AN INTEGRATED COMPUTING ENVIRONMENT FOR COLLABORATIVE, MULTI-DISCIPLINARY BUILDING DESIGN

LACHMI KHEMLANI and YEHUDA E. KALAY
Department of Architecture
University of California at Berkeley

Abstract. The increasing complexity of the built environment requires that more knowledge and experience be brought to bear on its design, construction and maintenance. The commensurate growth of knowledge in the participating disciplines—architecture, engineering, construction management, facilities management, and others—has tended to diversify each one into many sub-specializations. The resulting fragmentation of the design-built-use process is potentially detrimental to the overall quality of built environment. An efficient system of collaboration between all the specialist participants is needed to offset the effects of fragmentation. It is here that computers, with their ubiquitous presence in all disciplines, can serve as a medium of communication and form the basis of a collaborative, multi-disciplinary design environment. This paper describes the ongoing research on the development of such an integrated computing environment that will provide the basis for design and evaluation tools ranging across the many building-related disciplines. The bulk of the discussion will focus on the problem of a building representation that can be shared by all these disciplines, which, we posit, lies at the core of such an environment. We discuss the criteria that characterize this shared building representation, and present our solution to the problem. The proposed model has been adapted from geometric modeling, and addresses explicitly the difficult problem of generality versus completeness of the represented information. The other components of the integrated environment that are under development are also described. The paper concludes with some implementation details and a brief look at two evaluation tools that use the proposed building representation for their task.

1. Introduction

If an opinion poll were to be conducted among people about their approval of the built environment, most would express their dissatisfaction with some aspect or another of their living or working conditions. Undeniably, modern buildings are far from perfect in their ability to satisfy the physical, social, cultural, and economic needs of the people who are affected by them. The need for an improvement in the overall quality of our buildings is consistently felt in all sections of our society.

What does an “improvement in quality” really entail? It would mean raising the standards of all the individual processes that a building “lives” through in its lifecycle its design, construction, and maintenance—as well as the overall quality of the combined effects of these processes. For they are not discrete processes; rather, they are highly interdependent. Hence, coordinating them while striving to achieve greater overall quality is highly critical to the building process.

As our society becomes increasingly more complex, a proportional increase in the complexity of the built environment is inevitable. Buildings must fulfill a host of diverse criteria, abide by innumerable codes and rules, and an ever-increasing list of
constraints. A direct consequence of this increased complexity has been an enormous growth in the number of diverse professionals who need to be involved in the design and construction of a building. Gone are the days when this responsibility rested almost entirely on the shoulders of the "master builder." Today's built structures incorporate so many aspects-social, aesthetic, technical, financial, and legal, to name a few—that it is impossible for any single discipline to deal effectively with each and every one of them. Specialization has become inevitable as the only way in which the intricacies of each aspect of the building can be dealt with, and this trend will only continue to grow in the future.

Specialization makes the already involved process of building design even more intricate and time-consuming. All the individual design and construction specialists do not work together on a design. Not only are they physically located in different places, they are also not usually working on the same design model. The architects may do a preliminary design and send it over to the structural engineers and the HVAC engineers; the structural engineers may demand certain changes, but by this time the HVAC engineers may have already done their job, which they would have to repeat because of the changes. Meanwhile, the architects might have a chat with their clients and make some more changes, which, by the time are communicated to all the concerned participants might result in much wasted work. Since every project has a budget constraint, and consequently a limited time for “finalizing” the design, such delays in communication can be disastrous. Often, there are also gross errors in transmitting information, and in interpreting it. All these can lead to costly mistakes, which may not be rectifiable. Naturally, they can all end up severely compromising the quality of the design.

Over and above the communication issue, however, there is yet another serious problem that specialization brings in its wake. It is very difficult, if not impossible, for the specialists to have a clear vision of the "overall goodness" of the project. Due to their limited viewpoints, each specialist tries to optimize the design for his/her own discipline, which quite conceivably may come at the expense of other disciplines. Eliminating windows on the west side of a building to save energy is a case in point: it might also deprive the inhabitants of a fabulous view.

What is really needed to bring the various specializations together into a coherent whole is an effective system of collaboration. This calls for the development of an environment within which all the building design professionals can work together, so that there are no delays, inconsistencies, miscommunications, and other errors. This environment also has to provide a means to negotiate partial solutions, to trade them off against each other so the overall result is improved. Essentially, the environment has to specifically recognize the over-riding significance of collaboration in enhancing the quality of building design, and be geared towards actively fostering and providing for that collaboration.

It is in the development of such an environment for collaborative, multi-disciplinary building design that computers may play their most important role in the field yet. The use of computers in enhancing the quality of building design, as well as the efficiency of the design process itself, has been the subject of much research for more than three decades. In some areas, these efforts have met with remarkable success. A wide variety of sophisticated computer tools is available for design visualization in the form of drafting, 3D modeling, rendering and animation software. These have become an inseparable part of the “toolkit” of each and every one of the professionals involved in the design and construction of buildings. In sharp contrast to the widespread availability
and use of visualization software, computer tools that can provide "intelligent" assistance in the actual design process in the form of analyses, criticism, evaluation, prediction or generation, are noticeably absent. Yet, with the increasing need for collaboration in building design, such computer aids are now needed more than ever before. In fact, in today's building scenario, a collaborative design environment could be made possible only through computers, since this is the only medium both common yet powerful enough to serve as the communication channel.

This paper describes our research on such a computerized environment for collaborative building design. More specifically, it details our solution to the problem of a shared building representation for the envisioned collaborative design environment, which we see as one of the most critical components of the sought environment.

2. The Problem of Collaboration

We have seen that a building is a complex artifact with manifold aspects to it, each of which is attended to by professionals who specialize in that aspect. Not surprisingly, these professionals are concerned with quite varied sets of information about the same building. The architect is mostly concerned with spaces (i.e. voids), the structural engineer is mostly concerned with walls, columns, and floor slabs (i.e. solids), the fire safety engineer is concerned primarily with openings, and so on (Figure 1.)

