P3: An Integrated Environment to Support Design Collaboration

Yehuda E. Kalay
University of California, Berkeley
Kalay@ced.berkeley.edu

Buildings are the combined product of the efforts of many participants interacting in complex ways over a prolonged period of time. Currently, sequential communication among the participants is the standard means of collaboration. This method, which is well suited to current legal and professional practices, is inefficient, fraught with loss of information and prone to errors, cost and schedule overruns, and promotes optimization of individual parts at the expense of the overall project. This paper describes an integrated design environment that will facilitate collaborative decision making among the various participants, not merely communicate the results of decisions made by one participant to the other participants in the design team. It is based on the convergence of computing and telecommunication technologies, coupled with the emergence of new design paradigms, which together can overcome the technical difficulties associate with current collaborative design practices. It comprises three different modules: a product model, a performance model, and a process model (hence it is called P3). The paper presents each of these models and their integration into a unified framework.

introduction

Historically, the process of designing, constructing and using buildings has been handled by a large number of independent organizations (e.g., building owners, financial institutions, building users or their representatives, architectural and engineering firms, general and subcontracting companies, product manufacturers, etc.). They establish links with each other when the need arises, and sever them when their task has been completed. The relationships between the different organizations are, therefore, short-lived, and the interactions between them are temporary (Mohsini 1992). Thus, although a large number of organizations are involved in the building project for some period of time or another, not all are involved in the process simultaneously or at all times, as depicted schematically in *Figure 1*. As a result, 'collaboration' among the various participants in the design-build-use of buildings has been mostly sequential, and the individual organizations' work has been largely independent of each other.

Nonetheless, while the composition of the project team changes as the project evolves, and while individual organizations join the team at different times, their actions and the decisions they make are not independent from each other. In fact, they are highly inter-dependent: each organization begins its involvement not only with its own task-specific list of requirements, but also with the output of the tasks and the decisions made by the organizations that preceded its involvement. This high degree of interdependence works in two ways. On the one hand, an organization joining the team earlier in the process passes on its 'products' as constraints to the organizations joining the team later in the process. On the other hand, the ultimate performance of 'upstream' organizations are highly dependent on the performance of 'downstream' organizations. For example, a perfectly designed heating, ventilating, and air-conditioning (HVAC) system may be rendered ineffective if the installation of the windows is substandard, causing air leakage that exceeds the volume anticipated by the designers of the HVAC system, or if the users open the windows, thereby introducing unanticipated heating and cooling loads.

While this traditional mode of collaboration has sustained the building industry for many years, the steadily growing complexity of the built environment and the needs it must meet have been eroding its adequacy as we approach the 21st century. Complexities arise, in part, from the growth and evolution of industrialized societies and the social trends accompanying their growth, and in part from the growing complexity of the processes leading to the design, construction and management of buildings. The first is responsible for the grow-

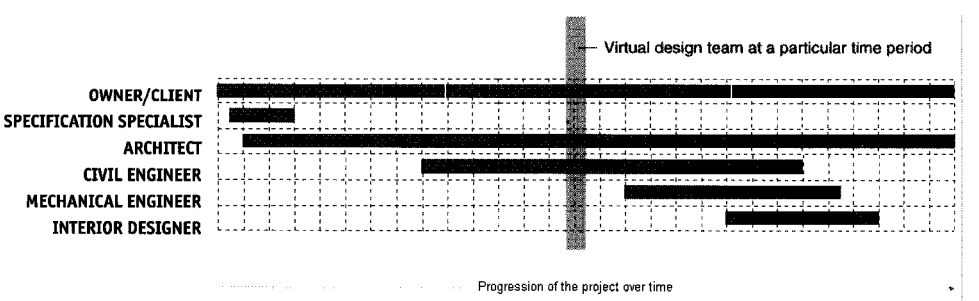

Figure 1 *A generalized schema of current design processes and organizations.*

ing emphasis on accountability and sustainability of the built environment. The latter is responsible for the steady expansion of theoretical, technological, and organizational knowledge and practices used by each one of the building-related professions, along with the growing awareness that decisions made by one participant in the design-build-use process impact the other participants. These difficulties have increased the need to better coordinate building-related activities and align them with technological, economical, political and other developments, in an efficient yet socially and environmentally responsible manner.

By most measures it is quite obvious that 'sequential' collaboration is inefficient, costly, and error-prone, for reasons that include loss of information, duplication of efforts, optimization of sub-systems at the expense of the whole assembly, and mismatches at the boundaries where one sub-system meets other sub-systems. This realization promoted such notions as 'work groups' and 'value engineering' in many disciplines, including aerospace and electrical engineering (witness the success of Boeing's development of the 777 in record time and cost). Why, then, collaboration methods that seem to work well in other design disciplines cannot be incorporated in the construction industry? What makes the design of buildings different from the design of airplanes or computer chips? An understanding of these inherent differences must, necessarily, precede any effort to develop new collaboration methods for the building industry, which will overcome the deficiencies of the established, largely sequential methods of collaboration.

the uniqueness of designing buildings

Much like in other industries, specialization has pervaded the building industry, and delegated each one of the building-related sub-systems to the care and responsibility of a different discipline. Thus, for example, the spatial layout of a building is the responsibility of architects, whereas the HVAC system is the province of mechanical engineers, and the structural stability of the building is guaranteed by civil engineers.

