INTEGRATION OF SIMULATION INTO THE BUILDING DESIGN PROCESS

Godfried Augenbroe
Dept. of Civil Engineering
Delft University of Technology
P.O. Box 5048, Delft, The Netherlands

Frederick Winkelmann
Simulation Research Group
Lawrence Berkeley Laboratory
Univ. California Berkeley, CA94720, USA

ABSTRACT

We describe the need for a joint effort between design researchers and simulation tool developers in formulating procedures and standards for integrating simulation into the building design process. We review and discuss current efforts in the US and Europe in the development of next-generation simulation tools and design integration techniques.

In particular, we describe initiatives in object-oriented simulation environments (including the US Energy 'Kernel System, the Swedish Ida system, the UK Energy Kernel System, and the French ZOOM program) and consider the relationship of these environments to recent R&D initiatives in design integration (the COMBINE project in Europe and the AEDOT project in the US).

Topics discussed include the role of simulation in building design, deficiencies of current energy performance evaluation tools, characteristics of intelligent building design systems, transfer of data and knowledge between simulation and design, and the STEP standard for the exchange of product model data.

1. INTRODUCTION

Recent advances in simulation, computer-aided design, intelligent systems, and information technology raise important expectations for future integrated intelligent building-design systems (IIBDS's).

In this paper we emphasize the critical area of simulation and its integration into IIBDS's.

Reflecting the background of the authors, we concentrate on energy-related performance evaluation, which is taken as representative of the kind of simulation that can provide needed information in the building design process.

A conceptual framework is presented that shows the necessity for a joint approach among design and simulation researchers in developing IIBDS's. This framework also shows the challenges that will be faced in establishing design links between architecture and engineering, the building design professions that, in both Europe and the US, have traditionally acted as separate, non-integrated disciplines.

The concepts and goals of several ongoing research projects will be discussed. It will be shown that these projects can have a major impact on the design systems of the future, provided that their potential in a joint approach is recognized.

To set the tone for the following sections, we give here a short assessment of what can be expected from IIBDS's.

In order to introduce some concepts let us consider design as a process in which many *actors* participate. Actors can be regarded as a generic name for anything or anybody playing a certain role or performing a certain task in the design. Actors are often characterized by the *design domain* they belong to. Design domains can be attributed to certain groups of actors, like a particular discipline, profession or building sector with particular skills inside the building industry, or embodied in a specialized department of an enterprise involved in building projects.

Actors are furthermore characterized by the set of *aspects* of the design object they consider. Typical aspects of a building are strength, durability, and cost. Aspects must be clearly destinguished from building *sub-systems*, which represent 'parts' of the building. Typical sub-systems are the building structure, a room, the HVAC equipment.

Intuitively, it is clear that an IIBDS should have two major ingredients:

- A set of design support tools under complete control of the designer. The comprehensiveness of the design tools (representing the design actors) and the flexibility with which they can exchange their descriptions of the design object determines the level of integration offered by the IIBDS.
- A system in which these tools are embedded. The way this system provides intelligent assistance in terms of when and how to use particular tools and eventually negotiate between them in case these tools suggest conflicting design options, determines the level of intelligence offered by the IIBDS.

We make the following observations:

- There is a great variety of design support tools since these tools are "tuned" to a specific design domain or goal (for example, to support presentation, specification, construction, etc.). These tools usually perform evaluations (for example, by calling specialized simulation programs) to support design decisions.
- In no way do we want to imply that IIBDS's will do "automatic" design. On the contrary, the designer will retain control over the creative process, with the IIBDS providing the information necessary to make decisions.
- The notion of a single person, a "superdesigner", at the controls of the system is by no means implied, nor is it realistic; an IIBDS would normally be used by several team members, each with individual expertise.
- We must acknowledge the fact that presently-available design and simulation tools cannot easily be integrated into IIBDS's.

We will report below on a new generation of simulation environments that can solve this problem. We also note that current CAD tools are aimed at drawing and display and thus provide very limited design support.

We will consider the integration problem from the point of view of the two basically different approaches that designers use. In the *top-down* approach, which is methodology oriented and typically used by architects, the questions asked are when and how to do what, based on what information. In the *bottom-up* approach, which is performance oriented and typically used by consulting engineers, the question is how a particular aspect or component of a building will perform.

Current projects tend to emphasize one or the other of these two points of view.

We will discuss two of these projects (COMBINE and AEDOT) in the sections that follow.

To be successful, an integration scheme must account for both approaches and provide an interface between them.

2. INTELLIGENT BUILDING DESIGN SYSTEMS

Although much work is being done to develop general design theories, a clear-cut and widely accepted theory for the very complex process of building design (Gero, 1985; Gero 1987) is still a long way off.

Among other things, such a *process model* will show the interactions that exist in the design space, which is composed of many actors (i.e, entities having a specific role in a building project, e.g., architect, consultant, design tool, software programme, etc.), as well as tasks, design stages, etc. In fact, as design tools are used within an *enterprise* (any organization with a role in a building project and typically involved in many simultaneous projects), the process model should ideally also support enterprise project relations.

In any case, process models should closely correspond to existing working methods and real *design scenerios* (coherent sets of design actions used by experienced designers). Any design system that deviates from this by imposing a rigid and unnatural way of working will find very little acceptance.