![Figure 1. Multiple "viewpoints" of the same building.](image)

In fact, we might say that each one of the specialists—architects, engineers, construction managers, facilities managers, even building owners and end-users—seem to have a different "world view" of the building. In general, architects often 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 the building. Owners and end-users are usually not interested in their environment, as long as it does not impinge on their activities and interfere 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 the healing of patients).

All these unique world views have been developed by the different professionals through their education and practice. They are trained to understand the world in a particular way, and they develop a unique language to facilitate representation, communication and comprehension within their own sub-culture. Yet, all these
professionals are undeniably working on one project, and they need to think in terms of optimization of the "whole" versus the individual "parts" they are each concerned with.

The difficulty also arises when these professionals have to collaborate, as they must do. How do they communicate and exchange information with one another? How does each profession interpret and understand the information presented to it by another profession? Currently, there is no firmly established protocol of communication, much less a means to explicate that which is communicated.

2.1. REQUIREMENTS OF A COLLABORATIVE DESIGN ENVIRONMENT

Considering the problem of multiple viewpoints, the most crucial aspect of a collaborative environment would be a building representation that can be shared by all disciplines, and can become the basis of a shared understanding of the overall improvement of the project. If all the discussions, feedback loops, and design decisions related to the building could be based on one and the same representation, there would be less room for cross-disciplinary miscommunication, delays and errors. The situation would be analogous to the "ultimate" collaborative scenario where all the "experts" are in the same room, working together off the same table, fulfilling their respective roles as and when needed, making decisions and resolving conflicts as and when they arise by talking to each other, clarifying doubts and bargaining trade-offs.

Over the years, a host of stand-alone computer tools for supporting different aspects of building design, such as energy, cost, fire safety, circulation, and so on, have been developed. But few of these tools are capable of interfacing with the digital representations of the building that the different professionals prepare for the purpose of communication with each other. They each have their own unique input format which is often non-graphical and therefore very tedious, and output their results in again, a very specific format, which is difficult to redirect gainfully back into the overall design process. A building representation that is common to all the disciplines involved in design and construction could, therefore, also facilitate communication, control, and effective decision-making by bringing together all the individual discipline-specific design tools and aids into one, integrated environment.

In short, we propose that what is needed for effective computer-aided collaborative design is a unified building representation which is accessible and comprehensible to all the professionals in the building team, which not only allows the sharing of information but also the sharing of understanding, and which facilitates the development of design tools for different aspects that can be "plugged" into it.

2.2. CRITERIA FOR A "GOOD" BUILDING REPRESENTATION

How can the suitability of a proposed building representation for collaborative, multidisciplinary building design be assessed? We posit that a "good" representation will be distinguished by four important characteristics:

2.2.1. Well-formedness
The most basic requirement of a building representation is that it must have semantic integrity. Windows must be within walls rather than floating in space. If a wall is moved, all the spaces that the wall is part of must change in configuration accordingly. This calls for a representation far more semantically "intelligent" than the one, for instance, used by popular CAD softwares which are severely limited in their ability to provide adequate support for design tools.
2.2.2. Completeness
The ideal building representation would undoubtedly be one which is as close as possible to the actual building itself. This would mean that all the architectural elements—spaces, walls, doors, windows, staircases, and so on—would be represented along with all their attributes, not just individually, but also in their relationships with each other. Likewise, all the structural, mechanical, security, and other elements need to be represented explicitly. This calls for a building representation that is complete in all respects, leaving nothing to be inferred (or possibly-mis-inferred) by the users.

2.2.3. Generality
A building representation, to be of universal value, should be general-purpose enough to be able to describe any building, irrespective of its function. No doubt, it might be over-ambitious for a representation to aim to model every kind of building possible; there will always be some configuration far too complicated for it to handle. For instance, the Sydney Opera House might be difficult to represent accurately compared to another, structurally less complicated, building of the same function. Thus, it might be reasonable for a representation to target possibly 60-70% of architecture as within its capability to model, with the deciding factor being the complexity of the building rather than anything else.

2.2.4. Efficiency
An efficient building representation is one that is computationally fast and compact. Yet, these two criteria will almost always conflict with each other. Speed can be enhanced by redundancy, because the sought information will be immediately available without the need to compute or search for it. Redundancy, on the other hand, will make the space requirements enormous, and is a potential source of errors in both the creation and the manipulation of the building-related data. A good building representation has to find an efficient balance between speed and non-redundancy.

2.3. DIFFICULTIES IN DEVELOPING A COLLABORATIVE DESIGN MODEL

The main difficulty in deriving a building representation for collaborative, multidisciplinary design arises precisely due to the fact that all the different "world views" of the individual specialists have to be integrated in one shared representation. Partial models, which deal only with the domain of one specialist—such as space planning, structural engineering, or energy evaluation—are relatively easier to accomplish, and many such models are already becoming available commercially (Eastman and Siabiris 1995). However, they typically support only one or a few tasks, usually at the expense of other tasks. Thus, a model designed to support the structural analysis of buildings, for example, is less capable in the area of habitability analysis, which requires representation of intended activities more than building components. Therefore, while such "focused" models provide support for some common design, prediction and evaluation utilities for the individual specialist, they are unable to support the involved and open-ended range of design and evaluation activities that characterize the overall process of building design.
On the other hand, there are also models which are aimed at supporting a broad range of applications, but these typically sacrifice completeness for the sake of generality. A good example is a system like Autocad, which can represent almost any geometrical feature of a building, but lacks the semantic knowledge of building parts such as walls, windows, kitchens, and so on.