Yet, the success of the 'whole' depends as much on the success of the parts as it does on their successful integration, because the assemblage of the parts must have a logical and appropriate relationship to the whole. The specialists who are responsible for the individual parts often lack the expertise needed to integrate the parts into a meaningful whole. Similarly, the individual whose task it is to integrate the parts cannot also be an expert in all the specialties that contribute to the definition of the parts. This reliance on both wholeness and specialization at the same time can be likened to an orchestra, where the music is the result of both the artistry and talent of the individual musicians as well as the conductor's ability to harmonize them into a larger, more complete symphony.

The construction industry, however, differs from other industries in that the participants have professional backgrounds that differ vastly from each other. These include such specialties as civil engineering, electrical engineering, mechanical engineering, landscape architecture, interior design, economics, and management, to name a few. In fact, according to the AIA's Professional Practice Handbook, up to 32 different professionals may be involved in the design of any particular building. Others (Cuff 1991) estimate this number to be closer to 67. The integration of the products of all these specialists is the province of the architect, who is concerned with the completeness of the assembly and its 'fit' within the given context.

Much like in other enterprises, professionalism itself is a source of conflicts among the participants. This is not surprising, given that specialization is seen by society as the conferer of privileges not shared by others (hence it justifies higher compensation, among other things). Moreover, our legal system and litigious society

encourages limited risk-taking and staying within the bounds of established professional knowledge and practices. Professionals are trained to understand the world in a particular way, and develop a unique language to facilitate communication within their own sub-culture. Professional education teaches a 'right' way of seeing the world, and instills faith in that way that, over time, like a religion, becomes no longer open to challenge. Unfortunately, different professionals have different 'world views': architects, engineers, construction managers, facilities managers, building owners, and end-users all have different constructions of 'reality'–itself a product of the social systems through which human knowledge is developed, transmitted and maintained (Berger and Luckmann 67).

Accordingly, in general architects emphasize the *quality* of artifacts over their function, purpose, and the processes of making them. Engineers tend to emphasize the *function or purpose* of the artifact, placing less emphasis on the process of *making*, and still less on its formal qualities. Construction managers are interested mostly in the process of making, whereas facilities managers are interested in the process of *maintaining*. Owners and end-users are usually not interested in their environment, as long as it does not impinge on their activities and interferes with the achievement of their personal or institutional goals (i.e., how well the place supports the education of students, the fabrication of goods, or making people well).

While similar tendencies characterize most professions, there are two special characteristics that distinguish the process of designing buildings from all other collaborative enterprises:

~ The 'specialists' cannot begin their tasks before the 'integrator' has specified the overall assembly. For example, the engineers cannot begin to design the structure of the building before the architect has specified its shape. Yet, the architect cannot specify the shape of the building without considerable specialized information that is rightly the province of the engineers.

~ The language used by architects, and indeed their way of thinking, is radically different from the language and mode of thinking employed by the 'specialists,' leading to miscommunication or even resentment on both sides.

The focus of this paper is the emerging means that can address the first of these two premises: integration in the face of growing specialization. Yet, this discussion will be meaningless without first understanding the profoundness and ramifications of the second premise.

problem solving vs. puzzle making

Most of the specialists who are involved in the design and construction of buildings work in what has become known as *problem-solving* mode: a term coined by Herbert Simon in the 1960s to denote a process where the desired needs and effects of the problem can be stated explicitly prior to initiating the process for finding solutions that satisfy these needs. The design process thus becomes a process of testing alternative solutions against the stated needs, until one is found that meets them. Thus, for example, an HVAC engineer would design the ventilation system to meet stated cooling and heating needs, and a civil engineer would design the structural system of the building to meet the expected lateral and vertical loads (Simon 1969).

In contrast with the other specialists who are involved in the building process, architects cannot articulate the sought effects of the building prior to initiating the design process, because the sought building is always new and unique. As such, its design involves discovery and invention whose parameters can only be revealed through the design process itself. The process of *designing* reveals to the architect the limitations and the opportunities of a given context, as well as the tradeoffs that are needed to develop the most appropriate set of architectur-

al elements and their attributes that best fit the client's needs in a given context. This mode of action has been termed *puzzle making* by Archea (1987). Thus, rather than use the client's definition of the desired effects of the sought building as a complete problem definition, architects can only use them as a starting point and a catalyst for the design process, something that provides a sense of direction and a sounding board for potential resolutions. Instead, they are guided in their search for a solution by precedents, symbols, and metaphors. These are 'complete' solution packages, which have been tested through years or centuries of acculturation. They provide a sense of 'wholeness,' thus they can be modified to fit the specific needs of the project without losing sight of how they ultimately fit together to make a complete assembly.

These two modes of practice are incompatible with each other: on one hand, the specialists need a precise definition of the desired effects their part in the overall product must achieve, before they can initiate the search for solutions that meet them. On the other hand, the architect cannot define the desired effects until the design process is well under way, and the effects of various partial 'solutions' can be tested to see if they achieve the desired 'fit.'

While these different modes of operation are the butt of many professional jokes and a source of tension between architects and consulting engineers, they are, nonetheless, very real. They are, moreover, indicative of the difficulties facing integration and collaboration in the design team, which is composed of distinct participants who use different modes of operation, yet all of whom must work in concert to accomplish the best results.

habitual modes of collaboration

The importance of collaboration in the building industry has been recognized even during the Middle Ages, when the complexity of large building projects, like cathedrals and castles, required the joint effort of many skilled artisans. Collaboration, at that time, could only be achieved by physical and temporal proximity, where the artisans formed a society, or *lodge* of professionals, who lived and worked together for the duration of the project (typically, many years) (Erlande-Brandenburg 1993).

