Generation and communication of design information: a building performance simulation perspective

Georg Suter, Ardeshir Mahdavi
Carnegie Mellon University
School of Architecture
Pittsburgh, Pennsylvania
USA

ABSTRACT

There is general agreement that the process of design and construction of buildings typically involves multiple players. This has been the impetus to develop concepts for computational environments that would support collaborative design. While there has been considerable progress with regard to hardware and electronic communication, the underlying representations of design ideas and artifacts have not kept pace with this progress. In this paper we deal with this problem not from a global conceptual perspective, but rather from the specific point of view of those designers who use design representation toward extraction and manipulation of specialized technical information. For example, engineers in various fields of building technology require a rich representation of building information in terms of geometry (with special focus on topology), materials, systems attributes, etc. We argue that the current building analysis tools do not operate on the basis of such rich informational representations. Instead the needed information is often assembled on an *ad hoc* basis from various non-integrated informational sources. We review three representations as they are implemented in commercial or research systems and explore their potential for communicating design information to computational building analysis tools. Based on this review, we describe desirable characteristics of more sophisticated building representations.

1 INTRODUCTION

One major goal in computer-aided design research has been the development of integrated design systems that allow for convenient generation and evaluation of evolving designs. While industries such as car and airplane manufacturing have had certain success with integrated design support systems, progress in the building industry has been slow. Although support for visualization and, more recently, scheduling and facility management has emerged, architectural CAD (computer-aided design) remains largely limited to drafting. Integrated design systems in the building industry have not been developed beyond the prototypical stage and so far have failed to influence how buildings are designed in professional practice.

There is no shortage of explanations for the current state, which include fragmented nature of the building industry, lack of knowledge about the architectural design process, and discrepancies between representations inside computers and designers (Bijl 1985, Akin and Anadol 1993).

In the building performance simulation domain, stand-alone analysis tools have been commercially available for some time. Unfortunately, most tools require considerable preparation and detailed domain expertise for proper operation. As a result, such tools are used by specialized professionals rather than architects. This state of affairs is unsatisfying because building performance analysis could be particularly useful in the early phases of building design. Instead, crucial decisions are typically made by architects without detailed quantitative consideration of issues that deal with a facility’s energy use, occupant comfort, ecological impact, etc.

More attention has therefore been given recently to usability issues. Research that addresses usability of simulation applications can be divided into graphical user interfaces for input support, integration of simulation tools, integration with drafting systems, knowledge-based support, and data aggregation methods for output support. Recent research projects aim at creating comprehensive integrated simulation systems and make contributions in several of the areas mentioned (Augenbroe 1992, Mahdavi 1996, Papamichael 1996).

This paper explores the potential for the integration of simulation with existing representations as they are implemented in commercial or research drafting systems. These systems appear attractive from a simulation perspective because they allow for intuitive generation and visualization of designs. Without interactive graphics, geometric information has to be specified exclusively in numerical form inside simulation tools, an error-prone and time consuming approach. We first establish informational requirements for simulation, and then evaluate the appropriateness of three existing representations with regard to simulation.

2 REPRESENTATION AND SIMULATION

Before reviewing and evaluating existing representations and their appropriateness with regard to simulation, it is useful to define what we mean by key terms such as representation and simulation.

Representation can be seen as an organized collection of data or symbols that stand for entities in another domain so as to provide relevant information with regard to represented entities. In architecture, for instance, drawings are used to represent an architect's design. Drawing symbols allow architects to conveniently share some of their design intentions with other professionals who are involved in the process of constructing a building in reality. Those symbols capture some of the relevant features of physical entities they refer to. Representation always involves abstraction because some properties of the represented entity that are not considered as important may be left out. Furthermore, the choice of a representation depends not only on the state of the world, but also on an individual's internal representation as well, which includes experience and intentions (Mahdavi 1997).

From a computational perspective, representations may be defined as consisting of "both data organization and operators suited to that organization" (Eastman 1979, p. 1). For instance, information about a space and its relations to neighboring spaces may be stored in a data structure, which is accessible by operators that may add new neighbors, modify dimensions, etc. According to this definition, the representation encompasses both the space data structure as well as the algorithms or procedures accessing that information. In the context of solid modeling, Requicha (1980) defines the term more restrictively. He refers to representations as symbol structures designating abstract solids that model physical solids. According to this view, representations are "sources of data for algorithms that compute useful properties of objects" (Requicha 1980, p. 437). The term representation is used in the following in its wider meaning.