2.1 Availability of tools

At present there are very few integrated tools available, with some exceptions in limited domains, such as HVAC design (Doheny and Monaghan, 1987).

It is common practice to hire specialist consultants to whom the design context and design object are communicated. However, the form of communication usually leaves a lot to be desired. This is one of the reasons why such consultants are involved primarily in the later stages of design, when the design context is rather limited and confined. Lack of integration inhibits involvement of the consultant in the early design stages.

Consultants usually handle only a small part of a design domain. They generally use *building performance evaluation* (BPE) tools that are specific to their domain and are operated in a stand-alone mode. Without integrated tools, supervision of concurrent processes in different domains is very difficult. As a result, supervision and negotiation are inefficient, with critical decisions made prematurely or based on insufficient information.

Future design systems should be able to avoid this problem by offering easy communication between tools, guidance in using these tools, and support in supervising the design task as a whole.

2.2 R & D Approaches

Looking at ongoing R & D efforts, we distinguish two seemingly different ways to progress to the next generation of IIBDS's:

Project-driven approach:

This approach is based on a more or less preconceived scenario for a limited class of design projects (involving high-rise office buildings or super markets, for example). As a targeted class of similar projects lies at the heart of the IIBDS development, we call them project-driven.

These scenarios presuppose a flow of design actions, each of which is assigned to specific components inside the design system. The set of possible interactions is specified at the origin of the development of the design system. An example of this approach, AEDOT, is discussed in Section 6.2.

The following observations can be made:

- the top-down nature of this approach lends itself to an implementation-oriented development. In fact, most projects in this category are aimed at developing marketable software products. These products will then effectively represent the first generation of future IIBDS's.
- this approach will result in a limited level of integration because building design as a discipline confronts us with enormous challenges in the terms of number of actors and their interrelations and design intentions.

  However, recent developments in this area (Tomiyama and Yoshikawa, 1986) suggest that general design theories could (in principle) be applicable to building design, but the implementation effort required will be tremendous.
- exaggerating somewhat, one could criticise the resulting design tools (and especially the expected short term implementations) for providing little more than just some form of *parametrized design* facilities, i.e., offering only a limited number of degrees of freedom with respect to an otherwise "hardwired" sequence of design activities.

  Actual use will determine if such systems are acceptable to designers.
- in contrast with the last observation, the present trend to increased specialization of design offices (stimulated mainly by more cost-efficient and competitive 'off-the-shelf design') might prove to be the determining factor for the success of this kind of restrictive, but very efficient design system.
- the need for an open development strategy and related support of external communication is rather small. This is because the targeted design tools themselves represent closed environments since they are customized to a specific need and are limited to a single 'mini-world' view. Moreover, such a design system would be composed of specific BPE tools, selected on the basis of their capability to perform a single, well-defined task (e.g. some specific kind of simulation).

Object-driven approach

In this approach, the primary emphasis is on the complete description of the design object in order to support all imaginable communication requirements among design actors. No design process model needs to be assumed (at least not in principle), hence no restricted set of interactions are presupposed. The philosophy behind this approach is obviously less design orientated in that it targets merely an interaction tool for actors participating in a design project.

An example of this approach, COMBINE, is discussed in Section 6.1.

The following observations can be made:

- the bottom-up nature of the approach prohibits early implementation in 'closed' IIBDS's. First-generation products will primarily support easy ('friction-less') communication among design-actors.

  Medium-term enhancements could turn them into communication tools among members of design teams, enhancing the present day low-level, error-prone, and inefficient way of communicating, which is today still mainly based on the exchange of drawings.

  In the far term, interactions could be monitored, supported and even negotiated through some sort of design supervisor, which could be added as an extra actor on top of the system. Recent initiatives on coherence control and negotiation supervision provide interesting ideas in this area (Dupagne, 1991).

  Resulting (second-generation) design systems would thus be able to truly support concurrent design, clearly the challenge for the next decades.
- in the meantime, first generation IIBDS's will provide complete building descriptions in the form of a conceptual schema (i.e. a building data model) along with a physical implementation (e.g. a database to hold the data of an actual building) and interface specifications (e.g. in some neutral format) which specify how the data is actually exchanged among a broad range of actors.
- the need for open development is pre-eminent; the emerging standard for the exchange of product definitions, ISO-STEP (discussed in Section 5.2), plays a key role in guaranteeing openness for adding future actors.
- in contrast to the project-driven approach, no mapping of design activities to specific preselected analysis tools is attempted. On the contrary, taking into account the great variety of available and future tools (exhibiting many overlaps) is an important requirement for the development of the object definition, in order to guarantee its completeness, i.e. to make it a true 'image' of the real world object and thus putting no restrictions on the design activities one would be allowed to perform.

2.3 Identification of R&D requirements

In considering the R&D that is required for IIBDS's, it is useful to distinguish two different areas of integration, reflecting the two approaches described in the previous section:

- *Data Integration*: R&D in this area will lead to a standard for describing design objects and methods for making object descriptions available through a neutral format to different design domains, and within each design domain, to different

design aspects.
This is the main target of the objecty-driven approach, based on a great variety of actor-views, but providing as yet little support for interactions other than data exchange.