The advent of paper-based drawings during the Renaissance made it possible to collaborate a-synchronously, by passing the results of the efforts of one professional to the others in the form of drawings, a method that is still very much in use today (Jones 1980).

Over the past two decades, technological innovations made possible by advances in telecommunication and computing have extended the means available to building professionals for coordinating and for managing collaborative design processes. Several different directions have emerged, including collaborations based on developing common *data exchange* models, starting with IGES (Smith 1983), through IPAD (Boeing 1980), PDES (Brauner 1986), and culminating in STEP (Augenbroe & Winkelmann 1990). These efforts have been aimed at solving the technical aspects of communicating building data, and are characterized by a steadily growing ability to communicate the semantic content of the exchanged information (i.e., from lines and other graphical symbols to product information). By and large, these efforts have not addressed the *process* aspects of the collaborative design problem, and continued to assume sequentiality as the dominant mode of communication among the professionals. Thus, while they have made *communication* easier and more efficient, they really did not make *collaboration* any better: the decisions made by each participant are still isolated and do not benefit from the input of other participants.

More recently these shared-product based efforts have been aimed at developing *integrated design environments*, where all the different participants have access to a common database,

which is complete, up-to-date, and self-consistent. Examples of such integrated design development environments include ones developed at Carnegie Mellon University (Fenves et al 1994), at Cal Poly (Pohl & Myers 1994), and at Stanford University (Khedro et al 1993). These approaches address mainly the technical difficulties associated with concurrent access to engineering databases, and the need to provide all the participants with up-to-date and consistent information. To their credit, they did provide a means for communicating among the participants in the project, but did not address the process-related difficulties that were discussed earlier.

Another approach that was intended to enhance collaboration has been aimed at *cross-educating* professionals involved in the design of buildings, making each aware of the philosophies, needs, and methods of the others. Such efforts have taken the form of integrative, multi-disciplinary courses, such as a project funded by the U.S. Army Corps of Engineers Construction Productivity Advancement Research program (CPAR 1994), and NSF's Synthesis Coalition joint A/E/C course in Stanford University and UC Berkeley (Fruchter 1994). These efforts addressed collaboration as an expansion of the skills of the individuals involved in the process, rather than the pulling together of the dispersed expertise.

computational models of collaboration

The common thread in all these directions is their emphasis on sharing the product of the efforts of the disparate professionals. They have focused mainly on solving the first problem of collaboration, namely–the organizational and control issues associated with the dissemination of the products designed by each professional to the other professionals, in a timely and efficient manner. Their secondary goal has been the discovery, investigation and solution of problems arising from the attempt to integrate the various subsystems into one whole, namely– conflict detection and resolution, management of tradeoffs, and correction of errors.

Yet, the central problem associated with collaborative design remains unsolved: the development of a design environment which will facilitate collaboration among the various participants *while* design ideas are being formed, rather than *after* they have been formed. In other words, the real challenge facing the building industry is not finding a more efficient means for disseminating the intellectual products of the specialists, but rather their orchestration in a *joint decision making process*. That is, we need to develop an integrated *collaborative design process* model, in addition to an integrated design *product* model. Only then will there exist the necessary integration of the knowledge which has become distributed among many specialists, without sacrificing the advances that were made by each one of the participating disciplines.

Why have the efforts of many talented researchers failed to accomplish what many believe to be a desirable outcome? In general, the development of an integrated collaborative design process model has been hampered by two difficulties:

~Sharing the semantic content of the information in addition to its syntactical content.

~Providing timely and appropriate feedback on design proposals made by each professional from the point of view of the other professionals.

While it is easy enough to represent the geometry of a building and the materials it is made of, it is much more difficult to represent the rationale behind selecting a particular geometry or material, which is where the expertise of the various participants really resides. By sharing semantic information, and by providing timely feedback–whether asked for or not– the various specialists will have a better understanding of the reasons behind the decisions made by their colleagues, be able to present their objections, if any, and generally help make decisions that are in tune with those made by others, or suggest modifications to

those decisions which will preserve their intent while offering a different solution to implement them.

product modeling

Sharing semantic information is complicated by the multitude of different 'semiotics' which are brought to bear on the same product by the different professionals. Efforts to make the database all inclusive have been hampered by the ubiquitous problem of generality vs. specialization: the more general the data needs to be (so it will be useful to as many different applications as possible), the less specialized it can be, and vice versa–the more specialized it is, the less general it becomes (Kalay 1989). Hence, a building product databases typically includes only factual information about the objects they describe (often only geometry and materials). They assume that the readers of the data will interpret it correctly, using their own professional knowledge. This assumption, however, is false, because the different professionals who are involved in the building project have been educated to interpret the same artifacts differently (Valkenburg 1996).

performance modeling

To overcome this problem, researchers have tended to compensate for the low level of semantic information contained in the building database by embedding more semantics in the disciplinary-specific tools that process the data. Thus, for example, energy evaluation tools use the geometric and material information that is embedded in the product database to calculate the R values and heat capacity of the walls, which are then combined with externally-supplied climactic information and expected occupancy schedules to calculate the energy performance of the building. This performance prediction is used to estimate the level of comfort afforded by the built environment, which, in turn, is used to estimate the productivity and health of its occupants. The information pertaining to the predicted uses of the building, and the results of the evaluation, typically reside in the *evaluation* routines, rather than in the evolving *building model* itself. Hence, they are unavailable to other evaluators, who typically must make their own assumptions, often re-creating the same intermediate constructs. There are no assurances that these evaluators (e.g., fire egress, lifecycle costing, etc.) will make the same assumptions as the ones made by the thermal comfort evaluator, since they often use their own knowledge bases and defaults (Shaviv & Kalay 1992).