Simulation refers to the prediction of behavior of natural or man-made things. The most popular simulation in buildings has traditionally been visualization, which is used by architects to explain to clients how the building may look like, and, more importantly, for iterative design. Due to increasing complexity of building technologies, environmental awareness and energy conservation, building modeling has been extended to include other aspects of buildings such as lighting, energy, thermal comfort, and acoustics (Mahdavi 1997). The emergence of powerful computers has made the detailed evaluation of these aspects feasible.

Representation and simulation are closely related. From a computational perspective, simulation can be interpreted as a special kind of representation that includes an artifact description and predictive algorithms that compute the behavior of certain aspects of that artifact. The generic term simulation is used in the following to refer more specifically to detailed, first-principles based building performance simulation.

3 SIMULATION REQUIREMENTS

3.1 **Description of constitutive building components**

Integrated building performance computing such as energy, lighting, or acoustics analysis requires designers to communicate building descriptions to multiple applications. Detailed computational analysis using first-principles physics is usually based on information that captures properties of three-dimensional building components. Attributes that need to be specified by the user include location, dimensions, and material.

The axonometric views of a residential house in figure 1 illustrate geometric abstractions of building entities that are frequently used for simulation. Walls, for instance, may not have a thickness, or windows may be represented as simple openings with frame or glazing areas being specified as a fraction of that area. Furthermore, secondary architectural elements such as stairs, fireplaces and chimneys may be ignored. These abstractions reduce the complexity of performance computing without necessarily sacrificing too much accuracy. Despite these simplifications, the resulting three-dimensional model of this relatively small house consists of 93 polygons and 380 vertices that define 38 walls, 43 openings, and 16 spaces.

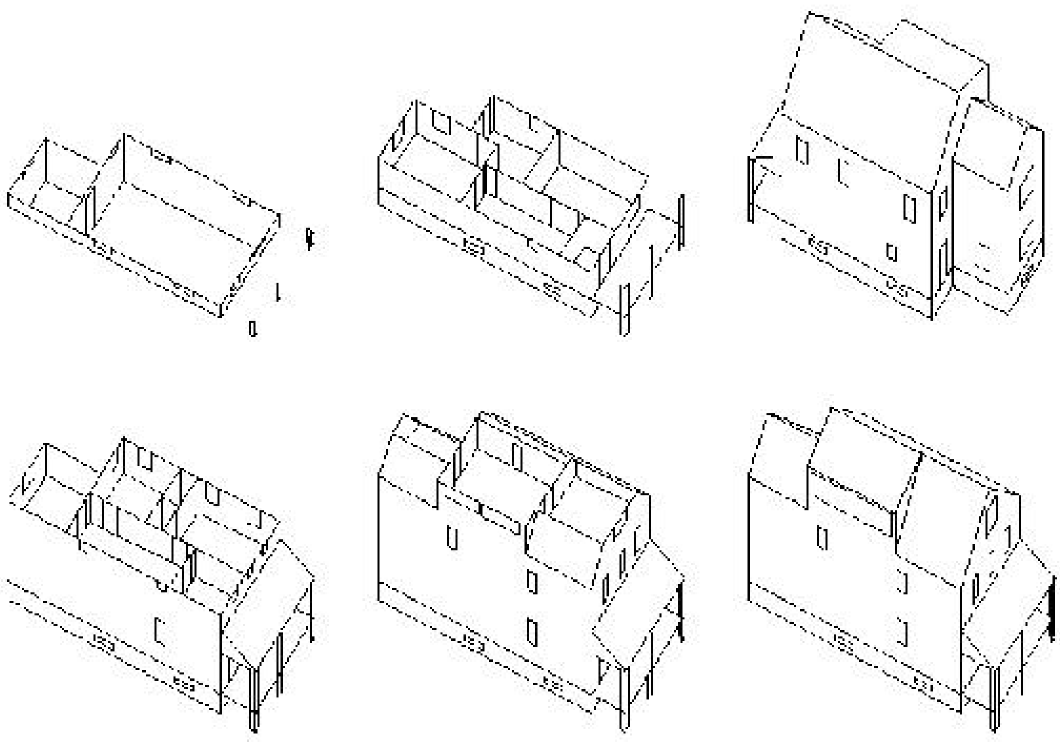
The definition of generic spaces consisting of voids enclosed by walls, ceilings and floors that form a polyhedron is useful for simulation domains to construct their internal representations (Mahdavi 1996). These representations can be computed automatically and with relative ease because they strongly resemble to a generic space-based representation. A representation for energy simulation, for instance, involves the generation of a three-dimensional cell grid that is defined by space surfaces. Furthermore, cell surfaces need to be distinguished as being either air, internal, or external, depending on whether neighboring cells belong to the same space, to a different space, or to the outdoor environment. Particularly for the last phase of generating an energy simulation representation, space related information is indispensable because it unambiguously distinguishes indoor from outdoor environment.