From the point of view of collaborative design, neither type of model is acceptable: the narrow-range but detailed models can support only a few applications, and are therefore not useful when a broad range of applications is desired. The broad-ranged but insubstantial models can only partially support all applications, lacking the ability to represent the semantics of the design. Our efforts have been focused on developing a model that can do both: have a considerable, if not exhaustive knowledge of the semantics of the designed building, while at the same time have the capability of supporting a wide and open-ended range of applications from different disciplines.

2.4. OTHER ATTEMPTS

The importance of a collaborative design environment for the effective application of computer aids in enhancing the quality of building design has, by now, become well established within the CAAD research community.

Two general directions have emerged: one which provides a low-level, shared information exchange platform, augmented by separate disciplinary models; the other which provides a comprehensive common database that can be used by all the participating disciplines. The first direction has been demonstrated by efforts such as STEP, COMBINE and IBDE, and is predicated on the belief that each discipline knows best what it needs, hence should be entrusted with developing its own disciplinary model. While this solution is, in many ways, more practical and easier to accomplish, it suffers from a lack of shared understanding, and from a reduced level of semantic communication. The other path, demonstrated by efforts such as OXSYS, ICADS, EDM and BDA, has recognized these potential deficiencies, and attempted to develop more comprehensive design databases.

In this section we briefly review some of these individual research efforts from the point of view of the four characteristics of "goodness" defined earlier-well-formedness, completeness, generality, and efficiency. A more comprehensive description of these systems can be found in Galle's recent survey of current research into computer modeling of buildings (Galle 1995).

The focus of STEP is on data exchange, rather than the actual representation of the building. In fact, it is concerned with defining a standard for the representation and exchange of "product information" in general, not just building data (Bjork and Wix 1991). Thus, STEP seems to be too general-purpose to provide an efficient and workable solution to the problem of building representation, and too shallow as far as semantic content is concerned.

The COMBINE project, working with the STEP standard, deals with a neutral file exchange of data among various application programs-primarily energy and HVAC (Amor et al 1995). It does have, at its core, an "integrated data model" for the building description, which gets translated to the applications. However, this data model was developed by synthesizing the individual data schemas of the various design tools it wanted to integrate, so it is not a complete model. Each time a new tool has to be integrated into the system, the data model has to be re-worked.
Both the OXSYS and the IBDE systems are too specialized to provide the solution to a
generic building representation. The OXSYS system was developed for the design of
hospitals and ancillary buildings in accordance with a particular style of construction
and relies on a predefined "kit of parts" to cover a large portion of the design and
construction process (Richens 1977). The IBDE system consists of seven stand-alone,
knowledge-based design tools, each dealing with a specific aspect of high-rise
officebuilding design, integrated into one system (Fenves et al 1994).

The focus of the ICADS system is not on the detailed representation of a building,
but on developing a controller for multiple rule-based expert systems that it brings
together to collaborate in evaluating a design (Pohl and Myers 1994). Only a limited
number of architectural objects are recognized by this system, so it is not very versatile.
The EDM and BDA building representations come closest to our definition of
"suitability" for collaborative design, based on the four criteria discussed earlier. EDM
was developed to address issues in the design of buildings with wide ranges of design
abstractions, construction technologies, and variations in building use. These are
represented by the constructs BOUNDED_SPACE, CONSTRUCTED - FORM, and
ACTIVITY, respectively, which in turn, form the three main components of the Generic
Building Model, the basic framework of the EDM (Eastman and Siabiris, 1995). It is
meant to be open-ended, so that additional descriptions can be added. The EDM thus
aims to be general-purpose as well as complete. As far as efficiency is concerned, it
stores both space and structure explicitly which might lead to considerable redundancy.
It is in the issue of semantic integrity, however, that the EDM differs most from our
approach. The EDM separates the geometric representation of the building from the
integrity constraints that endow it with semantic validity. The semantics are computed
as and when needed. Our model, on the other hand, aims to have as much of the
semantic integrity built into the topological representation of the building itself.

BDA (Building Design Advisor) is an integrator of a host of existing (and yet to be
developed) evaluation systems such as DOE-2, Superlite, etc. (Papamichael et al 1996).
As such, it provides a unified interface and a common data representation and exchange
format. Much effort has been devoted to rigorously define the values that are passed
from one application to another, and capture the semantics, origin, and nature of the
exchanged data. BDA does so by providing custom sockets into which the evaluation
systems plug into the BDA environment. Nonetheless, while BDA recognizes the need
for a common database, which is the repository of the values that are produced and are
required by the various applications, it assumes that the design of the building will
occur elsewhere (even though BDA provides a simple graphical editor). This
assumption is evident in the structure of the database, which seems to be more a
passive repository of objects than a dynamically updateable one. In other words, while
there is a "place" to store objects and the relationships between them, maintaining the
database's semantic integrity is a task that must be assumed by some intelligent agents
that are not part of BDA system itself. These agents are assumed to be very capable,
since the structure of the database does not, in and of itself, enforce well-formedness.

3. Our Solution

Our approach to the development of a collaborative design environment is different
from many of the research efforts just discussed. Of most significance is the fact that we
are not starting out with any specific existing application programs and developing a
building representation based on what information they need and how they need it. Instead, what we are asking is: "How should a building be represented so that it is understood as a building, so that the underlying data structure has been created to represent only a building and nothing else." We seek a building representation that can give us any kind of information about a building, from any disciplinary point of view. A few examples are:

- What spaces does it have? (architecture)
- Where are the columns and walls on each floor located? (structural engineering)
- For a given external wall, how many separate internal surfaces does it have? (energy evaluation)
- In which direction does a particular door open? (architecture and construction)
- What is the total area of wall surface? (cost analysis, construction and maintenance)

We are not concerned with exactly which specific application programs will be using this information. Rather, we would like to ensure that such information is easily available from the representation, simply because it is the representation of a building! In other words, we accept the need for specialization, but at the whole building, or project level, rather than on the disciplinary level. As such, our representation may not be suitable for mechanical engineering, or for VLSI design, as many of the current CAD systems are. In return, we gain semantic knowledge, because the representation is familiar with such terms as "wall," "window," or "HVAC system."

