Building and Environment, Vol. 27, No. 2, pp. 149–161, 1992.
Printed in Great Britain.

0360-1323/92 $5.00+0.00

#6123

Integrated Building Performance Evaluation in the Early Design Stages

GODFRIED AUGENBROE*

A general framework for the development of future Intelligent Integrated Building Design Systems IIBDS is discussed. After introducing a general integration framework, both from a process as well as from a product view, it is argued that a key-requisite for development-strategies towards full-blown design systems is an open, conceptual approach, avoiding premature excessive implementation efforts and adhering closely to (emerging) standards, such as STEP. Results from the European R&D project COMBINE (Computer Models for the Building Industry in Europe), jointly carried out by 15 partners from 8 countries are presented. COMBINE deals primarily with data integration in the early design stage, the 'design actors' being both from the architectural (plan-layout in sketch design phase) as well as consultancy (energy performance and HVAC) disciplines.

1. INTRODUCTION

RECENT studies have shown that technological innovation is of the greatest importance if the building industry is to meet the requirements of a technologically advancing society.

In European countries, in the traditionally craft-based and rather conservative construction sector there is a growing realisation that technological innovation is the most appropriate way to catch up on some advancements that are apparent on external markets. Such innovation promises to be the most cost-effective way to regain and strengthen a competitive position of European building industry, both within the internal and in the external markets.

The economic importance of the issue can be appreciated through the realization that, on average, the building sector constitutes 11% of the gross national product of the EC member states; it is the Community's second largest industry and employs 8% of the working population.

There is an increasing conviction that use of computers during the design and construction stages is a vital means in achieving techno/economically optimal buildings and furthermore in supplying the appropriate client support and service during the operation of the building.

Moreover, we see new demands and requirements entering the market, either through government codes and regulations or through higher client procurement standards. These include increased demands on energy and maintenance efficiency, maximum flexibility and adaptability of buildings, higher quality standards (to be expressed in quality indexes or certification of buildings), smart buildings, guarantees against sick building syndrome, etc.

Furthermore, in many countries we see a shift in construction activities from new buildings to urban renewal projects.

The flexibility required to respond to these changes in products, support and organizational structures will be best offered through a rationalization of industrial processes, much of which will be based upon new information technology. We can distinguish some general trends in technological innovation in the building industry:

—design technology: new design methods and techniques are in continuous development. Their enhancement by Computer Aided Design (CAD) tools is evident.

—production technology: here, new developments of component-prefabrication methods and new information technology aimed at Computer Aided Manufacturing (CAM) and Flexible Production Automation (FPA) will be combined.

—communication: in the highly segmented, multi-disciplinary, multi-partner approach to a building project, communication is of vital importance.

Summarizing, we can conclude that information technology is an integral part of present trends in innovation.

Within these trends, the focus of the European collaborative project COMBINE is on improving communication in the design stage, i.e. enabling multi-criterion design through integration of a range of specific disciplinary tools and bringing them at the disposal of the design-team. In the process, each specialized Building Performance Evaluation (BPE)-tool will be embedded in an intelligent design tool, allowing easy use, without requiring specific skills.

By this approach several of the reasons behind the lacking absorption of BPE-tools in design practice are targeted to be removed. At a workshop on future building energy modelling in 1987 [1], organized by the Commission of the European Communities (CEC), great con-

---

* Faculty of Civil Engineering, Delft University of Technology, Delft, The Netherlands.

cern was expressed by many speakers with respect to the use of available tools in building practice and more specifically in design practice. In spite of the great number of BPE-tools available, ranging over a broad spectrum of BPE-aspects and modelling approaches, their use in actual design is rather limited. In recent years, two more or less separate approaches have been attempted to overcome this:

- —introduction of simplified tools into the 'design office' (the technology-demand approach).
- —design-oriented enhancements of sophisticated tools, traditionally used by specialized consultants (the technology-push approach).