The concept of spaces has played a central role in the search for generic design representations (see, for example, Eastman 1979, Khemlani and Kalay 1997). A space-based representation, however, may not be convenient or necessary for structural analysis or calculation of environmental loads.

3.2 **Completeness and integrity**

Informational completeness and integrity of a representation are concepts of particular relevance for simulation. Completeness always has to be assessed with respect to the kind of analysis that

Figure 1: **Abstraction of a residential house for certain types of simulation**



needs to be performed. Requicha argues that “the importance of completeness [...] is best understood in the context of applications, where one must compute properties of represented entities” (Requicha 1980, pp. 442 - 443). A representation is considered complete in the context of simulation if it fulfills the requirements for detailed, first principles based performance simulation outlined in the previous section.

Integrity of a representation is an issue that is often ignored in drafting and simulation systems. The term integrity refers to rules that data must satisfy. Lack of integrity in data structures may cause a system failure or meaningless feedback to the user. Eastman and LaFue distinguish two basic kinds of semantic integrity relevant to design databases: existence and value integrity (Eastman and LaFue 1981). An example for the former is a wall that is defined by a starting and end point, for the latter a window that should not be larger than a certain area. Designers, the real world, or other applications may be sources for integrity rules. A building suitable for simulation is only geometrically valid if none of its elements interfere with each other, and if all spaces are enclosed by surfaces. Gaps would not be allowed either.

Ideally, a simulation tool should let the user know if a building model is flawed because it could cause the simulation to fail or, worse, generate misleading results. At present, most tools require massive and often repetitive manual input, which opens many possibilities for errors. Unfortunately, tools typically do not check, for instance, whether openings overlap, or whether each space is properly enclosed. As a result, integrity maintenance is almost entirely left to the user. In case of integration with simulation, drafting systems should relieve the user from maintaining integrity. Simulation modules could thus expect building models to be consistent.

3.3 **Support for efficient spatial queries and design development**

Completeness and integrity as outlined above could be considered as necessary and sufficient prerequisites for a representation that is useful for simulation. However, additional requirements that appear to be desirable are support for efficient spatial queries and design development.

A lighting simulation module, for instance, may require information about external walls or neighboring spaces. Similarly, an acoustics module may also need such information. Rather than having individual modules query for the information internally, a shared representation that serves as a source for various modules' internal representations could either directly incorporate frequently accessed relational information, or at least allow for its efficient derivation.

Design development support refers to the ease of generating and modifying building models from a user perspective. This issue is of particular importance in integrated simulation environments, where potentially large and complex models are created by the user.

4 ADAPTING EXISTING REPRESENTATIONS FOR SIMULATION

4.1 **Introduction**

Three typical but different existing representations are reviewed in the following: primitive-geometry, component-based, and space-based representations. The suitability of each representation for simulation is evaluated according to the requirements established above. We also outline improvements that would be needed to overcome eventual limitations preventing immediate reuse of a representation for simulation purposes.

4.2 **Primitive-geometry representation**

4.2.1 Definition and characteristics

The example of a residential house that was introduced earlier could be represented by a set of traditional architectural drawings as illustrated in figure 2. The representation would consist of geometric primitives such as lines and rectangles that give no explicit indication of what building entities they stand for. We therefore label traditional drawings as primitive-geometry representations.

Professionals interpret arrangements of shapes in drawings by recognizing certain symbols which are shared across the building community. Thus, by using their internal knowledge and experience about architectural drawings and buildings in general, professionals are able to distinguish floor plans from sections, or windows from roofs.

Since primitive-geometry representations contain no information that explicitly identifies building entities, they would be useless for simulation without some form of semantic enrichment. It has been thus suggested to develop a computational mechanism that would allow for the automated translation of two-dimensional primitive geometric entities into three-dimensional spaces, walls, and windows. The designer's role would be limited to the definition of semantic information that is not already implicitly contained in the drawings such as wall materials, glazing types, etc.