This *evaluation-based* representation made it possible to develop the current host of design and evaluation tools used by the building industry. It came, however, at the expense of collaboration: the database includes only the *results* of design decisions made by each one of the participating disciplines, and none of the *reasons* for making them, nor their expected *implications*. Given that the building database is often the only persistent record of the building's design process, the complex and expensive efforts that were made by the various participants in the process are lost, and must be re-created when needed (e.g., when the building needs to be modified to accommodate new uses) (Hitchcock 1996).

process modeling

Product and performance modeling approaches have contributed much to our understanding of how products can and need to be represented such that they can be used by the different professionals involved in the building design process. However, they failed to result in an overall design environment that is conducive of collaborative, multidisciplinary design.

Their failure can be attributed to the fact that both product and performance models are *static* representations of the design process: they do not provide a representation for the processes of deliberation, negotiations, and the other dynamics that characterize a design process.

These 'operational' aspects of design have, thus far, been assumed to reside outside the purview of the representational models.

We argue that a complete model of collaborative design must also represent the dynamics of the transactions and tradeoffs that occur while design ideas are developed, for that is where the design intent is represented, negotiated, modified and tested. That is where the overall *valuation* of the emerging collective product is done.

Thus, a *process model* of design is needed, in addition to the product and performance models, to fully represent the collaborative design process.

The development a process model of design is not a new idea. It was first suggested by Rittel (Kunz & Rittel 1970), and later implemented in his IBIS system (Noble & Rittel 1988). Another version of the same idea was implemented by McCall in his PHIBIS system (McCall 1987). Portion of this idea have been tested by Pohl in his ICADS system (Pohl & Myers 1994). Yet, all these systems have sacrificed the other two components (product and performance models) for the sake of promoting the deliberative model: each has developed private, non-sharable product and performance models.

A sharable process model cannot, however, rely on private product and performance models, since it relies on them for representing the substance and subject of the deliberations. In the following we present how we propose to develop such an integrated, sharable model of collaborative design, which includes all three models.

P3: an integrated environment for design collaboration

Three complementary constructs are needed to support a computational environment capable of collaborative, multidisciplinary design:

~A shared *product* model, which is the repository of all the information pertaining to the emerging design solution and the information that is needed to understand and evaluate it.

~A sharable *performance* model, which is a collection of procedures that can predict the expected performance of the emerging design solution, and evaluate it in comparison to set goals.

~A shared *process* model, which represents the deliberative process, tracks design intents and decisions, their supportive and counter arguments, tradeoffs, and negotiations.

We call the collection of these three components **P3**, for **Product, Performance, and Process** models. In the following, we outline briefly how we are implementing each one of the three major components, and how they all work together.

product model

The product model is a collection of databases that provide the means to share information about the emerging design solution among the participating professionals, and a means to keep tack of the evolving product. It is a collection of several different kinds of information that, for data management reasons, could be partitioned into the following types:

~*Object data* represents information that is object-specific but project-independent; that is, information that can be attached to a specific object (shape, material, cost, behavior, etc.), and does not change from one project to another. For example, information that describes the work-triangle in kitchens, minimum and maximum kitchen areas, cabinet height and depth, lighting needs, and so on, is information that can be attached to a KITCHEN object, and does not depend on the project in which this object is used. Moreover, this construct allows us to represent *exceptions*, in the form of if-then-else rules, which modify the attributes or behavior of the object in specific, well-known cases. For example, the work-triangle in the kitchen can be modified to allow for two cooks, rather than the traditional one-cook arrangement, a case

that involves dimensional and internal-connectivity issues. The Object Database (ODB) is also the repository of the *semantic* information that pertains to all the objects used in the project. This is how the *meaning* behind the objects is conveyed to all the participants.

~*Project data* represents the particular *instances* of the more general object data, which are pertinent to the particular project. For example, this information includes the *specific* dimensions of a particular kitchen, the *specific* materials it is comprised of, as well as the connections between them–the *assemblies* that individual entities make when they come together in the specific project. The Project Database (PDB) represents the evolving state of the design in which the participants operate. This is how the *syntax* of the project is shared among the participants.

~*Context data* represents information about the specific context of the project. Context is taken here in its broadest sense: information that architects (and other design professionals) must respond to in their design, and over which they have little or no control (e.g., topography, climate, views, cultural environment, economic and political environment, zoning codes). Additionally, the context also comprises the predominant activities that the building must support that are typically implied by the nature of the project (e.g., medical procedures for treating patients in a hospital, the method of teaching in a school, and traditional habits of a family within its own house). The Context Database (CDB) is the repository of information concerning the physical and temporal information about the project.

object databases (ODBs)

For reasons of efficiency, as well as the need to allow for independent growth, the Object Database is really a collection of many Object Databases. Each one of the ODBs is the repository of specific object-knowledge. It contains the information that is needed to understand and to evaluate objects of the kind it represents. This information includes:

~*Classification* relationships that define an inheritance path for properties this object shares with other, more generalized objects of the same type (e.g., that the kitchen is a kind of indoor space).