Some observations can be made with respect to these approaches:

- —both simplified as well as advanced tools were the results of rather mono-disciplinary R&D efforts, mostly lacking the designer's viewpoint, which makes them unsuited to handle typical design requests, which are typically 'inverse', 'interrogative' and 'incremental' by nature.
- —all available tools usually handle only a few performance aspects of a building whereas concepts for incorporating them in a multicriterion, multi-aspect approach are as yet not offered.
- —it is unlikely that preconceived generic design rules, which usually form the basis of the simplified tools-category, are applicable to real unique building projects given the fact that they are seldom routine or simple. Moreover, the 'old' reasons for developing simplified tools in the first place, e.g. the lack of sufficient computing power, are becoming rapidly obsolete.
- —whereas some of the state of the art BPE-tools are very large and sophisticated programs, some of their inherent deficiencies [2, 3, 4], have given rise to the start of a new generation of simulation environments, based on an object oriented approach [5, 6]. It is expected that this new generation will produce modular and expressive tools, which will lend themselves to be handled easily in an integrated framework.

So, not surprisingly both approaches have not produced any real long-lasting solutions to the observed lack of direct design-support of the present suite of tools.

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 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 distinguished from building sub-systems, which represent 'parts' of the building. Typical sub-systems are the building structure, a room, the HVAC equipment. Realizing that we are trying to integrate a large set of interacting actors, design stages, tools, enterprise-aspects, etc. we must acknowledge that no short term-solutions exist. Certainly, any attempt to produce a complete and final solution will prove to be over-ambitious. This leads us to the belief that any effort within a reasonable time frame will only produce partial solutions. Yet, any partial solution will in itself be difficult to introduce into the 'design office' (used as a generic term here to represent any design-team or design environment) because it will confront us with yet a new integration problem.

Many R&D projects, advertised to the funding bodies as 'integrated design tools', have failed to stress this aspect sufficiently.

Some key-concepts of a particular approach taken in the COMBINE project will be shown in the next sections, starting from an integration framework of the building process as a whole, followed by the specific integration of simulation tools in design and the actual data exchange through an integrated data model, using a standardized (STEP) approach. The actual integration effort, which will result in a prototype of a suite of integrated design tools is discussed in some depths.

2. INTEGRATION FRAMEWORK

Modern IT is widely recognized as a key-enabler of integration of data, process, actors etc. in the building process. It is to be expected that this integration process will make significant progress in the course of this decade.

Looking at the building industry as a whole we can see that integration basically affects the following relations:

enterprise < many to many > project
project < many to many > process/stage
process < many to many > actor
actor < many to many > task
task < many to many > tool (machine, software, robot, labor)

It should be noted that the fragmented nature of the building market (e.g. many enterprises involved in one project and many actors participating in one process) adds an additional layer of complexity, compared to several other industrial activities. Each relation, all of which are of the 'many to many'-kind exhibit a complex structure, thus putting specific requirements on any global integration approach. To establish these requirements, we consider the 'integration space', as consisting of four orthogonal integration axes [8]:

Each axis emphasizes a main integration direction:

Horizontal integration

- —along the life cycle axis: information and decision flow among different stages in the life cycle of the building product, such as briefing, design, construction, use and demolition.
- —along the actor/task axis: flow of information, constraints, regulations, jurisdiction among partners involved in the building industry (not restricted to

project partners, but involving also regulatory bodies, component suppliers and others).

Vertical integration

- along the enterprise axis: relating company-wide administrative and organizational tasks to the making of many separate (one of a kind) building products.
- along the IT axis: dealing with the actual implementation in a suitable integrated software and hardware environment supported by a set of conventions and standards.

If for nothing else, having a sound integration framework is very useful for the following reasons:

- the definition of orthogonal axes is an excellent way to structure concurrent development strategies and define limits of integration exercises e.g. belonging to just one axis or one plane. Interfaces with other projects can be assessed and resulting development-constraints can be stated in an early stage.
- all distributed and concurrent data modelling tasks can start from the same top down view; resulting data models will ultimately exhibit the same orthogonality, a first prerequisite for further integration.

Let us take a closer look at Fig. 1 and try to distinguish different types of integration (on different hierarchical levels if you like) that have traditionally attracted interest from different disciplines:

Organization view (project-enterprise integration): the need for vertical integration in an enterprise is implicit, as the sole purpose for its existence is to bring processes together to work jointly on a number of projects. Yet, traditionally this is the vulnerable part of any building project, due to the fragmented involvement of many partners from different enterprises.