Figure 2: **Floor plans and elevations of a residential house**

4.2.2 Automated translation of primitive-geometry representations

The above scenario implies the recognition of constitutive building elements such as walls, ceilings, openings and spaces by the system similar to how humans interpret architectural drawings. Automated recognition of architectural elements from line drawings has been a research area for a long time (see, for example, Negroponte 1975, Do 1996), but robust solutions have yet to emerge. The following problems arise when considering automated geometry interpretation of two-dimensional drawings in the context of simulation:

- As mentioned above, considerable real-world knowledge is necessary to derive building elements such as walls and spaces from a collection of lines. There are ambiguities that are impossible to resolve without additional information. For instance, a rectangle contained by another rectangle in a floor plan may constitute an internal space as well as an outdoor courtyard, a distinction that is of crucial importance in energy analysis.

- Automated identification of elements alone is not sufficient for simulation purposes. A three-dimensional representation of a window, for instance, would have to be reconstructed from plan and elevation. Powerful algorithms would be needed to identify information about elements that are spread over several drawings. They would only work with the unlikely assumption that all drawings are consistent with each other.

- Designers use different drafting styles and symbols when they draw floor plans and elevations. There is no one correct way of defining a building's geometry by drawing. Instead there are loose conventions that make it almost impossible to develop robust and universally applicable interpretative algorithms. Furthermore, simulation requires abstractions of walls and openings that are most likely different from abstractions used in architectural drawings.

- Typical sets of floor plans, sections and elevations do usually not define a building geometry uniquely. Rather, designers use sections and elevations to capture important features while omitting others. In figure 2, for instance, several sections would be necessary to determine the height of internal walls and openings. In other words, many architectural drawings are incomplete from a simulation perspective.

Automated geometry interpretation appears conceptually appealing for two reasons. On one hand, it could provide an alternative to explicit user-based definition of semantic information by facilitating its automated extraction from geometric primitives. On the other hand, designers have always been comfortable with communicating and testing ideas by sketches and simple drawings, and it would therefore be desirable to have computational support that would facilitate the enrichment of simple drawings with semantic information. However, too many challenging issues are unresolved at this time to make primitive-geometry interpretation robust enough for simulation purposes.

4.3 **Component-based representations**

4.3.1 Definition and characteristics

Most commercial drafting systems nowadays support the explicit definition of three-dimensional building entities such as walls, windows, doors, floor slabs, and roofs. Material and volumetric information can be accessed by rendering or scheduling algorithms. This representation incorporates far more design information than traditional drawings. Furthermore, it allows designers to create and modify one model rather than several computationally unrelated floor plans, sections, and elevations. We use the term component-based representation when referring to this kind of representation.

As mentioned earlier, the kind of simulation that is of interest in this paper requires the definition of spaces. While virtually every commercial drafting systems supports nowadays basic entities such as walls, openings, and floor slabs, functionality for defining spaces is less common. All component-based representations that do no explicitly deal with spaces can therefore be considered informationally incomplete for simulation purposes.

4.3.2 Augmented component-based representation

Rather than relying on automated geometry interpretation to semantically enrich a representation, some commercial drafting systems implement a semi-automated procedure that allows users to define spaces (see, for example, IEZ 1995). Operations that use space information include gross area calculation and facility management. According to the procedure, a condition for creating a space is the existence of a set of walls in plan whose intersection points define at least one polygon. By clicking inside an area that is enclosed by walls, the designer activates a space creation procedure which computes a wall polygon. The boundary of the resulting space is defined by that polygon.

This sort of “after-the-fact” enrichment of an existing model with additional semantic information is illustrated in figure 3 as a series of snapshots during the generation of the first floor of the example residential house.

4.3.3 Completeness

From a simulation perspective, a serious limitation of component-based representations that implement space derivation procedures is the fact that their space definitions usually do not require the existence of floor slabs, ceilings or roofs. The only condition for successful space generation is the existence of a wall polygon in plan. While this may be sufficient for area calculations, the resulting representation is incomplete for simulation purposes.

Even when it is assumed that a designer correctly includes floor slabs and roofs in his/her model, the relationship between these elements and spaces is not explicitly stored in the representation. Algorithms would therefore be needed that would identify the relations and extend the existing representation. Similar algorithms would be necessary if the representation of a space is just a polygon without explicit relations to the enclosing walls.

4.3.4 Integrity

Integrity management in conventional drafting systems in general is rudimentary at best and to a large extent left to the user, which becomes an unacceptable burden in large and complex models. In some situations, integrity may be checked but is usually not actively enforced. The user may be informed, for instance, about windows extending beyond a wall. However, integrity checking is often incomplete in other situations involving windows, e.g., when two windows overlap.