3.1. THEORETICAL FRAMEWORK

We present in this section, the theoretical underpinnings of our solution. First, the building representation that we have developed at the core of our proposed model will be described in detail. Thereafter, the overall schema of the integrated building environment, including the representation, will be presented.

3.1.1. Basic Requirements of a Building Representation

All buildings are essentially assemblies of similar components, such as walls, slabs, doors, windows, and so on. Where one building differs from any other is the manner in which these components are put together—the combinations are potentially infinite. In other words, it is the assembly that makes every building unique. Therefore, as far as a building representation is concerned, it seems appropriate to make a distinction between the components themselves and the manner in which they are assembled. This has the added, and very important advantage, of re-usability: the detailed information about the various components would be common across various building projects and types, and does not have to be developed or maintained for each building project separately.

The building representation, therefore, gets naturally partitioned into two logically separate components:

1. A Project database
2. Several Object databases

Project Database (PDB). This database links all the elements of the building into the appropriate hierarchy, and stores information about the geometrical location of elements with respect to each other. It is unique to each building project. Detailed information about any element can be obtained from the Object Databases. Because of the
hierarchical organization of the elements, much of the semantic content of the building representation will be inherent in the Project Database; for example, a space will necessarily belong to a floor (i.e. level), an opening will necessarily be contained within a wall or slab, and so on. Hence, it is in this component of the overall representation that the semantic integrity of the information is maintained.

Object Databases (ODBs). These databases will be commonly shared by all building projects. They will contain detailed information about every possible component used in buildings, not just for physical elements such as doors, windows, and so on, but also for more abstract elements such as spaces. The information would include material composition (for physical elements), as well as attributes, behavior, dimensional limits, and so on.

The relationship between the PDB and the ODBs is depicted in Figure 2. It shows how a PDB can draw on several ODBs, and how the ODBs can contribute to several projects at the same time.

Both these components of the building representation will now be discussed detail. The PDB is more extensively presented since it is the data structure we have developed underlying the PDB that is most unique about our proposed building representation. We believe that this data structure satisfies all four criteria of a suitable building representation listed earlier, particularly that of generality and completeness which we have seen are the criteria that other systems fail to accomplish adequately.

3.1.2. The Problem of "Completeness": Duality of Space and Structure
In developing a data structure for the representation of a building, it might be very natural to arrive at a structure similar to that shown in Figure 3, where the building has been divided into its two most basic components: the spaces that are contained within the building, and the structure of the building. Each one of these components is then expanded further. This method of organization would seem to work, except that it does
not take into account the close coupling between space and structure. For it is the structure that defines the space; alternatively, a space also determines the structure that bounds it. Thus, when either entity is explicitly represented, the other is also defined; moreover, one entity cannot be modified without affecting the other.

![Figure 3. A possible schema for building representation.](image)

A semantically-rich model, which can support multi-disciplinary design and performance evaluation of buildings, must necessarily represent both space and structure. Most commercial CAD systems represent only the structure, leaving the space it encloses to be inferred implicitly by the designers. Building models that represent both entities separately, following a representation scheme similar to that shown in Figure 3, have been developed in some research programs (e.g., EDM). Such redundancy, however, is both wasteful and a potential source for errors.

What is needed is a representation in which space and structure are both explicitly represented, but in a non-redundant, inter-dependent, complementary way. We have developed such a representation, which includes both structure and space, using a data structure modeled after the well-known, geometric, winged-edge model.

3.1.3. The Winged-Edge Data Structure
The problem of duality between space and structure in architecture is analogous to that of the duality between edges and faces in (planar) polyhedra. A polyhedron consists of a number of connected faces bounded by shared edges. Moving an edge should automatically affect both the faces that share the edge (as well as the other faces that share the corresponding vertices). Considerable research was devoted to the development of an appropriate representation scheme that could facilitate such operations efficiently. In 1972, Baumgart came up with the "Winged-Edge" data structure (Figure 4), which has since then been successfully implemented in several modeling systems such as GLIDE, BUILD, WORLDVIEW, and DESIGN (Baumgart 1972, Baer et al 1979, Kalay 1987, Kalay 1989).
This data structure is centered upon the representation of the EDGE. The boundary of a solid can be completely defined by a data structure in which each edge is referenced by two preceding edges (Epred1 and Epred2), two succeeding edges (Esucc1 and Esucc2), two faces (Face1 and Face2), and two vertices (V1 and V2). It then takes only the specification of the geometric coordinates (X, Y, and Z) of each vertex for the complete definition of the polyhedron itself. This is a compact, non-redundant, consistent, and efficient data structure.

Considering the faces and edges of a polyhedron as analogous to the spaces and walls of a building, respectively, we can use a similar data schema to solve the problem of duality of space and structure in building representation. As a simple example, consider the three-room apartment shown in Figure 5. The topological skeleton of this design is fully captured in the EDGE table, depicted in Table 1.
Table 1. The Edge table representing the building skeleton shown in Figure 5b, based upon the Winged-Edge data structure.