Product view (process/stage-actor-task integration): quality and producitivity improvements are the main issues here. Essential benefits are apparent, for instance:

- store information for later use, i.e. to make the 'as designed' information directly available to later stages, without loss of consistency, integrity and completeness, including the design intentions, underlying the 'as designed' object description.
- enable decisions when they count, i.e. make the 'how to make' and 'how to manage' information available in earlier (design) stages, giving control over the whole life cycle costs in the stage where it counts most.

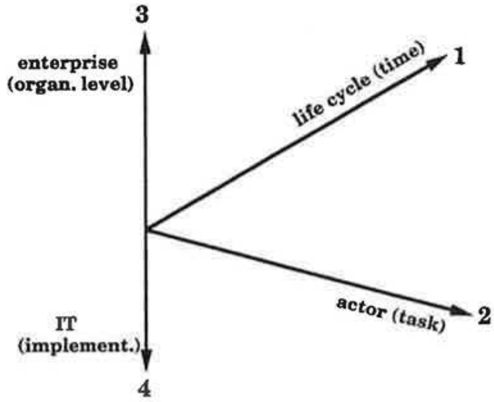

Fig. 1. Integration space (process view).

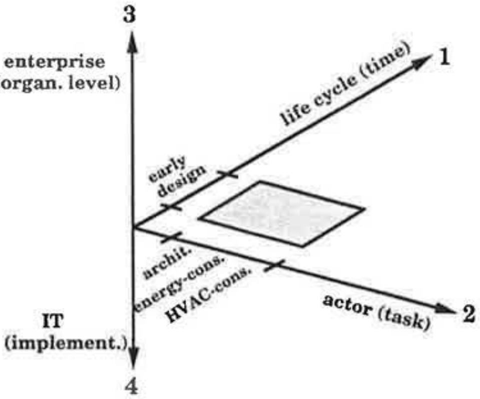

Fig. 2. COMBINE-project as part of general integration approach.

The negotiation of different actor views (abstractions) of the same object is crucial to support this kind of information exchange. It has to deal with conflict negotiation, trade-off between design criteria, authority-distribution, truth maintenance, etc. These aspects are essentially the subject of concurrent or cooperative designing, which is to be regarded as one of the greatest R&D-challenges of this and the next decade.

Software view concerned with providing the basic IT for information sharing and data exchange. The emphasis is on (distributed) data and knowledge base technology as well as development of standard interface specifications.

Communication view concerned with the basic technology to actually support electronic communication, i.e. networking, exchange protocols etc.

Our discussion will focus on design systems and thus cover only a small range of the integration space.

Figure 2 relates the COMBINE project to the overall integration framework. COMBINE is essentially situated in the horizontal (product view) plane, within a limited span along the life cycle axis (i.e. limited to the early design stage) and with a small range of BPE-actors along the actor/task axis. BPE-actors are mostly from the energy performance and HVAC-disciplines.

Within the coordinated European R&D efforts, a number of projects are dealing with the same type of integration aspects, in other industrial sectors. Most of them are carried out in the CEC's ESPRIT [35] programme.

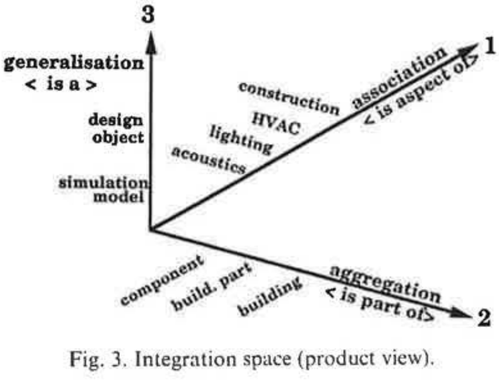

Fig. 3. Integration space (product view).

It should be remarked here that the integration space of Fig. 1 typically represents a process-view.