- *Process Integration*: this involves definition of the design context for any aspect-related task, such as performance evaluation. It also involves handling the flow of information and decisions between these tasks, between design domains, and between designers.
This is the main target of the project-driven approach, with only 'local' customized data integration, and based on a limited set of actors.

To achieve both data and process integration requires a joint approach that is initially limited in scope, with future progress based on incremental improvements.

We feel strongly that R&D should acknowledge that the key issue is the multicriterion nature of design, so that any restriction to a set of criteria specific to a particular building trade or discipline should be rejected. Also, limiting the domain is acceptable only if the domain can be clearly identified with a design specialist (HVAC engineer, for example).

There is a real danger that tools resulting from the project-driven approach will ultimately confront the design office with yet another integration problem, because short- and mid-term tools will cover only (small) parts of the design process. This danger becomes most evident when the underlying process models treat design as a sequential flow of aspect-oriented (i.e., energy, structure, layout, etc.) tasks.

For example, a design system that deals only with energy related aspects fails to acknowledge that "energy" is not a design domain, so that there is no such thing as "energy design". Rather, energy-related aspects are present in all design domains and, therefore, must be dealt with in all phases of design.

Although there are significant differences in the two different approaches there is little doubt that both will provide substantial contributions to the development of future IIBDS's. In the end, both approaches will no doubt converge to the same type of IIBDS. Initiatives from either approach can benefit from early cooperation.

A joint approach through international cooperation seems the obvious way to proceed. In sections 6.1 and 6.2 an initiative representing each approach is described.

3.THE ROLE OF SIMULATION IN DESIGN

During the design process, decisions are often made based on an *evaluation* of the design as it exists at a particular point in time. The evaluation may involve many different aspects and may serve several purposes, such as checking code compliance, verifying goal fulfilment, choosing among alternatives, and satisfying budget constraints.

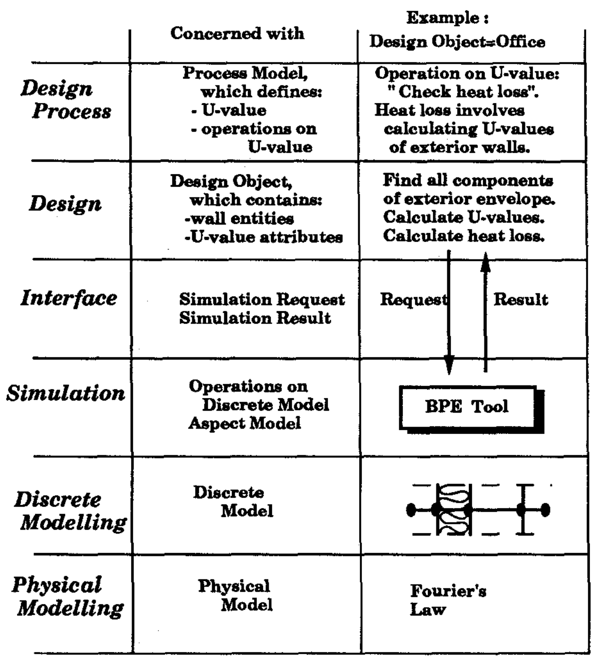
At a lower level, the evaluation may involve extensive computation, such as calculating thermal loads or energy use. Based on such "hard" evaluation, judgment, and experience, the designer will be able to make adequate decisions and proceed with the design.

Considering energy use evaluation in more detail, it is likely that a dynamic simulation of the building would be required to assess thermal performance as a function of the time-varying exterior weather conditions. In this case, the design system obviously needs to provide access to an appropriate simulation tool. *Integrated* use of this tool requires that physical knowledge from the field of heat transfer has to "migrate" into meaningful design information.

This is illustrated in Fig. 1, which shows a simple example from the envelope design domain of determining the heat loss through exterior walls.

The diagram shows the top-down migration of general design knowledge and the bottom-up migration of physical knowledge. The interface layer handles the "client-supplier" relation by providing the translation of information in the two directions.

It is interesting to note that this interface operates in present design practice mainly as a person-to-person communication via exchange of object descriptions (architect gives drawings to engineering consultant), formulation of a design-oriented request (which is cast by the consultant into a simulation input), and description of simulation output (to be translated into the design context by either the architect or the consulant). This type of person-to-person interface is generally cumbersome, time-consuming, and inefficient, and thus performs poorly in everyday practice.



	Concerned with	Example : Design Object=Office
Design Process	**Process Model, which defines:** **- U-value** **- operations on U-value**	**Operation on U-value: " Check heat loss". Heat loss involves calculating U-values of exterior walls.**
Design	**Design Object, which contains:** **-wall entities** **-U-value attributes**	**Find all components of exterior envelope.** **Calculate U-values.** **Calculate heat loss.**
Interface	**Simulation Request** **Simulation Result**	**Request** **Result**
Simulation	**Operations on Discrete Model** **Aspect Model**	**BPE Tool**
Discrete Modelling	**Discrete Model**	
Physical Modelling	**Physical Model**	**Fourier's Law**

Fig. 1. Relation between simulation and design

The general requirements of the interface are as follows:

- For *data transfer*: the interface has to map design-oriented data to BPE-oriented input. We discuss this further in Section 5.1.
- For *knowledge transfer*: the interface has to translate design requests into simulation instructions and translate simulation output into meaningful design rules.