<table>
<thead>
<tr>
<th>Edge ID</th>
<th>Space 1</th>
<th>Vert 1</th>
<th>E-Pred 1</th>
<th>E-Succ 1</th>
<th>Space 2</th>
<th>Vert 2</th>
<th>E-Pred 2</th>
<th>E-Succ 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Living</td>
<td>V1</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td>V2</td>
<td>E2</td>
<td>E10</td>
</tr>
<tr>
<td>E2</td>
<td>Kitchen</td>
<td>V2</td>
<td>E8</td>
<td>E3</td>
<td>Outside</td>
<td>V3</td>
<td>E3</td>
<td>E8</td>
</tr>
<tr>
<td>E3</td>
<td>Toilet</td>
<td>V3</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td>V4</td>
<td>E4</td>
<td>E2</td>
</tr>
<tr>
<td>E4</td>
<td>Toilet</td>
<td>V4</td>
<td>E6</td>
<td>E8</td>
<td>Outside</td>
<td>V5</td>
<td>E5</td>
<td>E6</td>
</tr>
<tr>
<td>E5</td>
<td>Kitchen</td>
<td>V5</td>
<td>E9</td>
<td>E10</td>
<td>Outside</td>
<td>V6</td>
<td>E6</td>
<td>E9</td>
</tr>
<tr>
<td>E6</td>
<td>Toilet</td>
<td>V6</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td>V7</td>
<td>E7</td>
<td>E10</td>
</tr>
<tr>
<td>E7</td>
<td>Living</td>
<td>V7</td>
<td>E6</td>
<td>E8</td>
<td>Outside</td>
<td>V8</td>
<td>E8</td>
<td>E6</td>
</tr>
<tr>
<td>E8</td>
<td>Living</td>
<td>V8</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td>V9</td>
<td>E9</td>
<td>E10</td>
</tr>
<tr>
<td>E9</td>
<td>Living</td>
<td>V9</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td>V10</td>
<td>E10</td>
<td>E10</td>
</tr>
<tr>
<td>E10</td>
<td>Living</td>
<td>V10</td>
<td>E10</td>
<td>E7</td>
<td>Outside</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is now easy to see how more information can be added to this representation, in the form of additional tables, or pointers to attribute records. For example, it is possible to add geometry, simply by adding X, Y, Z coordinates to the vertices, as depicted in Table 2. Likewise, information can be added about the spaces and the walls, by referencing the appropriate object in one of the accompanying ODBs.

Table 2. The list of X, Y, and Z values for the vertices of the building skeleton shown in Figure 5.

<table>
<thead>
<tr>
<th>Vertex ID</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V2</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V3</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

This representation would then suffice to adequately represent the given building skeleton compactly and efficiently. Various kinds of information, related to both space and structure, can be obtained by simply traversing the tables in the correct fashion. For example, a space can be "constructed" by finding any one of its edges, and moving on to each appropriate "successor" edge until that first edge is reached again; similarly, the adjacencies of any space can be derived by drawing up its edge-list and then reading off the "other" space for each edge; all the wall segments coming together at a vertex (which could be a column) can be derived by finding any one edge of which that vertex is a part, and then checking the vertices of all its preceding and succeeding edges. Similarly, all the spaces that share an external wall can be easily located.

3.1.4. The Split-Edge Data Structure

We have found that a slight modification of the original Winged-Edge data structure enables it to adapt more effectively to the representation of walls and spaces of a building. Instead of a single edge representing a wall segment, we now have a pair of half-edges as shown in Figure 6. This is known as the Split-Edge data structure, which has been adopted for geometric modeling as well, instead of the Winged-Edge model (Kalay 1987, Kalay 1989).
In the Split-Edge model, every half-edge is associated with one preceding half-edge, one succeeding half-edge, one vertex, one face, and finally, a corresponding opposite half-edge. Translated into the building data structure, this means that each physical wall is represented by two wall segments, each of which is associated with only one space the space that it adjoins or faces. Thus the two wall surfaces are defined separately; this enables the explicit specification of the finishes and other properties of each surface information that will be crucial in evaluations such as lighting and energy consumption.

Taking the example of the same three-room apartment shown earlier, its topological skeleton using the Split-Edge Data Structure would be as shown in Figure 7.
The Edge table for this topological skeleton is shown in Table 3.

Table 3. The Edge table representing the building skeleton shown in Figure 7.

The Half-Edge table is twice as long as the Winged-Edge table, but has fewer columns, so the space requirements are only marginally larger. It has the advantage that there is now only one vertex and one space associated with an edge. At the same time, each space has its own unique set of vector lines (i.e. edges). This not only enables a more efficient way to retrieve information, but is conceptually much simpler and "cleaner". The confusion about the direction of an edge when it is associated with two spaces, as in the Winged-Edge structure, is avoided altogether in the Split-Edge structure.

It should be noted that the using a pair of segments to represent a wall is not an attempt to represent the thickness of the wall. Thus, the two segments are actually coincident; they have been shown pulled apart in Figure 7 for the sake of clarity only. The actual thickness of the wall, along with its other attributes such as U-value, finishes, and other such information will reside with the description of the wall object in the ODB which that wall references in the PDB.

The added benefit of adopting the Winged-Edge or Split-Edge concepts from geometric modeling is that both come with a full complement of operators, commonly known as Euler Operators, which maintain the semantic integrity of the data structure (Wilson 1985). These operators ensure, for example, that when an edge is split in two, a new vertex is added, and all the pointers are updated as needed.

3.1.5. The Data Structure of the Project Database
We have implemented a working version of the project database, based upon the SplitEdge data structure. It is schematically depicted in Figure 8. Although far from complete, it is able to represent a building skeleton quite comprehensively, including multiple levels (floors), as well as doors and windows.
The PDB is a relational database comprising, for now, seven "entities" related to each other as shown in Figure 8. (The backward pointers are not essential but are provided to improve the efficiency of search operations.) Where the entity is also an architectural element (e.g. space, wall, opening, etc.), it is linked to the corresponding instance from the ODB by having a field denoting the type, which holds more detailed information about that element.

It is not mandatory for a space to be bounded by a physical wall; in other words, all spaces do not have to be rooms. Two or three spaces can come together to form an actual physical "room," for example, a living+dining+kitchen in a house or an apartment. In this case, the individual spaces can be modeled as being bounded by "virtual walls"-walls that do not physically exist. However, it would be necessary for that wall to be represented in the WALL table, with a flag to indicate its type.

