environments seem to be the next logical step in this evolution. Presumably, next to visual and acoustical sensations, other types of phenomenal experience (say sensation of radiative heat) may be induced via physical

translation of digitally produced patterns of stimuli. However, a cautionary word is here in order. Generation of phenomenally effective emulations implies a considerable level of overhead beyond the means and possibilities of the majority of the members of the design community. Moreover, evaluation processes that are effective in view of design decision making are typically of aggregate nature. The aggregation over time, location and view points is another overhead challenge.

The last (but not least) gap. Sound and validated simulation algorithms may reduce the gap between computational predictions and actual behavior of buildings. Digital emulations of phenomenally relevant behavioral features of designs may reduce the gap between perception of models and reality. We should not forget, however, another formidable gap, namely the one between the sensory basis of a perceptual situation and the actual evaluative judgment that arises from such situation. Confused? Let us consider a little experiment we performed a few years ago:

We asked six separate groups of participants to evaluate three (recorded) acoustical events using a 7-point differential scale. Each event was evaluated by two groups. Before the evaluation, we told each group what the event was (we did not tell them the truth) and we showed them – while there were evaluating the event – a fitting slide. In case of the first event (white noise), one group was told it was the sound of a waterfall, the other group was told it was a factory. In case of the second event (anonymus ambient sound in a large space) one group was told it was the sound of a representative entrance lobby, the other group was told it was a recording from a chaotic stock market hall. In case of the third event (anonymus office ambient sound), one group was told it was a modern generous office space, the other group was told it was a tense and over-crowded office space. The evaluation results are shown in the figures 5 to 7. Obviously, the attitude toward the source of the sensation has a significant impact on how its attributes are evaluated.

Human-ecologically speaking (Mahdavi 1998), simulation can provide information on the energetic aspect of relevant environmental factors. However, subjective evaluations are not at all fully determined by energetic descriptors of the so-called environmental exposure. Rather, such evaluations emerge through the complex workings of human information processing faculties. Interestingly, though, the information processing faculties do not appear to be quite fathomable to themselves. This is the point, if at all, where there would be a point in discussing possible limits pertaining to the explanatory powers of computational models.

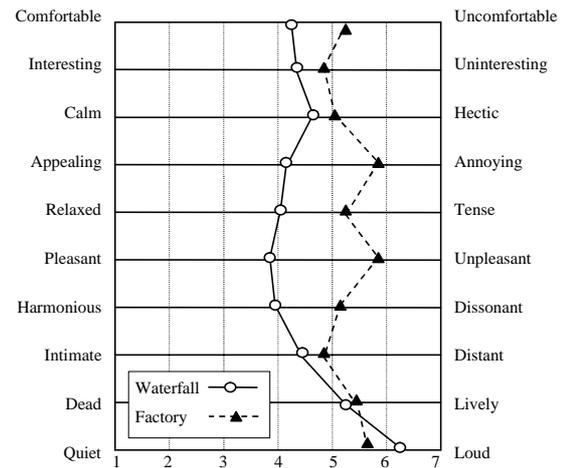


Figure 5. Evaluation of the first event a by two groups perceiving it as waterfall vs. factory

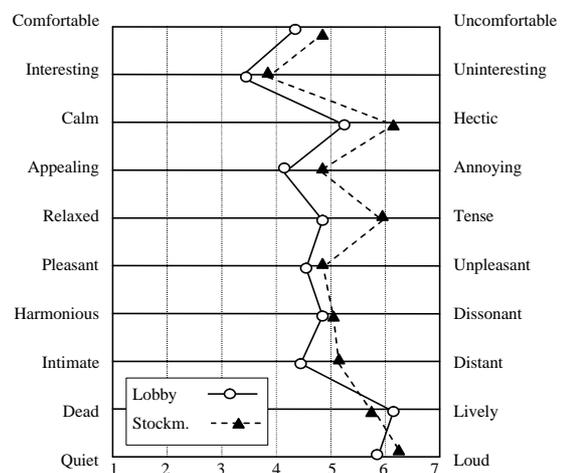


Figure 6. Evaluation of the second event a by two groups perceiving it as representative entrance lobby vs. chaotic stock market hall

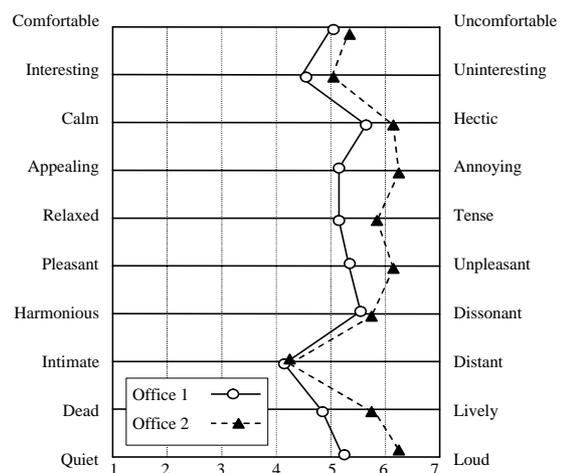


Figure 7. Evaluation of the third event a by two groups perceiving it as a modern (office 1) vs. over-crowded (office 2)