~*Attributes*, such as the object's name, its form (shape), the materials it is made of, and other properties (e.g., manufacturer, cost, etc.).

~*Functions* that describe the purpose(s) of the objects (e.g., that a kitchen is a place where food is prepared and stored).

~Integrity *constraints* that are logical propositions associated with attribute values, and define generic expectations from the object (e.g., that the work triangle must be within certain dimensional limits).

~*Cases*, in multimedia form that provide anecdotal information about the object.

We are implementing an object database as a collection of web pages. The World Wide Web (www) provides a particularly convenient means to represent objects of this kind: it is inherently a multi-media environment, and is eminently linkable. Objects can be represented as web pages that are linked in hierarchical classification structures. These links facilitate *inheritance*, which conserves storage space and eliminates redundancies. For example, the KITCHEN object shares many attributes with other indoor spaces, such as bedrooms and living rooms, in that it is enclosed by walls, has indoor thermal properties, it is a habitable space, and so on. The classification hierarchy stores shared attributes at the lowest level in the tree that is shareable by all the objects that can benefit from this information. An inheritance mechanism, to be implemented in Java, will make higher-level information available to objects that are stored in lower levels of the hierarchy.

The www and Java endow this representation with two distinct advantages that are most

useful for our purposes of facilitating collaborative design: it can be easily distributed, and it can be searched dynamically. The ability to easily link web pages that reside on many different www sites makes it possible to develop objects by different people, in physically and temporally disconnected places, and link them together to form a cohesive database. For example, the cabinets in the kitchen could be developed by a cabinet manufacture, while the appliances by appliance manufacturers. They can both be linked to the KITCHEN object, which might be developed by a kitchen designer. This static web structure can be made into a dynamic database by employing specially written Java programs that can search the database, and retrieve the inherited attributes.

(For a more detailed description of the ODB, see (Khemlani et al 1997) elsewhere in these Proceedings.)

project-specific database (PDB)

The Project Database (PDB) represents the specific, emerging product. It is the computational construct where the generic information form the ODBs is instantiated, where specific values replace default values, and where assembly information is added.

We have developed a non-redundant representation which includes both structural and spatial building elements, using a data structure modeled after the well-known winged-edge model (Kalay 1987). The specific information in the PDB can be linked to the objects data in the ODBs, providing a dynamically updatable representation of the building. For example, the specific shape of the kitchen and its relationship to other rooms in the house would be represented in the PDB. However, the nature of the kitchen as a kind of indoor space is represented in the ODB.

(For a more detailed description of the PDB, see (Khemlani et al 1997) elsewhere in these Proceedings.)

performance model

The Performance model is being implemented in the form of operators that we call *IDeAs* (Intelligent Design Assistants). These are semi-autonomous 'agents' that are able to carry-out high-level instructions, perform tests and make suggestions on their own accord. They are computational constructs that can represent discipline-specific operational information. As a construct that is separate from the database, they have a global view of the information. Thus, they can access information that is scattered among several (perhaps all) objects. For example, they can calculate the overall cost of a project, its thermal and structural properties, and so on.

Unlike *agents* in the AI sense, IDeAs do not purport to replace the human actors. Rather, their role can be likened to highly skilled junior designers, financial analysts, or even secretaries, who have limited authority in making decisions. They are, however, able to interpret the (sometimes abstract) instructions given to them by the human actors. For example, given a general sketch of a kitchen layout, they should be able to look up standard cabinets and fixtures and transform the sketch into a detailed plan which conforms to certain design goals. The extent of their action depends on run-time circumstances, including the disposition of the human actor. This can range from simply providing information to the designer, all the way through creating a complete design solution for consideration by the client (Irazabal 1995, Llavaneras 1996).

An IDeA comprises a *semantic interpretation* module, a user interface, and several knowledge bases. *(Figure 2)* The semantic interpreter, which is essentially a decision-action system, is responsible for the communication between the PDB and the knowledge bases of which the IDeA is comprised. For example, it can search the PDB for the information needed to perform a thermal evaluation, and use the knowledge bases to perform the analysis. The user interface, in turn, provides both a means to control the actions of