As far as vertical relationships between elements are concerned, one option was to apply the Split-Edge data scheme in the same manner in which it was used to capture horizontal relationships between elements. However, we found this to be complex, inefficient, and wasteful, considering the fact that most spatial relationships in architectural design are conceived of horizontally, in plan. Hence, we chose to express vertical continuity simply by storing the contiguity of vertices in the form of the vertex
below and vertex above fields of the VERTEX entity. Any other kinds of vertical relationships, between spaces for instance, can be derived fairly easily from this information.

Special situations are resolved by associating additional attributes to the different entities. For example, the nested space field of the SPACE entity can be used to represent "holes" within a space (e.g., courtyards, ducts, atria, etc.). Similarly, the next_cont_wall field of the WALL entity can be used to collate all the individual, contiguous wall parts of a single, physical wall. For example, edges E10 and E11 of the building skeleton shown in Figure 7 are contiguous parts of the same wall.

Finally, it must be mentioned that this data structure is currently 2.5 dimensional rather than fully 3 dimensional. Volumes are, for now, considered to be standard extrusions of the planar spaces with the specified height; they are not explicitly modeled.

![Figure 9. An example of an object database.](image)

3.1.6. Object Databases (ODBs)

Each one of the ODBs is an object-specific knowledge base. It contains information needed to generate and to evaluate objects of the kind it represents. This information includes:
1. Attributes, such as the object's name, its form (shape), the materials it is made of, and other properties (e.g., manufacturer, cost, etc.).
2. Classification relationships, which 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).
3. Information that describes the behavior of the object (e.g., that the kitchen is where food is prepared).
4. Integrity constraints, which 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).
5. Cases, in multimedia form, which provide anecdotal information about the object.

Objects are stored in hierarchical classification structures, which facilitate more compact, non-redundant representation—a method that has now been popularized by object-oriented programming languages. An example of a KITCHEN object and its inheritance hierarchy is depicted in Figure 9. This 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, and so on. The classification hierarchy stores shared attributes at the lowest level in the tree that is still shareable by all the objects that can benefit from this information. An inheritance mechanism makes higher-level information available to objects that are stored in lower levels of the hierarchy. Figure 10 depicts a few more of such classification hierarchies.

Figure 10. Some examples of possible object databases.
3.1.7. Integrated Building Design Environment
The building representation just presented forms the basis of an integrated computing environment for collaborative, multi-disciplinary building design, the overall scheme of which is illustrated in Figure 11. It includes the PDB, the ODBs, as well as the design and evaluation tools for the many building-related disciplines that will interface with the building representation for the purpose of query and update. We refer to these tools as "operators" or "IDEAs" (for "Intelligent Design Assistants"). Several IDEAs can come together to collaborate on one building project, and they can also partake in several other projects at the same time. We envision that one of the IDEAs can act as the "project manager" for each project. Many of the IDEAs would be similar to the existing standalone applications for energy evaluation, costing, structural analysis, and so on. However, many others would also have to be conceived from scratch to work with the new collaborative paradigm, possibly as coordinating and mediating tools between existing applications in the integrated environment.

![Figure 11. The overall schema of the integrated building design environment.](image)

3.2. IMPLEMENTATION

3.2.1. Graphical Editor
An important component of the integrated computing environment that we felt the need to develop was a CAD editor which could be used to create a graphical description of the building in the representation scheme of the Project Database illustrated in Figure 8. While this may not be as sophisticated as a full-blown commercial drafting or modeling
software, it had to be reasonably efficient and user-friendly. More importantly, it needed to have the facility to carry out a variety of semantically-meaningful operations that currently available general-purpose CAD softwares do not provide. Taking a simple example, deletion of the peripheral room of a building should automatically update the walls of the adjoining rooms from being “internal” to “external” walls.

This task has been completed in our customized editor to the extent to which a proposed building project can be quickly and comprehensively described, including the connections between the PDB and the ODBs. Consider the sample plan shown in Figure 12 which took less than 5 minutes to generate. This, essentially, is the Project Database. The links to the various Object Databases are made by associating “types” to each of the PDB entities—levels, spaces, walls, surfaces, columns, and openings. Figure 13 shows this link being made through a dialogue box that opens up by clicking on one of the wall entities.

Figure 12. The customized graphical editor implemented to describe the PDB of a building.
3.2.2. Intermediary Utilities

In order to simplify the task of modifying existing applications or developing new IDEAs to work within the integrated design environment, as well as to avoid needless repetition, we realized the need to develop a layer of intermediary tools to serve as the interface between the building representation and the various kinds of evaluation tools that will interact with it. These tools would "filter" the database and provide customized "views" for different aspects of building design such as space layout, structures, construction, energy, cost, and so on. We have implemented these intermediary Utilities at two levels:

Low-level Utilities. These extract the most basic information about the building from its implemented representation. A few examples are given below:

- The sequence of wall surfaces that make up a space.
- The area of a space.
- The two wall surfaces that make up a wall.
- The length of a wall.
- The number of doors or windows in a given wall, and their individual as well as collective areas.
- The geometrical location of a given vertex (column) or wall.
- The position of an opening within a wall and its width.
- The object specifications of any entity which has been linked up to an ODB.

Figure 13. Linking elements of the PDB to various ODBs by attaching "types" to entities.
High-level Utilities. These make use of the low-level functions to provide some higher-level information about the building. A few examples follow:

- A list of all the spaces within a particular level.
- A list of all the external walls.
- An adjacency matrix, showing which spaces lie adjacent to each other.
- A connectivity matrix, showing which spaces are connected to each other.
- The shortest path to get from one space to another.
- Contiguous vertices in multiple level building projects.
- All instances of an entity with specified object characteristics.
- Extent of visibility of one space from another.