For reasons of conceptual clarity, the distinction between knowledge (rules) and simulation (instructions) is maintained at this point. However, in actual implementations of design systems this distinction will be vague since the simulation will be an integral part of the reasoning rules. The actual simulation tasks will be carried out by running existing "external" software. This is explored further in Section 5.2.

Further research is needed. For the top layers in Fig. 1, we need better models for the design process. These models will have to provide the purpose, context, and data for specific BPE requests. For the bottom layers, research on the integration of BPE tools and a methodological approach to validation in integrated environments are required. Also, additional research in needed to produce better simulation of physical process interactions (such as the coupling of the building envelope and the HVAC system and the coupling of interzone air flows and thermal loads).

4. ENERGY-RELATED BPE TOOLS

4.1 State of the art

Over the last 15 years, hundreds of energy-related BPE computer programs have been written for such applications as thermal comfort analysis, energy use calculation, HVAC equipment sizing, and lighting analysis. The spectrum of modeling approaches in these programs is quite broad. At the bottom range in terms of complexity are simplified methods that use fairly rough information about a building, such as the overall thermal conductance of the envelope and the number of degree days, and give a correspondingly rough indication of the performance of the building, such as annual heating load. In mid-range are programs that perform a quasi-steady-state hourly thermal calculation under actual weather conditions using transfer function or finite difference techniques.

At the upper end are very complex, detailed programs exemplified by finite element methods for the solution of the Navier-Stokes equations for natural convection and component-based programs that calculate the minute-by-minute dynamics of HVAC systems by iterativey solving large sets of coupled differential and algebraic equations.

4.2 Deficiencies of current tools

Despite the range and power of the current-generation BPE tools, their use in building design practice has been very limited. For example, a 1987 survey by the American Institute of Architects (AIA, 1987) showed that only 10% of architectural firms in the US use BPE software. A similar situation exists in Europe. A number of reasons for this low level of use can be identified:

- The programs are hard to learn.
- The input, particularly geometric information, is difficult and time consuming to prepare.
- Each program has its own particular input and output format so that using more than one program on a project is particulary frustrating.
- Most programs require detailed input data, which makes them hard to use for early design (which is when energy-related design decisions are most important).
- Program output is hard to interpret and is often too sparse or too voluminous.

Other deficiencies common to most BPE tools that affect their reliability and extendibility are the following:

- The programs are non-modular ("monolithic"), with calculation methods closely intertwined with data structures; this makes them difficult to enhance, even for the original developers.
- Their parts are not easily reusable; a routine from one program can rarely be used in another program without extensive rewriting.
- There are no standards for testing and validation.
- It is usually impossible to determine the accuracy of a program for a particular design application; as a result programs are often misused.

There are also deficiencies of a more general nature that prohibit the straightforward integration of these simulation tools into design systems:

- The simulation language lacks expressiveness, which prevents an adequate translation of design-oriented requests into input to the simulation tool.
  Until recently there have been no attempts to develop flexible, modular, externally configurable simulation environments based on expressive simulation languages (see Section 6.3).
  Because current tools use predefined solution paths from a numerical problem statement to a numerical result, they can't directly handle most design requests, which are generally "inverse" (What should I do to get the desired answer?), "interrogative" (Why is this not what I expected?), or "incremental" (Do as before but slightly differently).
- There is a lack of expressiveness in describing the object being simulated. For example, most tools allow only a limited set of geometries and topological structures, which makes it difficult to map real design objects to simulated objects.
- There is a lack of explicit knowledge on how to use the tool. BPE tools require expert knowledge to translate the design request into proper input. Unfortunately, only part of this knowledge is explicitly available; the rest depends heavily on experience or creativity, or is hidden inside the tool in the form of the particular mathematical models and algorithms used in the simulation.

4.3 New developments

Near- and long-term efforts are under way to address these deficiencies.

In the near term, BPE tools are being linked to computer-aided design and drafting (CAD) systems to simplify input of geometric data; interactive front-ends are being attached to programs to speed learning and data input; increased use is being made of graphics to assist in results interpretation; and BPE tools are being linked to expert systems as a first step towards incorporating domain-specific knowledge in the simulation process.

However, the majority of these developments are rather mono-disciplinary in nature (i.e., bottom-up), so that they fall short of the desired design system discussed in Section 2.

Although the products resulting from these efforts will increase the efficiency of specialized consultancies, a dramatic change in the present low level of use of these tools in architectural design cannot be expected.

There is, however, a long-term effort - the development of "object-oriented simulation environments" - that addresses the expressiveness deficiencies of current tools. These environments, which are discussed in more detail in Section 6.2, will produce the next generation of "user-friendly" BPE models and will facilitate the integration of these models into intelligent design systems.

5. INTERFACE REQUIREMENTS

5.1 Data Transfer

A building project requires generating, updating and communicating an enormous amount of data. Formally, we call the complete set of data about the design the *design object description*. This description includes the topology and structure of the object, along with information that is relevant to particular tasks or participants in the design process. Such information spans data about costs, manufacturing, function, strength, color, tolerances, etc.

Traditionally this description is stored and displayed in analog, segmented, and unstructured media, causing numerous problems due to the ambiguity, incompleteness, and inconsistency of the information. There is a strong consensus in the computer industry that the key to integration will be the definition of complete data models for each product type that will satisfy all of the above information needs. A major effort in this direction, the formulation of the STEP standard, is described in the next section.