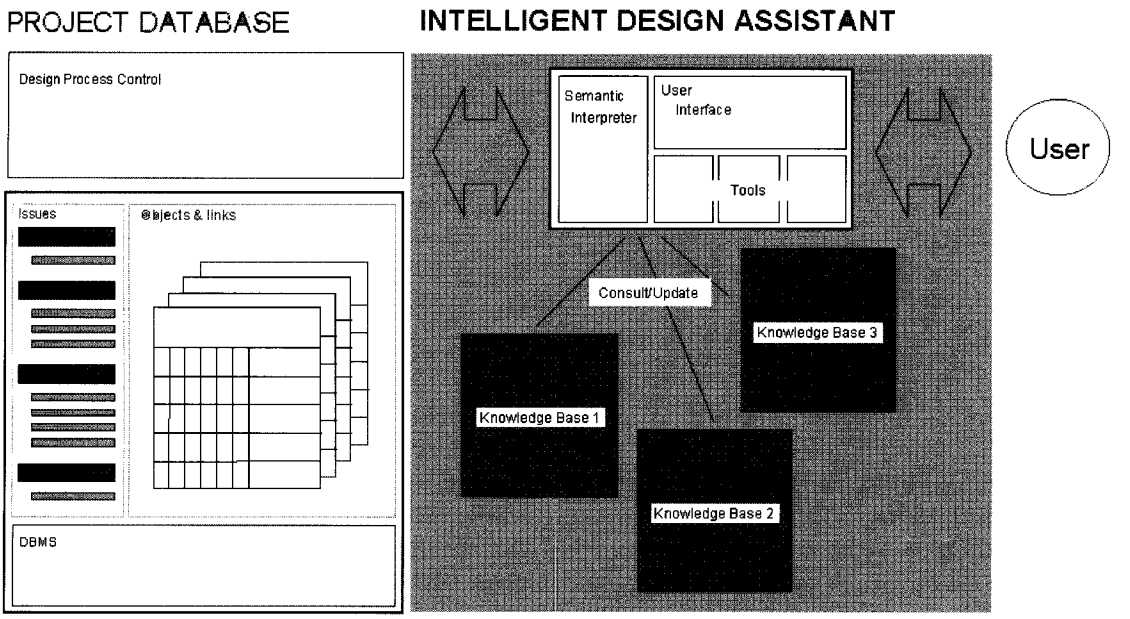

Figure 2 *Main problem page for a precedent design problem.*

the IDeA, and to present to the user the results of the consultation.

It is obvious what role Java can play in implementing this construct, enabling the IDeA to search both a remote PDB and many ODBs. The results of its action will be reported back to users who triggered the IDeA. Furthermore, such IDeAs could be developed by the most suitable professionals, and could 'inhabit' Cyberspace, waiting to be called to action...

process model

The Process model is where design intent is represented explicitly, and where tradeoffs are made. It comprises two main constructs: an Issues Database (IDB),and a tradeoff representation unit.

issues database

Design intents are represented in our model as an Issues Database (IDB). This is where the opinions, value judgments, and professional valuations concerning various parts of the project are recorded and shared with the other participants. Issues include such non-physical things as light, sound, and wayfinding, and represent the considered opinions of the participants regarding the treatment of such topics. The importance of tracking issues through the design process has been demonstrated by Robert Hitchcock, when applied to the commissioning of buildings (Hitchcock 1996). Essentially, the IDB could be thought of as a threaded discussion, albeit tagged for easy identification of pro, con and other types of responses.

tradeoffs

Tradeoffs are the hallmark of every design activity. Typically, all the functional needs of a building cannot be satisfied by any one design solution. The achievement of certain needs often must comes at the expense of other needs. For instance, eliminating windows on the west side of a building to save energy might also deprive the inhabitants of a fabulous view. Hence, the degree of satisfying some needs may have to be

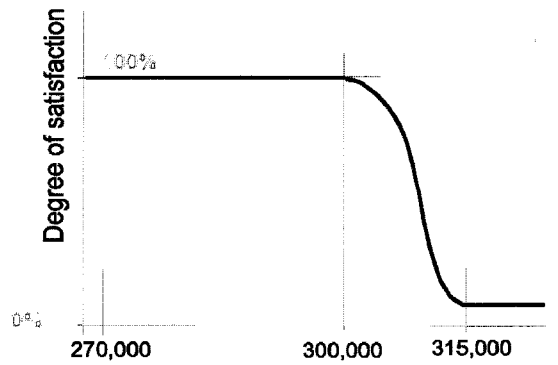

Figure 3 *A typical satisfaction curve.*

compromised, so that others are also satisfied. But how much should any one need be compromised? To do so, we have built upon a notion forwarded a long time ago by Horst Rittel, which he called Performance Curves and we re-named Satisfaction Functions.

Satisfaction functions are *mappings* that express the specific relationship between the *behavior* of a system (as predicted by some performance modeling agent) and the subjective measure of its *desirability* under specific circumstances. *Figure 3* depicts a typical satisfaction curve: on one axis, it measures the behavior of some aspects of the designed system (its cost, in this case). On the other axis, it measures the degree of satisfaction each behavior value elicits in the client.

The curve demonstrates several phenomena commonly associated with satisfaction. For example, that the client may generally be satisfied with the behavior of the system, until its behavior reaches a certain threshold. Then satisfaction diminishes, but the change from 100% (completely satisfied) to 0% (not satisfied), is gradual. The curve allows for such fuzzy notions as 'quite satisfied,' 'more or less satisfied,' or 'barely satisfied,' to be expressed. The slope of the curve allows us to express the rate of change: the steeper the slope, the more abrupt the change, which means that once the threshold has been reached a small change in the system's behavior will result in satisfaction or dissatisfaction. On the other hand, a shallow slope indicates a wider latitude in satisfying the client, which allows more room for tradeoffs with other satisfaction curves that may need to be modified.

The satisfaction functions must, of course, be set by the client, or by the designer. They are unary functions, in the sense that each curve pertains to satisfaction derived from one behavior only. This makes it possible to set them individually.

Satisfaction functions can be developed for each aspect of the building. The mappings they afford are expressed as numerical values, each of which expresses the client's satisfaction with respect to one specific behavior. To aggregate the separate satisfaction functions into one composite measure of performance, we add them up. But since different behaviors weigh differently in the overall performance measure, we must first assign to each of them a relative weight. This method is well established, and has been used by other researchers to develop aggregates of multi-criteria evaluations (Wizel & Becker 1989). The composite result of the summation of weighted, normalized satisfactions is presented to the client as the overall performance of a given design solution.