More elaborate integrity checking would be needed for simulation. Ceilings and floors, which are not explicitly included in the definition of spaces, would have to be tested for interference and adjointness with neighboring enclosure elements. Ensuring integrity of spaces in augmented component-based representations would most likely be a computationally expensive and complex process, involving solid modeling operations on components and searches of the whole building model.

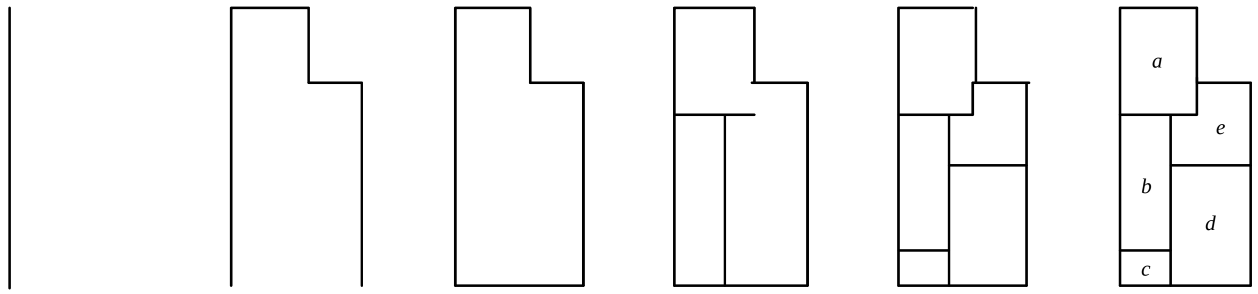
4.4 **Space-based representation**

4.4.1 Definition and characteristics

Rather than deriving spaces indirectly through identification of a region enclosed by walls that have been drawn before, a designer could also explicitly define enclosure elements and spaces in one operation. He/she could do so by drawing a polygon to define the walls of a space in a floor plan view (in the case of 2 1/2 D geometries), or he/she would explicitly define the information on the polygons that define the space enclosure. In the former case, when the polygon is closed, floor and ceiling elements can be automatically generated. Thus, a space is always ensured to be a polyhedron. We use the term space-based representation for this kind of representation.

A building representation that uses polyhedron-based spaces has been implemented in SEMPER, a research prototype for integrated simulation (Mahdavi 1996, Mahdavi et al. 1997). The simulation modules in SEMPER use internal geometric algorithms that allow them to

Figure 3: **Snapshots of wall definition sequence followed by space definition**



automatically compute their internal representations from a generic space-based building representation.

In contrast to an augmented component-based representation, this configuration is informationally complete for the class of simulation routines considered in SEMPER. Figure 4 shows a generation sequence for the first floor of the example building. Integrity is relatively easy to maintain because the bounding surfaces of a space are explicitly included in the representation and are guaranteed to form a valid polyhedron. In contrast, information about bounding surfaces in component-based representations is stored in an incoherent manner which makes integrity management much harder.

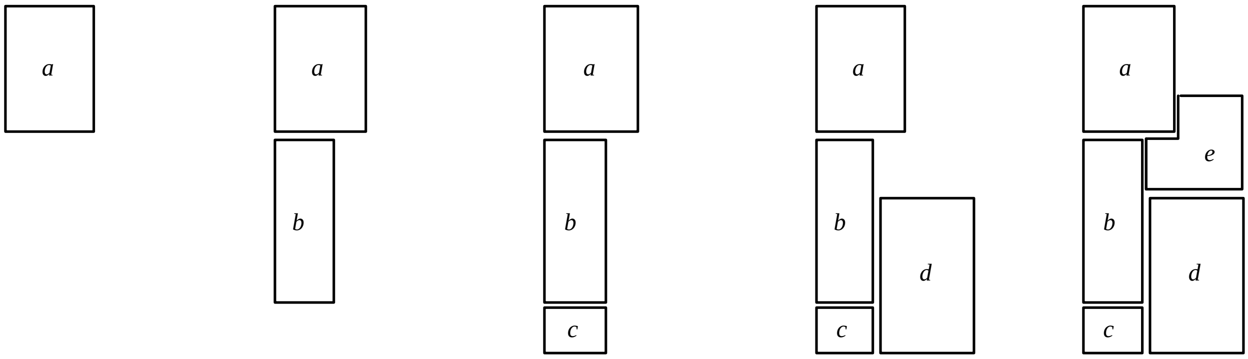
4.4.2 Limitations

This approach, however, has also some limitations. The definition of spaces currently involves a certain degree of redundancy because vertices of neighboring spaces may be visited more than once. In the generation of the first floor of the example building, 37 vertices are visited as opposed to 21 vertices when a semi-automated space derivation technique is used. Considerable accuracy may be required from the user in case of non-orthogonal configurations when snapping grids are not an option to facilitate input. Furthermore, no mechanism is currently implemented that allows designers to conveniently specify openings and non-orthogonal geometries that can be handled by the simulation modules (Mahdavi and Wong 1998). Instead, the entry of opening vertices is done manually by the user, which is a time-consuming task.