This complete data model is generally called a *product model*. Although we are mainly concerned with design in this paper, it must be realized that integration based on product models reaches beyond the design stage, spanning the entire life cycle of the product - design, construction, and operation.

Present product modelling efforts reflect this broader, Computer Integrated Manufacturing (CIM) scope.

5.2 The STEP standard

Since 1983, subcommittee TC184/SC4 of the International Standards Organization (ISO) has been working on a standard for the exchange of product model data (ISO/STEP). Other efforts in the same area, e.g. PDES (in the US) and CAD*I (in Europe), have produced substantial input to the STEP efforts, but will not be discussed here.

At present, the first version of STEP is about to become an ISO draft standard (ISO, 1989).

STEP's main target is the exchange of multiple representations of the design object between computers. These representations are critical in integrated environments since each domain, aspect, and particular simulation tool requires a different representation of the same object.

Since these representations are so diverse and are often particular to just one tool or even one software module, we need to decide on the level we want the data integration to take place, i.e., which views are to be accomodated by the product model. Since our focus is primarily on building design systems, we at least have to accomodate the views of those actors that constitute the main agents of the intended design system.

We will refer to the different "views" of the object as *aspect models*. The key to providing a standard that permits an exchange of data is the definition of a central and complete product model, which serves as a reference from which all aspect models can be derived (Turner, 1988; Reed, 1988).

Figure 2 shows how this is accomplished.

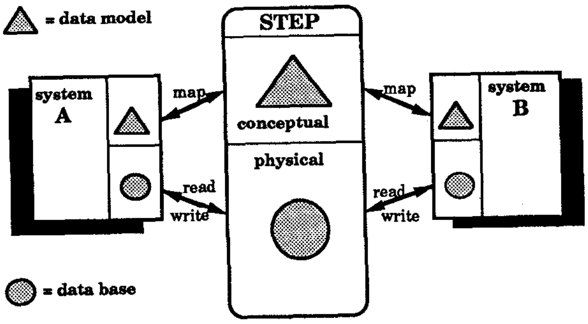


Fig. 2. Data exchange between two systems using a product model standard.

The data model is specified in two layers, a *conceptual layer* (schema) and a *physical layer* (file or database).

The conceptual layer is the exchange reference; it serves as the basis for implementing the physical layer (the neutral format for the storage of data), making possible the actual exchange of the product model data.

One should note that although all exchange is logically supported by the 'central' conceptual schema, the actual data transfer need not take place through a central data store. In fact, the actual transfer of data can be accomplished through one of several solutions, e.g. neutral file (ASCII), shared database, or distributed database. In the latter case data could physically reside with local actors, but the conceptual schema would have to 'know' about storage locations.

In Fig. 2, systems A and B might be two different IIBDS's.

If both systems were (independently) developed with the awareness of the emerging STEP standard, there is a fair chance that the exchange of object representations between the systems would be possible without much loss of information content. However, it is to be expected that the standard will be so huge that a complete, one-to-one mapping between systems will be unattainable in practice.

The standard will, however, enable the specification of the adherence to a particular subset of the standard (to be specified for system A and system B by their respective developers). Obviously both systems would have to supply a STEP *translator* to make the actual data exchange work. In this way the two systems (addressing different design domains, for example) could be easily integrated.

The requirements stated above have been given the most emphasis in the object-driven approaches.

The present situation is in sharp contrast to this picture. Current systems use their own conceptual data model, confined to the limited scope for which the system was designed. As a result, there are few "overlapping" entities in the three different conceptual models belonging to A, STEP, and B in Fig. 2, which makes a mapping from A to B virtually impossible.

We offer the following observations about STEP:

- as yet, STEP has little to offer when it comes to complete building-data models. Nevertheless some contributions to STEP, such as the general reference model, GARM (Gielingh, 1988) , can provide useful concepts for the building product definition effort.
- STEP is presently as much a research effort as it is a standard.
- recent developments focus on a general framework for product definitions, distinguishing between different levels: definition, representation, and presentation. This framework is an important step for future product type definitions.
- STEP has produced a number of working methods (such as EXPRESS for schema definition, neutral file format, and EXPRESS parsers) that provide a useful toolkit for further progress.

It is recognized that the results from the object-driven type of R&D can make important contributions to the international STEP effort.

5.3 Knowledge transfer

For a design system to be both integrated and intelligent it must offer more than just data exchange capabilities between its components.

In fact this is an area where the project- and object-driven approaches show significant differences:

Project-driven approach: the design scope and related design intelligence are defined from the outset.

R&D efforts center around AI-based components that supervise and accomplish all data and knowledge exchanges. AI developments like frame-based knowledge bases and blackboard communication form the central core of the 'tightly coupled' software system.

Object-driven approach: deliberately, no attempt is made to define a design scope from the outset. The focus is on sharing the object description among an open set of loosely coupled components. Yet, even if there is no ambition to provide central supervision of the design evolution, meaningful support of design actions performed by different actors on the same object obviously requires more than just data transfer.

One of the great challenges is to determine what 'knowledge' should (and can) reside in the object description itself and what type of intelligence must be added to provide real design-contextual exchange between two actors.