3.3. TESTING

Two independent research interests have been supported so far by the integrated design environment described in this paper, serving as a much-needed testbed for its further development. The first was an evaluator for the spatial layout of two-bedroom apartment units; the second was an evaluation of house designs specific to the Korean lifestyle and culture. For both evaluators, the building plans to be analyzed were first drawn using the graphical editor described in Section 3.2.1. Then, the data files generated were run through the Intermediary Utilities described in the preceding section. The output from the latter process served as input to the evaluator functions.

A brief description of each evaluator is given next, along with an evaluation summary generated by it for two test examples so that a comparison can be drawn.

3.3.1. Apartment Evaluator

The attempt of this tool was to simulate the "multi-disciplinary" nature of building design by a "multi-criteria" evaluation of a design problem. Accordingly, the apartment design was tested for several criteria—optimality of space, proximity, privacy, circulation, and climatic comfort—not just at the level of the whole unit, but at the level of each individual space of the apartment. The actual evaluation of the fulfillment of these criteria was based on well-established standards of good apartment design in architectural practice. The purview of the evaluation extended primarily to performance aspects related to layout and functionality. Aspects like structure, construction and materials, cost, energy consumption, and so on, were not considered.

Evaluation Methodology. The evaluation was carried out by determining the performance measure (i.e., "quality") of each room of the unit separately, by aggregating its performance measure for each of the five aspects mentioned earlier—optimality, proximity, privacy, circulation, and comfort. This, in turn was determined by aggregating the raw scores of all the individual criteria that provide a measure of that aspect. A raw score could be either 1 or 0, depending upon whether the criterion was satisfied or not. (Ideally, many intermediate values should be possible to bring out the finer shades or subtleties of differences in performance, but for the sake of simplicity here, we dealt with just the two extreme cases.)

All of the measures for individual spaces were then aggregated with certain aspects which applied to the apartment as a whole to give its total performance measure. The aggregation, at each level, was done by assigning appropriate weights to the various criteria.
Test Example 1. The graphical representation of the plan created using the customized CAD editor is shown in Figure 14.

![Figure 14. Test Plan I for the Apartment Evaluator.](image)

The output from the Evaluator for this design, giving the results of the evaluation methodology described earlier, is reproduced in part below. (The entire listing, showing the performance criteria for all aspects, and for all spaces, is too lengthy to present here.)

<table>
<thead>
<tr>
<th>Performance Aspect</th>
<th>Weight</th>
<th>Raw Score</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living Room Quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimality of Space</td>
<td>15</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Proximity Requirements</td>
<td>25</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Efficiency of Circulation</td>
<td>20</td>
<td>30.00</td>
<td></td>
</tr>
<tr>
<td>Privacy</td>
<td>25</td>
<td>70.00</td>
<td></td>
</tr>
<tr>
<td>Climatic Comfort</td>
<td>15</td>
<td>40.00</td>
<td></td>
</tr>
<tr>
<td>TOTAL (Weighted)</td>
<td></td>
<td>89.50</td>
<td></td>
</tr>
<tr>
<td>... and so on until ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Apartment Quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality of Outdoor Spaces</td>
<td>30</td>
<td>55.00</td>
<td></td>
</tr>
<tr>
<td>Other Adjunct Spaces</td>
<td>15</td>
<td>40.00</td>
<td></td>
</tr>
<tr>
<td>Overall Circulation Pattern</td>
<td>35</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Economy and Serviceability</td>
<td>10</td>
<td>9.00</td>
<td></td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>TOTAL (Weighted)</td>
<td></td>
<td>67.50</td>
<td></td>
</tr>
</tbody>
</table>
Test Example 2. Contrast Test Plan 1 with the plan shown in Figure 15, obviously a poorer one to an experienced architect. The evaluation results (only the summary is given) reflect the difference in quality.

Figure 15. Test Plan 2 for the Apartment Evaluator.
3.3.2. Korean House Evaluator

The objective of this tool was to evaluate a given house design to determine how well it would function according to Korean lifestyle and culture (Lee 1997). The unique social and cultural factors that were consistently identified by analyzing a range of traditional and modern Korean house designs were used to define the goals and the criteria of the evaluation.

Evaluation Methodology. Korean houses are characterized by high connectivity between a central courtyard (the center of the house), and the kitchen and certain other key spaces. The evaluation of the design proceeds from the notion that the topological morphology of the floor plan expresses the social and cultural characteristics of a dwelling. This notion has been developed by Hillier and Hanson (1984) into a quantifying method that converts the topological diagram of a floor plan to numeric data. It accounts for both the connectivity of the underlying graph (called "relative asymmetry", or RA), and for the size of the floor plan (called "real relative asymmetry" or RRA). This method was employed here to generate values predicting the social and cultural performance of the house by calculating the RA and RRA values of the proposed floor plan. These were then compared to a set of ideal values, statistically calculated from prototypical house designs that were considered to be fully embodying the unique social and cultural factors of Korean lifestyle. The comparison yielded a score for several criteria, which were aggregated to give a final score indicating the overall performance of the given design.

Test Example 1. The house design shown in Figure 16 comes reasonably close to being a "good" Korean house.