6. REFERENCES

- Augenbroe, G., Winkelmann, F. 1991. Integration of simulation into the building design process. Proceedings, Building Simulation 1991 (IBPSA) conference. pp. 367 – 374.
- Behrman, W. 2002. Best Practices for the Development and Use of XML Data Interchange Standards; CIFE Technical Report 143, Stanford University, Stanford, CA.
- Bertalanffy, L. 1962. Modern theories of development. Harper. New York.
- Brillouin, L. 1956. Science and Information Theory. Academic Press. New York.
- Broz, V., Carmichael, O., Thayer, S., Osborn, J., Hebert, M. 1999. ARTISAN: An Integrated Scene Mapping and Object Recognition System, American Nuclear Society 8th Intl. Topical Meeting on Robotics and Remote Systems, American Nuclear Society.
- Chang, S., Mahdavi, A. 2002. A hybrid system for daylight responsive lighting control. Journal of the Illuminating Engineering Society. Volume 31. Number 1. Winter 2002. pp. 147 - 157.
- CSI 1999. CSI: Integrated Information Initiative; Construction Specification Institute.
- Do, Y. L. 1996. The right tool at the right time – drawing as an interface to knowledge-based design aids. Proceedings of the 1996 ACADIA conference, Tucson, AZ, USA. pp. 91 – 97.
- Eggert D., Fitzgibbon A., Fisher, R. 1998. Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models, Computer Vision and Image Understanding Vol. 69, pp. 253-272.
- Elezkurtaj, T., Franck, G. 2001. Evolutionary Algorithms in Urban Planning. 6th International Symposium on Information and Communication Technologies in Urban and Spatial Planning, Vienna, Austria, 2001.
- Ellis, R. 2000. Efferent brain processes and the enactive approach to consciousness. In Humphrey, N. 2000. How to solve the mind-body problem. Imprint Academic. ISBN 0907845088. pp. 40-50.
- Fanger, P. O. 1970. Thermal comfort. Mc Graw-Hill Book Company.
- Faugeras O., Robert L., Laveau S., Csurka G., Zeller C., Gauclin C., Zoghliami I. 1998. 3-D Reconstruction of Urban Scenes from Image Sequences. Computer Vision and Image Understanding. Vol. 69, pp. 292-309.
- Finkenzeller, K. 2002. RFID-Handbuch. Hanser. ISBN 3-446-22071-2.
- Flemming, U., Baykan, C. A., Coyne, R. F., Fox, M. S. 1992. Hierarchical Generate-and-Test vs. Constraint-Directed Search. A Comparison in the Context of Layout Synthesis. Artificial Intelligence in Design '92. Boston: Kluwer Academic Publisher. pp. 817-838
- Harada, M., Witkin, A., Baraff, D. 1995. Interactive physically-based manipulation of discrete/continuous models. Computer Graphics (Proc. SIGGRAPH '95).
- Mahdavi, A. 2003. Notes on Representation in Computer-aided Architectural Design. (To appear).
- Mahdavi, A. 2001a. Aspects of self-aware buildings. International Journal of Design Sciences and Technology. Europa: Paris, France. Volume 9, Number 1. ISSN 1630 – 7267. pp. 35 – 52.
- Mahdavi, A. 2001b. Simulation-based control of building systems operation. Building and Environment. Volume 36, Issue 6, ISSN: 0360-1323. pp. 789-796.
- Mahdavi, A. 2000. Supporting collaborative design via integrated building performance computing. InterSymp-2000, 12th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany. Proceedings: Advances in computer-based and WEB-based collaborative systems. ISBN: 0-921836-88-0. pp. 91 – 102.
- Mahdavi, A. 1999. A comprehensive computational environment for performance based reasoning in building design and evaluation. Automation in Construction 8 (1999) pp. 427 – 435.
- Mahdavi, A. 1998. Steps to a General Theory of Habitability. Human Ecology Review. Summer 1998, Volume 5, Number 1. pp. 23 - 30.
- Mahdavi, A. 1997. Modeling-Assisted Building Control. Proceedings of the CAAD Futures '97 Conference, Munich, Germany. pp. 219 - 230.
- Mahdavi, A. 1993. Open Simulation Environments: A Preference-Based Approach. Proceedings of CAAD Futures '93. Pittsburgh, Pennsylvania, USA. pp. 195 - 214.
- Mahdavi, A., Berberidou, L. 1994. GESTALT: A prototypical Realization of an "Open" Daylighting Simulation Environment. Journal of the Illuminating Engineering Society. Volume 23, Number 2, Summer 1994. pp. 62 - 71.
- Mahdavi, A., Eissa, H. 2002. Subjective evaluation of architectural lighting via computationally rendered images. Journal of the Illuminating Engineering Society. Volume 31. Number 2. Summer 2002. pp. 11 - 20.