Moreover, we need to look closely at the requests that one should be able to handle since they determine to a great extent the intelligent support for the relevant data to be transferred, e.g.:

- "Tell me eveything you know" (requires huge amounts of data)
- "Tell me everything I need to know" (purpose-context must be added)
- "Tell me everything I don't know" (purpose- and history-context must be added)

An idea of how data transfer could be supported is shown in Fig. 3. The situation depicted in is one typically encountered in object-driven R&D prototypes. Most functions performed by the application interface are, however, generally valid.

Since we are especially interested in the interface between design and simulation, we will take a closer look at the type of knowledge that is required to support the functions of that interface (Rogier and Tolman, 1989).

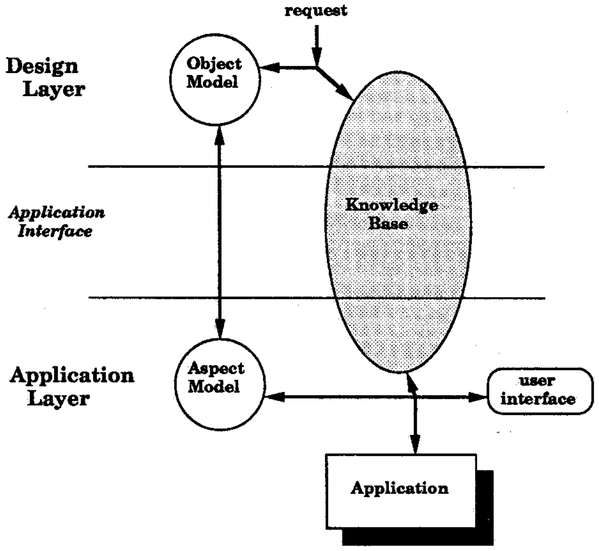


Fig. 3. Functions of the application interface supported by the knowledge base.

Assume that at a certain point in the design session a request for information (issued through the user interface of the IIBDS) is interpreted as a request that a particular application be invoked. The knowledge rules in the design layer should then be able to carry out the following tasks:

(1) Check the validity of the request and choose the appropriate application.
(2) Inspect the object model to determine whether it contains the information required by the application.

If no errors are detected the request is sent to the application interface, which should then:

(3) Translate the object model into the aspect model for the application.
(4) Invoke the application's user interface, which prompts the user for additional input and selective actions when appropriate.

Through combined support from the application interface and the IIBDS interface, the following takes place:

(5) Configure the application tool to carry out the desired simulation (through generation of input data or simulation instructions).
(6) Run the application (interactively, under user control)
(7) Add the resulting output data as "extensions" to the aspect model.

If the user exits the application layer, the following tasks will be carried out by the application interface:

(8) Inspect the extensions to the aspect model; translate them and then add them to the object model.

Knowledge rules in the design layer must be able to interpret the new data in the object model and provide the designer with context sensitive responses to his original request. We note that the above tasks involve knowledge rules for the design process (tasks 1,2,8) as well as for modelling and simulation (tasks 2,3,4,5).

We make the following observations:

- BPE tool developers will be required to express the modelling and applicability knowledge of their tools. They will have to adopt standardized formats for this purpose (Laret and Dubois, 1988; Sahlin and Sowell, 1989).
- BPE tool developers will be required to define aspect models for their tools.
- Tasks 5,6,7 will be greatly enhanced by future object-oriented simulation environments (Section 6.3).

6. RECENT R&D INITIATIVES

6.1 The COMBINE project

Following an initiative taken at the Commission of the European Communities (CEC) workshop on the future of building energy modelling (Hattem, 1987), it was decided that the CEC would fund work in the area of integrated design systems. The funding would be allocated within the scope of the JOULE program, which addresses energy-related research. COMBINE (Computer Models for the Building Industry in Europe) was chosen as the (very broad) name for this effort. Prior to the call for proposals, a definition study was carried out (Augenbroe and Laret, 1988).

Based on a number of selected proposals, the COMBINE project was started in mid 1990. The present phase is scheduled to end in the Fall of 1992. Fifteen groups from eight European countries are participating.

COMBINE is a typical example of an object-driven R&D project, and as such makes no attempt in its present phase to target an IIBDS. Rather, the research will attempt a first step towards integration of design actors into future design systems. Short term efforts will concentrate on those actors that deal with energy related BPE tools.

A number of software prototypes of design actors will be developed; these so-called DTP's (Design Tool Prototypes) will build upon existing as well as new BPE tools, adding local intelligent support to their use.

DTP's will show a variety with respect to

- the level of local intelligence they provide;
- the level of expressiveness of the BPE fuctions they perform (one of the DTP's will use an object-oriented simulation approach);
- the way in which the data transfer is accomplished (file exchange vs. shared database);

The COMBINE project is expected to result in:

- A tested and validated approach to the interface issues regarding the incorporation of BPE tools through design actors in current and future building design systems.
- A common data model for several design actors and several geometrical modellers.
  This model will be conceptually close to the emerging STEP standard discussed in Section 5.2.
- Implementation of a common data model in a suitable software environment.
- A first step in creating common goals and working procedures among BPE research groups, and between BPE groups and design research groups.
- A prototype of a limited integrated design system with emphasis on easy data exchange.