A product-view would lead to the data integration framework, depicted in Fig. 3. The orthogonalities in Fig. 3 can be straightforwardly applied to the data modelling. Both views (process and product) are complementary parts of the over-all integration strategy.

3. DESIGN SUPPORT LEVEL OF INTEGRATED TOOLS

After having introduced the general scope of the integrated building process as a long term goal, we will now focus on short term efforts dealing with IIBDS developments. Having situated a targeted IIBDS in the general framework, we are confronted with the following options:

- —choice of design stages, actors, tools etc. in order to scale down the IIBDS to manageable proportions.
- —choice of the level of data exchange, e.g. a loosely coupled set of existing tools (enhanced with an open data integration concept) as opposed to a tightly coupled new 'monolith' with an internal 'private' data collection.
- —choice of the level of design support, ranging somewhere in between merely easy data communication ('data integration') and complete and supervisory guidance of the design process ('process integration') from initiation right through to detailed design.

First let us consider the crucial question: 'what design support must be offered by an IIBDS to have any appeal to designers?' In spite of the broad attention that is being given to design methodology [9, 10, 11] i.e. 'how does a designer do whatever he is doing', the slightest form of consensus about an applicable process model for architectural design, will probably not exist for a long time. So, general 'complete design' systems, acceptable to all designers are not realistic to aim for. 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 IIBDSs 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 IIBDSs.

Ongoing R&D efforts in Europe and the US tend to make different choices with respect to the above cited options [13, 14, 15, 16]. Choices range from rather closed-system complete-design support to open loosely coupled tool-sets.

R&D APPROACHES

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

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. The following observations can be made:

- —the top-town 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 IIBDSs.
- —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 [37] 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 in practice 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 systems.
- —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 in a pre-defined context).

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. COMBINE is a typical example project of the object-driven approach. The following observations can be made:

—the bottom-up nature of the approach prohibits early implementation in 'closed' IIBDSs. 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. Resulting (second-generation) design systems would thus be able to truly support concurrent design.

—in the meantime, first generation IIBDSs will provide complete building descriptions in the form of a conceptual 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 4), 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.

In considering the R&D that is required for IIBDSs, it is useful to distinguish two different areas of integration, reflecting the two approaches described above:

—*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 object-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 IIBDSs. 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.

4. DATA INTEGRATION APPROACH

A building project deals with generating, updating and communicating an enormous amount of data, formally denoted by the design object description. The key to data integration, as it is widely recognized [17, 18] is provided by the concept of a product model, which in its purest essence is a complete 'model of reality'. A product model of a building comprises all data that is needed for a complete description of the product in its different stages and hence supports the extraction of all kinds of different views of the product. 'Horizontal' data integration requires the definition of a common conceptual model, able to generate and reconstruct the specific views or abstractions for all "clients", i.e. any actor and pertaining to any aspect or task.

Conceptually, a product model can be regarded as the central core to which all clients relate and with which all clients exchange their data. No direct data links exist between clients, other than through the central conceptual model.

As far as implementation is concerned, there can be many reasons (e.g. efficiency) to distribute the actual product data. Accessing the data through a common schema (which would then have to include location information) still guarantees the full benefits of the product model approach.

There are a number of critical issues at stake in implementing the product model approach in an IIBDS:

—adherence to the STEP/PDES developments [19, 20] that address standardization of the conceptual data model (STEP-entities), schema definition format (EXPRESS) and exchange file format (STEP physical file).
—the development of the conceptual data model in order to support view/abstraction generation; this requires a semantically rich set of entities and relations and (or) intelligent interfaces between core and client.
—the level of data exchange, supported in an actual implementation.

The remainder of the this section will be used to discuss these issues.

STEP STANDARD

An overview of the origin of product model standardization which began in 1984 in the US as successor to IGES can be found in [21, 23]. The US-driven activity, known as PDES (Product Data Exchange Specification) has joined in with the ISO-STEP line.

The ISO has adopted the PDES effort as its first version of the ISO-STEP standard which is available as a draft-standard [22].