Another difficulty is the cumbersome way in which currently construction information must be assigned to building components. All internal walls, for instance, are assigned to the same global construction type. Allowing the designer to individually set wall constructions is not a trivial task with respect to integrity management. If a wall construction is changed in one space, this change would have to be reflected in all neighboring spaces that share that wall, otherwise there would be multiple definitions for the same wall element. Also, a wall may need to be split into several segments in case of so-called T shaped wall intersections.

Despite its limitations, it is important to note that this representation is the only one among those reviewed that is complete and consistent enough for most simulation purposes.

Figure 4: **Snapshots of simultaneous generation of enclosure elements and spaces**



5 SUPPORT FOR SPATIAL QUERIES AND DESIGN DEVELOPMENT

5.1 **Support for efficient spatial queries**

Both augmented component-based and space-based representations typically do not contain information about topological relations among spaces. Spaces are basically "blind" in that they do not know what their neighbors are, or whether their enclosing surfaces are exterior or interior. This kind of geometric reasoning is usually either left to simulation modules that determine these relationships on their own, or it is performed by a separate reasoning engine for the applications.

This lack of structure makes searching a building model inefficient. As a consequence, neighbor identification in conventional representations, for instance, would involve checking all spaces in a building model. This can decrease usability in various situations. Direct feedback could be slow when the designer modifies construction types for an internal wall. This operation would require a search that identifies all the spaces affected by that change. Another effect of inefficient search could be delayed feedback from simulation modules that would have to search the whole building model to extract information about relationships between enclosure elements and spaces.

In daylighting analysis, for instance, the distinction between spaces along the building enclosure and internal spaces is important. Similarly, multiple modules require information about the neighbors of a given space, e.g. for the calculation of sound transmission between two spaces. In case of multiple simulation modules with similar and frequent queries about logical relationships, representations are thus desirable in which relevant relations are either permanently stored or at least can be more efficiently derived.

5.2 **Support for design development**

An important drawback common to all building representations reviewed here is an almost complete absence of support for rapid modification and generation of design alternatives. Many operations are not scaleable, which makes generating large models prohibitively time-consuming. This state of affairs is particularly unsatisfying from a building performance modeling point of view. One could argue that the potential benefits from building performance simulation are particularly promising in the case of large projects such as office or public buildings. Integrated simulation may require a detailed description of a large number of building elements as opposed to isolated parts such as individual spaces or floors in the case of stand-alone simulation.

We are not aware of any system that allows for the generation of more than one floor, space or opening at a time while ensuring full integrity of the resulting model. The functionality provided by conventional systems hardly exploit the redundancy that can be observed in many buildings.

Similarly, existing representations offer little support for efficient modification of building models. For instance, an architect may be interested in the relationship between the massing of a building and overall energy use. This could involve the modification of a building's overall width, depth, or height. In conventional systems, the designer would have to go through a series of tedious and error-prone changes in an existing building model to change overall dimensions.

Figure 5 shows snapshots of how the operation could be achieved in the first floor of the example building with a representation in a conventional drafting system. For simplicity, only the changes in walls are shown. In addition to most walls, many other building elements may be affected on all floors, including ceilings, roofs, and openings.

Another operation that can affect many components of a model is when the designer wants to remove a space, for instance. In all representations reviewed here, this operation could result in a configuration with undesirable holes or recesses in the enclosure. In many situations the user would have to go through a sequence of manual changes to reestablish the model according to his intentions, i.e. he/she would close holes and recesses by modifying other spaces.