Ensuing phases of the project will gradually move towards a more project- driven approach by adding a variety of other design actors and by providing more design process knowledge and supervision.

6.2 US Integrated Design System Initiative

In the U.S. work has begun on a 10-year effort to produce AEDOT (Advanced Energy Design and Operation Technologies), a computer-based system for the design, construction, and operation of energy-efficient buildings (Brambley et al., 1988). This project is sponsored by the U.S. Department of Energy and is being carried out by Pacific Northwest Laboratory, Lawrence Berkeley Laboratory, California Polytechnic Institute (San Luis Obispo), and the University of Oregon.

The attributes that are being aimed for in AEDOT are: provides design advice and guidance; supports iterative nature of design; supports multidisciplinary nature of design; useful throughout design process, from conceptual design to construction drawings; useful throughout building life cycle, construction, commissioning, and operation; accommodates different users (architects, engineers, etc.) and different skill levels; addresses both quantitative and qualitative aspects of the building.

Figure 4 shows a schematic of AEDOT, which will be built upon the ICADS (Intelligent Computer Aided Design System) software (Pohl and Chapman, 1988).

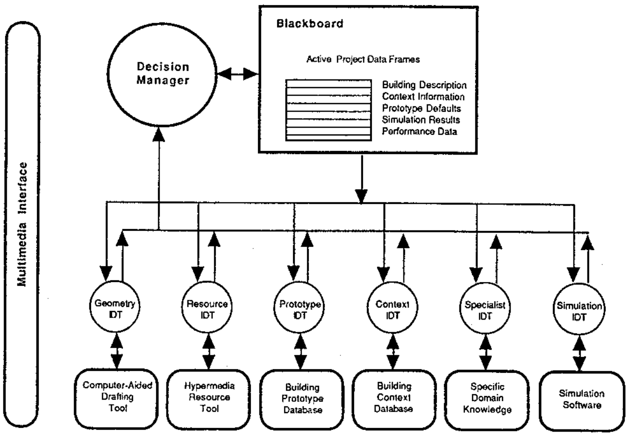


Fig. 4. Schematic of the proposed Advanced Energy Design and Operation Technologies (AEDOT) software in which simulation is integrated into an intelligent building design system.

The key elements are the following
- The "Decision Manager" provides the user with a primary link to databases and application tools, and manages the history of the project. Its functions are to store the history of the design, create a file of design goals and criteria, guide the user to relevant design prototypes and case studies, and determine conflicts between goals and the currently-active building model and suggest options for resolving conflicts.
- The "Blackboard" is a message center that holds the currently-active building description in a frame-based, object-oriented database; it serves as a conduit between all system components.
- "Intelligent Design Tools" (IDTs) are expert systems that process information that is exchanged between the Blackboard (under control of the Decision Manager) and the databases, software tools, domain-specific knowledge, and simulation programs.
- The "Multimedia Interface" allows users to take advantage of different media for receiving information from AEDOT, including text, graphics, voice, and video.

Initially, AEDOT will be restricted to a particular domain (e.g., building envelope design). Later, AEDOT will be expanded to handle all aspects of building design and operation.

6.3 Object-oriented simulation environments

As we have already pointed out, the monolithic, non-modular nature of current-generation BPE tools makes them extremely difficult to adapt to the future needs of designers.
However, such adaptation is crucial in order to provide users with up-to-date models that can simulate advanced building components and HVAC technologies, and that can take advantage of the improved solution techniques and user interfaces that will make the programs more robust and easier to use. To overcome these difficulties, model developers have been investigating new methods of structuring simulation programs. Out of this has emerged the idea of object-oriented simulation environments in which models of arbitrary complexity can be built by linking together calculation objects. Four such environments that are under development in different countries are:
- in the US: the US Energy Kernel System (EKS/US) (Buhl et al., 1990)
- in Sweden: Ida (formerly MODSIM) (Sahlin, 1988)
- in the UK: the UK Energy Kernel System (EKS/UK) (Clarke, 1988)
- in France: ZOOM (Bonin et al., 1989)

In EKS/US and Ida the calculation objects are differential and algebraic equations that describe physical processes. In the EKS/UK, objects are algorithmic procedures extracted from existing programs. In ZOOM, objects are "cells" (spatial domains or physical components) and "transfers" (quantities that can be exchanged between cells).
Although the structures of these systems are quite different, they share a number of common features:
- A processor links calculation objects together to form simulation models.
- According to the standard object-oriented programming paradigm, the methods and data associated with a calculation object are encapsulated; i.e., they are internal to the object and cannot, in general, be altered by other objects.
- Classes of objects can be defined, then instantiated to create particular instances of an object for use in a simulation model.
- Small objects can be assembled in a hierarchical fashion into larger objects (macro-objects or submodels).
- Objects and macro-objects can be stored in a library.

Such simulation environments provide several important advantages relative to traditional methods of program development:
- Depending on the objects selected and how they are linked together, a broad spectrum of models can be assembled, ranging from simplified methods appropriate to early design, to detailed methods appropriate to final design.
- Objects can easily be added to a model, and the internal calculation of an object can be modified without "knock-on" effects in the rest of the model. These features make models easy to upgrade and extend.
- Objects can be reused at a later time for building other models.
- Objects can be shared among different simulation environments if they are expressed in a standard form, then translated for use in a particular environment. To accomplish this, a "neutral model format" for calculation objects has been proposed (Sahlin and Sowell, 1989).