The satisfaction functions facilitate the difficult tradeoff decision making process, in three ways:

~By explicitly showing how well any one need is being satisfied, as a percentage between full and zero satisfaction

~By expressing the tolerance for satisfying the expressed need, in terms of the steepness of the curve.

~By prioritizing the relative importance of each need, in terms of the weights assigned to it.

Using these three measures, it is possible to identify needs that are not being satisfied, and those that are over-satisfied. It is possible, therefore, to seek a design solution that better achieves the under-satisfied needs, while achiev-

ing less-well the over-satisfied needs. In fact, an algorithm can be developed that provides hints to the designer, indicating possible tradeoffs. It first identifies the under-satisfied needs, then the over-satisfied ones. Among the over-satisfied needs, it would suggest that those of lower importance (as expressed by their associated weights) would be candidates for reduced-satisfaction. It will also indicate how much latitude exists in reducing their satisfaction levels.

Given that the inter-relationships between the different needs are not obvious, for the most part, the algorithm cannot tell which specific need ought to be compromised to achieve another need. Such advise could be added through a knowledge base, which stores rules about the relationships between the various needs. It might also store specific suggestions for improving under-satisfied needs. Nonetheless, since the correlation between the functions is non-linear, only a complete new design solution can, in general, make all the necessary adjustments.

bringing it all together

Figure 4 depicts the overall structure of the integrated design process model. It shows how several IDeAs can collaborate on one project, and how they can partake in several projects at the same time. One of the IDeAs acts as project manager for each project. Figure 4 also shows how a PDB can draw on several ODBs, and how the ODBs can contribute to several projects at the same time.

As the need for collaborative design grows, and as its benefits become more evident, the loss of information fostered by the separateness of design professionals becomes less acceptable. The highly inter-linked nature of design, construction and management of buildings makes it imperative that more information be shared by all the participants, all the time. This realization has promoted the concept of *value engineering*, a term that describes building (and other) design practices when many participants in a project come together to discuss pending issues and resolve outstanding problems. Such 'round-table' discussions have proven effective in breaking deadlocks and short-cutting otherwise long communication and authority lines. Yet, they require a significant expenditure of time, effort and money to bring together all the participants at the same time in the same place.

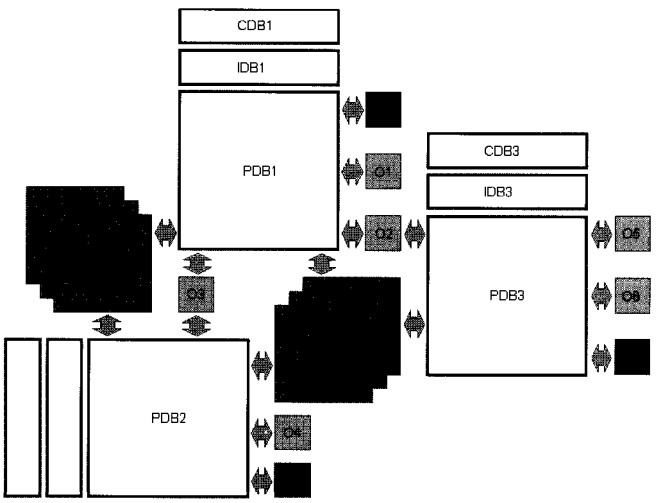

Figure 4 *The overall schema of the integrated building model.*

To reap the benefits of 'roundtable' collaboration, without incurring the costs associated with synchronicity, we propose to develop a unified collaborative design model that is accessible and comprehensible to all members of the building team, and can provide them with all the information that is relevant to their particular discipline. Moreover, they should be able to access this information a-synchronously, and from anywhere. The information that needs to be shared includes not only the *results* of decisions made by other participants, but also the *rationale* behind these decisions. Furthermore, this information needs to be provided *when* decisions are made, so input from other participants in the design process can be taken into account while it still counts.

We have provided a tentative solution to this problem, through the use of a three-tiered model, which we call **P3: Product, Performance, and Process** model. Needless to say, the actual implementation and testing of a large project such as the one described in this paper is a lengthy process, replete with many problems. Many questions remain unsolved at this time, or even un-asked. They include communication and control issues, knowledge representation issues, and user interface issues.

The development of the project proceeds through mock-ups of the overall structure, and through specific development of software for the various databases and for selected IDeAs. A multi-platform approach is currently being pursued, though we hope to migrate to Java as soon as practical. The mock-up is done through the World Wide Web. The PDB has been implemented in Code Warrior C++ on the Apple Macintosh platform. One of the IDeAs, a Fenestration Agent, has been implemented in Visual Basic 4.0 on a PC platform.

Probably the major outstanding question is how the design process itself will be affected by the provisions of the integrated model that were described in this paper, and how it could be managed. We do not currently have answers to these questions. It is the aim of this work to support designers, whichever way they choose to work. Given that collaborative design is on the rise, that is what needs to be supported.

acknowledgments

We have begun to develop and test the concepts that contribute to the project described here over 10 years ago, through several other projects (which included KAAD (Carrara et al 1994), WorldView (Kalay 1987), and ALEX (Swerdloff & Kalay 1987)). The current cycle of implementation and testing involves many individuals, from the departments of Architecture and Computer Science in Berkeley, as well as the department of Architecture at Ajou University, in South Korea, through the person of Professor Jin Won Choi , from the Department of Architecture, Ajou University, who spent nearly a year working with us at Berkeley as a Visiting Scholar.

references

Archea, J. 1987. "Puzzle-Making: What Architects Do When No One Is Looking," *Computability of Design* Y.E. Kalay, ed., Wiley Interscience, New York, NY.