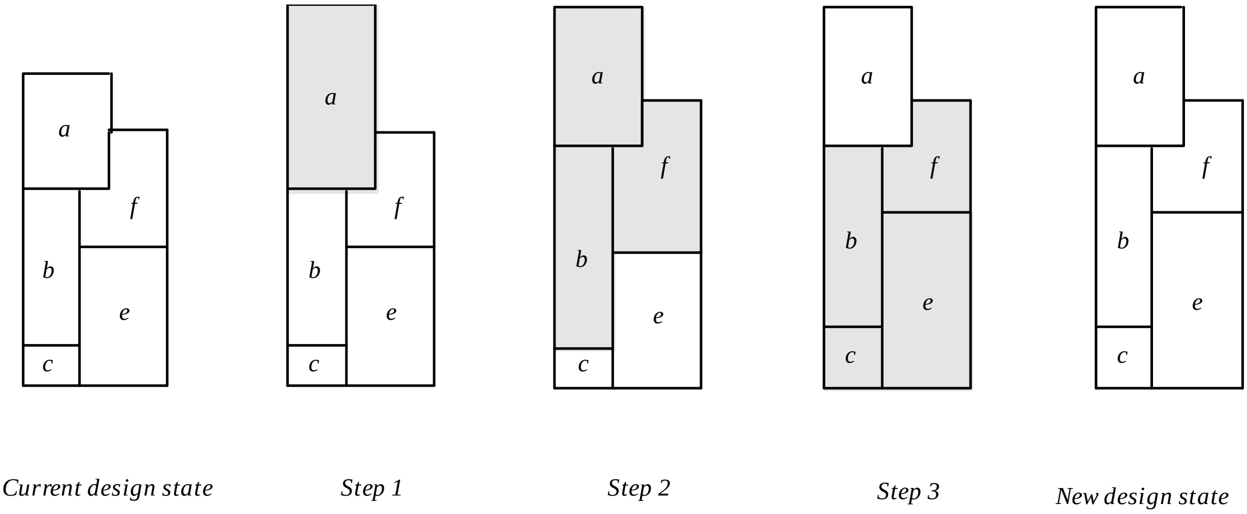
While the number of manual modifications necessary to accommodate for a change as described above is already considerable for the relatively small model of the example residential house, it would be unmanageably large in case of an office building consisting of many more entities. In general one can observe that, as a model based on conventional representations grows, its overall flexibility decreases rapidly. In all representations reviewed here, modifications of building models that affect a large number of elements are very costly. Yet, the ability to make such modifications appears highly desirable from a design development support perspective. It is conceivable to allow the designer to choose from a set of simple rules that define the geometric "behavior" of certain entities in case of change propagations. With such functionality available, modifications of overall dimensions or space deletions could be achieved in one operation regardless of the size of a building model.

6 DISCUSSION

Criteria were established earlier that are relevant for the evaluation of representations and their potential with regard to simulation and usability of simulation environments. The evaluation is summarized in table 1. The criteria include completeness, integrity, efficient spatial queries, and design development.

There are considerable differences among the reviewed representations as far as the first two criteria are concerned. The space-based representation was primarily developed with thermal simulation in mind, but it is certainly suitable for many other simulation modules in terms of completeness and integrity it provides. This conclusion is not justified in case of component based representations. Conventional component based representations are incomplete and lack

Figure 5: **Snapshots of building depth modification for the first floor (highlighted areas indicate space modifications by moving walls in the vertical direction)**



integrity to be seriously considered as primary information sources for simulation purposes. Extensive additional procedures would be needed to overcome their inherent informational limitations.

Augmenting primitive-geometry representations with automated geometry interpretation are even more problematic than component based representations. Such representations are much less complete, and integrity is much harder to maintain because information is distributed over several partial representations.

None of the representations provide much support for design development. One can of course argue that this conclusion is obvious because commercial drafting systems, for instance, were never designed to be more than drafting tools. In other areas of computational design support, this limitation has lead to the development of systems that do not rely on integration with conventional drafting systems (Flemming and Woodbury, 1995).

Many researchers in the area of building performance simulation, however, believe that integration with commercial drafting would solve most problems with regard to the communication of building descriptions by allowing the user to develop a building model with graphical means rather than numerical input. According to that position, commercial drafting systems are seen as an ideal infrastructure providing services that can readily be used for convenient communication of building descriptions by simulation tools. Lack of effective data exchange between drafting and simulation systems is perceived as the main obstacle to integration.

In contrast to that common view it is argued here that such surface level integration with conventional drafting systems is ignoring critical issues dealing with efficient spatial queries and manipulation of building models. We consider computational support in these two areas is as a necessary requirement for significantly increased usability of building simulation applications.