- Mahdavi, A., Gurtekin, B. 2002. Adventures in the design-performance space. European Meetings on Cybernetics and Systems Research Conference 2002, Vienna, 2-5 April, 2002.
- Mahdavi, A., Gurtekin B. 2001. Computational support for the generation and exploration of the design-performance space, Proceedings of IBPSA Conference, Rio de Janeiro, Brazil, 13-15 August 2001. pp 669-676.
- Mahdavi, A., Mahattanatawe, P. 2003. A computational environment for performance-based building enclosure design and operation. (To appear)
- Mahdavi, A., Mathew, P. 1995. Synchronous Generation of Homologous Representation in an Active, Multi-Aspect Design Environment. Proceedings of IBPSA Conference. Madison, Wisconsin. pp. 522 - 528.
- Mahdavi, A., Suter, G. 1997. On Implementing a Computational Facade Design Support Tool. Environment and Planning B: Planning and Design 1997, volume 24. pp. 493 - 508.
- Mahdavi, A., Wong, N. H. 1998. From Building Design Representations to Simulation Domain Representations: An Automated Mapping Solution for Complex Geometries. Computing in Civil Engineering; Proceedings of the International Computing Congress, ASCE Annual Convention. pp. 1-10.
- Mahdavi, A., Suter, G., Ries, R. 2002. A Representation Scheme for Integrated Building Performance Analysis. Proceedings of the 6th International Conference: Design and Decision Support Systems in Architecture. Ellecom, The Netherlands. ISBN 90-6814-141-4. pp 301 – 316.
- Mahdavi, A., Brahme, R., Gupta, S. 2001a. Performance-based computational design via differential modeling and two-staged mapping. Proceedings of the CAAD Futures Conference. Eindhoven, The Netherlands. pp. 667 – 680.
- Mahdavi, A., Brahme, R., Gupta, S. 2001b. Self-aware buildings: A simulation-based approach. Proceedings of the Seventh IBPSA Conference. Rio de Janeiro, Brazil. Vol. II. ISBN 85 – 901939 – 3 – 4. pp. 1241 – 1248.
- Mahdavi, A., Chang, S., Pal, V. 2000. Exploring Model-Based Reasoning in Lighting Systems Control. Journal of the Illuminating Engineering Society. Volume 29. Number 1. Winter 2000. pp. 34 – 40.
- Mahdavi, A., Ilal, M. E., Mathew, P., Ries, R., Suter, G., Brahme, R. 1999a. The architecture of S2. Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. III. ISBN 4-931416-03-9. pp. 1219 - 1226.
- Mahdavi, A., Chang, S., Pal, V. 1999b. Simulation-based integration of contextual forces into building systems control. Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. I. pp. 193 - 199.
- Mahdavi, A., Cho, D., Ries, R., Chang, S., Pal, V., Ilal, E., Lee, S., Boonyakiat, J. 1999c. A building performance signature for the "intelligent workplace": some preliminary results. Proceedings of the CIB Working Commission W098 International Conference: Intelligent and Responsive Buildings. Brugge. ISBN 90-76019-09-6. pp. 233 - 240.
- Mahdavi, A., Mathew, P., Wong, N. H. 1997a. A Homology-based Mapping Approach to Concurrent Multi-domain Performance Evaluation. Proceedings of the The Second Conference on Computer Aided Architectural Design Research in Asia: CAADRIA '97. Taiwan. (ISBN 957-575-057-8) pp. 237 - 246.
- Mahdavi, A., Mathew, P., Kumar, S., Wong, N. H. 1997b. Bi-directional Computational Design Support in the SEMPER Environment. Automation in Construction 6 (1997) pp. 353 - 373.
- Mahdavi, A., Brahme, R., Mathew, P. 1996. The LEK-concept and its applicability for the energy analysis of commercial buildings. Building and Environment 31(5). pp. 409 - 415.
- Markus, T. A., Morris, E. N. 1980. Buildings, Climate and Energy. London: Pitman Publishing Limited.
- Negroponete, N. 1975. Soft architecture machines. MIT Press, Cambridge, MA, USA, and London, England.
- Pelletret, R., Keilholz, W. 1999. Coupling CAD tools and building simulation evaluators. Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. III. ISBN 4-931416-03-9. pp. 1197 – 1202.
- Pohl, J., Reys, I. 1988. An integrated intelligent CAD environment. Proceedings, 4th International Conference on System Research, Informatics, and Cybernetics. Baden-Baden, Germany.
- Radford, A. D., Gero, J. S. 1988. Design by Optimization in Architecture and Building. Van Nostrand Reinhold, New York.
- Suter, G., Mahdavi, A. 1999. Performance-inspired building representations for computational design. Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. III. ISBN 4-931416-03-9. pp. 1203 - 1210.