Because of their modularity and flexibility, these simulation environments have the potential to facilitate the integration of simulation into intelligent design systems. Models appropriate to the time, domains, and aspects coordinates of design can be created with a simulation environment and then incorporated in an IIBDS. Alternatively, the simulation environment itself could be imbedded in the IIBDS, so that application models tailored to the design questions as they emerge could be generated "in real time", executed to provide answers, and then saved for later use or released at the end of the design session. Whichever model-creation approach is taken, the potential of object-oriented simulation environments will only be realized if the developers of these environments and and the developers of design systems begin to work together now to formulate the mutual specifications and protocols that will allow this integration to proceed naturally and efficiently.

7. CONCLUSIONS

Based on the awareness that only a small fraction of the buildings that are designed today undergo an energy performance evaluation, we have argued that the next generation of design-support software should offer the designer easy access to these evaluation tools in integrated design systems.
We have suggested that current design system development efforts fall in two categories, each having significant merit for the eventually converging initiatives towards future full-blown IIBDS's.
Apart from this difference in approach, we have suggested that the successful development of any such design system will require a "top-down" effort by the design community to define real-world design process models, and a "bottom-up" effort by simulation researchers to refine and validate their calculation models and to develop flexible simulation environments that will facilitate integration of these models into design systems.
We have stressed that a joint approach, both between design and simulation community as well as between the project-driven and object-driven developments, joining forces and expertise, will be necessary.
In particular, strong links with ongoing efforts to develop new object-oriented simulation environments will have to be initiated.
We feel that organizations like the International Building Performance Simulation Association (IBPSA) in North America, the Building Environmental Performance Analysis Club in the UK, or a future European organization have major roles to play in establishing these links.

REFERENCES

AIA, 1987, "AIA Firm Study Profiles U.S. Design Market", Building Design and Construction, November 1987.

Augenbroe, G. and Laret, L., 1988, "COMBINE, Pilot Study Report", CSTB Report Presented to the Commission of the European Communities.

Bonin, J.L., Grandpeix, J.Y., Joly, J.L., Lahellec, A., Platel, V., and Rigal, M., 1989, "Multimodel Simulation: the TEF Approach", Proceedings, European Simulation Conference, Rome.

Brambley, M.R. et al, 1988, "Advanced Energy Design and Operation Technologies Research," Pacific Northwest Laboratory report no. PNL-6255.

Buhl, W.F. et al., 1990, "The U.S. EKS: Advances in the SPANK-based Energy Kernel System," Proceedings, Third International Conference on System Simulation in Buildings, Liege, and Lawrence Berkeley Laboratory report no. LBL-29419.

Clarke, J.A., 1988, "The Energy Kernel System", Energy and Buildings, Vol. 10, pp. 259-266.

Doheny, J.G. and Monaghan, P.F., 1987, "IDABES: An Expert System for the Preliminary Stages of Building Energy Systems", Artificial Intelligence in Engineering, 1987, Vol 2, no 2,pp 54-64.

Dupagne, A., 1991, Private Communication.

Gero, J.S., 1985, "Knowledge Engineering in Computer Aided Design", Proceedings, IFIP WG 5.2 Working Conference, Budapest.

Gero, J.S., 1987, "Expert Systems in Computer Aided Design", Proceedings, IFIP WG 5.2 Working Conference, Sydney.

Gielingh, W., 1988, "General AEC Reference Model (GARM)", Proceedings, CIB Seminar on Conceptual Modelling of Buildings, Lund.

ISO - TC184 / SC4, 1989, "External Representation of Product Definition Data", ISO Status Report.

Laret, L. and Dubois, A.M., 1988, "Modelisation et Simulation de Fonctionnement Thermique des Batiments et des Equipements Energetique Associes", CSTB Report MGL/88-1031/NB.

Rogier, J. and Tolman, F., 1989, "De Formulering van een Ontwerpparadigma voor bouwkundige Ontwerpers", Proceedings, Cape '89, Amsterdam.

Reed, K.A., 1988, "Product Modelling of Buildings for Data Exchange Standards; from IGES to PDES/STEP and Beyond". Proceedings, CIB Seminar on Conceptual Modelling of Buildings, Lund.

Sahlin, P., 1988, "MODSIM, a Program for Dynamical Simulation of Continuous Systems", Proceedings, The 30th Annual Meeting of the Scandinavian Simulation Society.

Sahlin, P. and Sowell, E.F., 1989, "A Neutral Format for Building Simulation Models", Proceedings, Building Simulation '89, pp. 147-154.

Buhl, W.F. et al., 1990, "The U.S. EKS: Advances in the SPANK-based Energy Kernel System," Proceedings, Third International Conference on System Simulation in Buildings, Liege, and Lawrence Berkeley Laboratory report no. 29414.

Tomiyama, T. and Yoshikawa, H., 1986, "Extended General design theory", CWI Report, No. CS-R8604, Centrum voor Wiskunde en Informatica, Amsterdam.

Turner, J.A., 1988, "A Systems Approach to the Conceptual Modelling of Buildings", Proceedings, CIB Seminar on Conceptual Modelling of Buildings, Lund.