Figure 16. Test Plan 1 for the Korean House evaluator.
Some excerpts from the evaluation follow. After deriving the adjacencies and connections, the shortest path from each space to all the other spaces was determined as shown below.

\[ \text{Shortest Paths between Spaces} \]

<table>
<thead>
<tr>
<th>From Space:</th>
<th>To Space:</th>
<th>Nodes</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>outside of building</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>child_room</td>
<td>1</td>
<td>3</td>
<td>1 --- 3 --- 9 --- 0</td>
</tr>
<tr>
<td>master_room</td>
<td>2</td>
<td>2</td>
<td>2 --- 4 --- 0</td>
</tr>
<tr>
<td>living_room</td>
<td>3</td>
<td>2</td>
<td>2 --- 3 --- 9 --- 0</td>
</tr>
<tr>
<td>kitchen</td>
<td>4</td>
<td>1</td>
<td>4 --- 0</td>
</tr>
<tr>
<td>guest_room</td>
<td>5</td>
<td>2</td>
<td>5 --- 9 --- 0</td>
</tr>
<tr>
<td>bedroom</td>
<td>6</td>
<td>2</td>
<td>6 --- 9 --- 0</td>
</tr>
<tr>
<td>stock</td>
<td>7</td>
<td>2</td>
<td>7 --- 9 --- 0</td>
</tr>
<tr>
<td>bathroom</td>
<td>8</td>
<td>2</td>
<td>8 --- 9 --- 0</td>
</tr>
<tr>
<td>courtyard</td>
<td>9</td>
<td>1</td>
<td>9 --- 0</td>
</tr>
</tbody>
</table>

\[ \text{Sum of Nodes of Space 0 = 17} \]

… and so on. The shortest paths, in turn, were used to derive the RA and RRA values for each space.

\[ \text{Relative Asymmetry Values} \]

- R.A. value of space 0 = 0.222222: outside of building
- R.A. value of space 1 = 0.368889: child_room
- R.A. value of space 2 = 0.305556: master_room
- R.A. value of space 3 = 0.166667: living_room
- R.A. value of space 4 = 0.194444: kitchen
- R.A. value of space 5 = 0.277778: guest_room
- R.A. value of space 6 = 0.277778: bedroom
- R.A. value of space 7 = 0.277778: stock
- R.A. value of space 8 = 0.277778: bathroom
- R.A. value of space 9 = 0.055556: courtyard

\[ \text{Real Relative Asymmetry Values} \]

- R.R.A. value of space 0 = 0.726216: outside of building
- R.R.A. value of space 1 = 1.270879: child_room
- R.R.A. value of space 2 = 0.998548: master_room
- R.R.A. value of space 3 = 0.544662: living_room
- R.R.A. value of space 4 = 0.635439: kitchen
- R.R.A. value of space 5 = 0.907771: guest_room
- R.R.A. value of space 6 = 0.907771: bedroom
- R.R.A. value of space 7 = 0.907771: stock
- R.R.A. value of space 8 = 0.907771: bathroom
- R.R.A. value of space 9 = 0.181554: courtyard

Much of the final evaluation was based on these RA and RRA values. It is summarized below.

\[ \text{Evaluation Criterion} \]

<table>
<thead>
<tr>
<th>Evaluation Criterion</th>
<th>Score</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difference of Real Relative Asymmetry (RRA) values from standard values</td>
<td>94</td>
<td>1</td>
</tr>
<tr>
<td>Hierarchy of space, evaluated by comparing Depth Values from Entrance with standard ones</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>Frequency of Contact (Degree of Communication), evaluated by Depth Values of Public Spaces—Living_Room, Corridor, Dining</td>
<td>75</td>
<td>1</td>
</tr>
</tbody>
</table>
Test Example 2. On the other hand, the design shown in Figure 17 does not perform as well.

The evaluation summary for this design, generated by the evaluator, is reproduced below.

![Figure 17. Test Plan 2 for the Korean House evaluator.](image)
4. Conclusions and Further Work

We have shown that the proposed Split-Edge data structure can serve to construct a suitable building representation that can potentially support the needs of the varied professionals involved in the design and construction of buildings. It is complete because it can represent both space and structure, it is general because the type and use of the building being modeled is immaterial, and it is well-formed by virtue of the hierarchical organization and coupling of the entities. We are optimistic that it is also efficient, but we will wait until we have tested it for a very large building project before we claim that with absolute certainty.

We have already tested the representation with two evaluation tools for architectural design that were described in this paper. These tools are, of course, just a precursor to more design and evaluation tools for other building-related disciplines that we intend to develop. We see the tool-building process as one of the best ways to ratify and further develop the underlying data structure for the PDB. To this end, we plan to invite various disciplinary experts to use our proposed representation as the basis for the development of their design aids.

At the same time, we will continue to develop further the database itself; it is still in a very rudimentary state. As we proceed further with its design development, there are several issues that need to be investigated and resolved: How can spaces come together to form specific groups or clusters, in addition to the pre-defined grouping mechanism of levels? Do floor/ceiling slabs need to be represented separately, or should they be inferred from the LEVEL entity? Should spaces on the same physical level but with discontinuous floor slabs be grouped under one level? How can inclined or even more complicated roof structures, and walls that are not strictly vertical and have irregular heights, be represented? What about columns that lie within a space rather than at a vertex—do they need a separate means of representation? How can special, but very common elements like staircases, be compactly represented? How can the data structure be made extensible so that additional information for any element can be readily added as and when needed?

Extensive further work is also required on the other components of the integrated building environment. The ODBs, and their interface with the PDB, is another substantial component of the research project. We also intend to focus on the communication and control issues, knowledge representation issues, and user interface issues as the research progresses.

Finally, we plan to use the new and powerful communication medium of the Internet within which to situate the integrated environment, and exploit this medium's potential for universal access of any information to facilitate the desired collaborative design paradigm. Accordingly, all current and future development work will be carried out in the Java programming language. Our ultimate vision is to have the building representation as well as all the desired tools and evaluators located in "cyberspace", from where they can be accessed by any member of the building team, from any place, at any time, as and when needed.

Acknowledgments

It is obvious that this project is very large, and involves many individuals, too many to list. We wish to thank, in particular, Dr. Jin Won Choi from the Department of
Architecture, Ajou University, Korea, for his significant contribution towards developing the PDB and the customized editor. Thanks are also due to all the PhD and MSc students in the CAD Research Group of the Department of Architecture at Berkeley, particularly Choong-Hoon Lee, who developed the Korean House evaluator described in this paper.

References