Augenbroe, G. and F. Winkelmann 1990. *Integration of Simulation Into Building Design: the Need For A Joint Approach*, Technical Report, Building Physics Group Delft University of Technology Delft, The Netherlands, and the Simulation Research Group, Lawrence Berkeley Laboratory University of California.

Berger P.L. and T. Luckmann 1967. *The Social Construction of Reality*, Anchor, New York.

Boeing Commercial Airplane Co. 1980. "Development of Integrated Programs for Aerospace Vehicle Design IPAD: IPAD Evaluation and Alternatives," Report No. D6-IPAD-70036D, Vol. 3.

Brauner, K. et al 1986. *PDES Initiation Activities*, IPO, May.

Carrara G., Y.E. Kalay and G. Novembri 1994. "Knowledge-Based Computational Support for Architectural Design," *Knowledge-Based Computer-Aided Architectural Design* G. Carrara & Y.E. Kalay, eds., Elsevier Science Publishers, Amsterdam, The Netherlands.

CPAR 1994. *Request for Proposals*, U.S. Army Corps of Engineers Construction Engineering Research Laboratory, Champaign, Illinois.

Cuff, D. 1991. *Architecture: The Story of Practice*. MIT Press, Cambridge, MA.

Eastman, C.M. 1989. "Building Modeling in Architectural Design," Technical Report, Graduate School of Architecture and Urban Planning, UCLA.

Erlande-Brandenburg, A. 1993. *Cathedrals and Castles Building in the Middle Ages*. Harry N. Abrams, Inc. New York.

Fenves, S., U. Flemming, C. Hendrickson, M.L. Maher, R. Quadrel, M. Terk, R. Woodbury 1994. *Concurrent Computer-Aided Integrated Building Design*, Prentice-Hall, Inc. Englewood Cliffs, NJ.

Fruchter, R. 1994. "The Virtual Atelier," *Bridging The Generations*, CAE Workshop, Carnegie Mellon University.

Hitchcock, R. 1996. "Improving Life-Cycle Information Management through Documentation of Project Objectives and Design Rationale," PhD dissertation, Department of Civil Engineering, University of California, Berkeley.

Irazabal, C. 1995. *Intelligent Design Agents*, Masters Thesis, University of California, Berkeley.

Jones J.C. 1980. *Design Methods*, John Wiley & Sons, London, UK.

Kalay, Y.E. 1987. "WORLDVIEW: An Integrated Geometric-Modeling/Drafting System," *IEEE Computer Graphics & Applications*, 27:36–46.

Kalay, Y.E. 1989. *Modeling Objects and Environments*, Wiley Interscience, John Wiley and Sons, New York.

Khedro, T., M. Genesereth and P. Teicholz 1993. "FCDA: A Framework for Collaborative Distributed Multi-disciplinary Design," in *AI in Collaborative Design* Gero & Maher, eds., pp. 67–82, AAAI, Menlo Park.

Kunz, W. and H.W.J. Rittel 1970. "Issues as Elements of Information Systems," IURD Working Paper # 131, University of California, Berkeley.

Llavaneras G. 1996. "Fenestration Agent," Technical Report, Department of Architecture, University of California, Berkeley.

McCall, R. 1987. "PHIBIS: Procedurally Hierarchical Issue-Based Information System," proceedings of the conference on Planning and Design in Architecture J.P. Protzen, ed., pp. 17–22, Boston, MA.

Mohsini, R. 1992. "On Measuring Project Performance: Some Problems of Aggregation," *Evaluation and Prediction in Design* Y.E. Kalay, ed., John Wiley & Sons, New York, NY.

Noble, D. and H.W.J. Rittel 1988. "Issue-Based Information Systems for Design," ACADIA proceedings, pp: 275–287.

Pohl, J. and L. Myers 1994. "A Distributed Cooperative Model for Architectural Design," *Knowledge-Based Computer-Aided Architectural Design* G. Carrara & Y.E. Kalay, eds., Elsevier Science Publishers, Amsterdam, The Netherlands.

Shaviv E. and Y.E. Kalay 1992. "Combined Procedural and Heuristic Method to Energy Conscious Building Design and Evaluation," in *Evaluating and Predicting Design Performance* Y.E. Kalay, ed., John Wiley & Sons, New York, NY.

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

Smith B.M. 1983. "IGES: A Key to CAD/CAM Systems Integration," *IEEE Computer Graphics and Applications*, 113:78–83.

Swerdloff L.C. and Y.E. Kalay 1987. "A Partnership Approach to Computer-Aided Design," Computability of Design Y.E. Kalay, ed., Wiley Interscience, New York, NY.

Valkenburg, R.C. 1996. "Shared Understanding as a Condition for Design Team Decision Making," in proceedings of Descriptive Models of Design Ö. Akin, ed., Istanbul, Turkey.

Wizel A. and R. Becker 1992. "Integration of Performance Evaluation in Computer-Aided Design," Evaluating and Predicting Design Performance Y.E. Kalay ed., Wiley Interscience, New York.