In conclusion, it should be stated that what conventional representations lack most in our opinion are hierarchical organization of building data and relations among similar entities such as floors, spaces, walls, windows, and doors. Many drafting systems generate the impression of building elements that are hierarchically organized. However, different levels of the hierarchy

Table 1: **Evaluation of existing representations (+ supported, - not supported)**

REPRESENTATION TYPE	COMPLETENESS	INTEGRITY	EFFICIENT SPATIAL QUERIES	DESIGN DEVELOPMENT
Primitive-geometry	-	-	-	-
Component-based	-	-	-	-
Space-based	+	+	-	-

are only loosely coupled (e.g. floors and wall elevations). Hierarchy in representations only makes sense if logical relations are actually enforced or at least checked. In contrast, more structured representations suitable for simulation would conceivably be fundamentally different from conventional representations. They would have to strictly enforce integrity and provide mechanisms for automated change propagation. Among the benefits would be simpler and more efficient algorithms for simulation queries, allowing for faster feedback, and scalable commands for the user.

Space-based building representations have paved, in principle, the way for a seamless compuational accommodation of design generation and evaluation activities. We hope that a next generation of highly structured representations with deep hierarchies and sufficient geometric complexity will further enhance the user support in view of creative designer-tool interactions toward a more productive design development process.

ACKNOWLEDGMENT

This study was partially funded by ABSIC (Advanced Building Systems Integration Consortium) and CBPD (Center for Building Performance and Diagnostics), School of Architecture, Carnegie Mellon University, Pittsburgh, PA, USA.

REFERENCES

Akin, Ö. and Z. Anadol (1993) What's wrong with CAAD?, *Proceedings Systems Research Symposium*, Baden-Baden, Germany, August 1993.

Augenbroe, G. (1992) Integrated building performance evaluation in the early design stages, Building and Environment, **27 (2)**, pp. 149 - 161.

Bijl, A. (1985) Graphical input: can computers understand people?, *Computer Graphics* **9 (2)**, pp. 85 - 95.

Do, Y.-L. (1996) The right tool at the right time - drawing as an interface to knowledge-based design aids, *Proceedings of the 1996 ACADIA conference*, Tucson, Arizona, pp. 91 - 97.

Eastman, C. (1979) The computer as a design medium, Research report, Institute of Building Sciences and Design Research Center, Carnegie Mellon University, Pittsburgh, PA, USA.

Eastman, C., and G. LaFue (1981) Semantic integrity transactions in design databases, Research report, Institute of Building Sciences and Design Research Center, Carnegie Mellon University, Pittsburgh, PA, USA.

Flemming, U., R. Woodbury (1995) Software Environment to Support Early Phases in Building Design: Overview, *Journal of Architectural Engineering* **1 (4)**, pp. 147 - 152.

IEZ (1995) Speedikon reference manual, Vienna, Austria.

Khemlani L., Y. Kalay, (1997) An integrated computing environment for collaborative, multi-disciplinary building design, *Proceedings of the 7th International Conference on Computer-Aided Architectural Design CAAD Futures*, Munich, Germany.

Mahdavi, A. (1996) SEMPER: a new computational environment for simulation-based building design assistance, *Proceedings of the 1996 International Symposium of CIB W67 (Energy and Mass Flows in the Life Cycle of Buildings)*, Vienna, Austria, pp. 467 - 472.

Mahdavi, A. (1997) A negentropic view of computational modeling, *Proceedings of the The Second Conference on Computer Aided Architectural Design Research in Asia: CAADRIA '97* (Ed.: Yu-Tung Liu, Jin-Yen Tsou, June-Hao Hou). Hsinchu, Taiwan pp. 107 - 122.

Mahdavi, A. and P. Mathew (1995) Synchronous generation of homologous representations in an active multi-aspect design environment, *Proceedings of the Fourth International Conference*, International Building Performance Simulation Association, Madison WI, pp. 522 - 528.

Mahdavi, A., P. Mathew, S. Kumar, N.-H. Wong (1997) Bi-directional computational design support in the environment, Automation and Construction **6 (1997)**, pp. 353 - 373.

Mahdavi, A. and N. H. Wong (1998) From building design representations to simulation domain representations: an automated mapping solution for complex geometries, to appear in *Proceedings of American Society of Civil Engineers Conference*, Boston, MA.

Negroponte, N. (1975) Soft Architecture Machines, MIT Press, Cambridge, MA, and London, England.

Papamichael, K., J. La Porta, H. Chauvet, D. Collins, T. Trzcinski, J. Thorpe, S. Selkowitz (1996) The building design advisor, *Proceedings of the 1996 ACADIA conference*, Tucson, Arizona, pp. 85 - 97.

Requicha, A. (1980) Representations for rigid solids: theory, methods, and systems, ACM Computing Surveys **12 (4)**, pp. 437 - 464.