STEP's main target is the exchange of multiple representations of the design object between computers. We will refer to these different representations, required for instance by a particular actor (e.g. a BPE-application), as Aspect Models. The central specification through which the exchange takes place is specified in two layers, a conceptual layer and a physical layer. Figure 4 shows how these layers support the actual exchange.

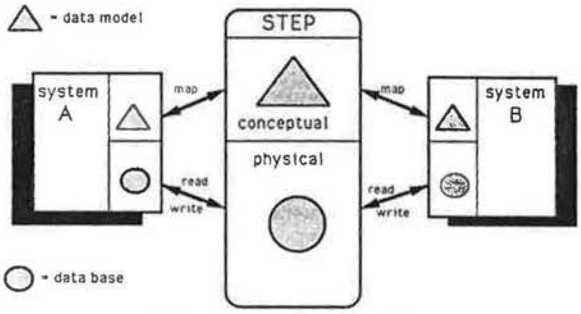

Fig. 4. Data Exchange between two systems.

In its present form the STEP first series of standards offers:

—a conceptual schema, containing a set of entities, for the time being rather limited to geometry and shape representation.
—a standardized schema definition language EXPRESS, which is a powerful tool for concise and complete data definition.
—a standardized neutral file format; this so-called STEP-file contains a header part (the EXPRESS data schema) followed by the data-part, containing the instantiated entities describing an actual product.

A recent strategic paper [23] defines the future directions of the STEP. Amongst others it introduces the concept of Application Protocols. Making use of the aforementioned three STEP-elements is regarded as a key-factor for supporting open system development. A few observations can be made:

—STEP is NOT a solution, but neither is it a constraint.
—STEP is still very limited in its scope; actual integrated building models have still to be developed.
—As STEP is trying to standardize what hasn't been produced yet, it is as much research as it is standardization. This aspect makes it an awkward endeavour by any standard!
—STEP is gaining momentum in Europe through its use in several market oriented CAD-exchange research (e.g. the ESPRIT project CADEX) and in several CIM-prototype efforts.

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

CONCEPTUAL DATA MODEL DEVELOPMENT

A data model exists of a coherent set of entities, relationships, attributes etc., describing the information contained in the communication flows of a process. Usually the process is of a complex nature, which requires us to break it up into smaller manageable parts, i.e. subprocesses or activities. Formal function modelling tools can help us in doing so. Looking at the information flows on a granular level, the atomic data items or entities can be defined, together with the relationships that exist among them. This data analysis must obviously be supported by an appropriate formal tool of which several exist. IDEFIX [24] and NIAM [25] are so-called graphical tools, whereas the EXPRESS language [26] is a concise computer-readable data definition language, developed as part of the STEP-standard.

EXPRESS not only adds exactness to the data model definition, it also adds a number of concepts which cannot be modelled in standard graphical tools, like rules, constraints and methods. These concepts render the language an object-oriented flavor.

The use of graphical tools still offers some advantages in the early stages of the data modelling exercise, i.e. when top-level entities are being defined. Moreover they have proven to be indispensable as a means of communication between the data analyst and the design office, construction site or shop floor, that happen to be the target of the process analysis.

A blend of both approaches in which EXPRESS is progressively used towards the final data analysis stages, during which low level details and additional constraints and rules are added, seems the most appropriate. If we apply this general approach in the data integration phase of an IIBDS development project we can make the following observations:

- —the processes are identifiable with the design-tools which perform the basic design supporting functions, usually limited to a certain design-domain, e.g. shape definition task and performance evaluation tasks. If we suppose these tasks and functions to be performed by separate software programmes, the first step in the data analysis exercise will be to develop data models for the 'external' data (input and output) of these applications, being 'actors' in the IIBDS. Each separate model effectively represents an actor's view on some common conceptual model.
- —having conceptual models for each actor then confronts us with the major and difficult task to integrate these Aspect Models into this common generic conceptual building model.

The successful integration of separate, yet partly overlapping data views in a common conceptual model is a key-requirement for obtaining effective data integration. It has to be supported by flexible data analysis tools and must be based on a collaborative effort of bottom-up data modelling by many separate domain experts and a top-down analysis based on an appropriate top down approach, based on a set of building design related integration concepts. It must be acknowledged that in its present state, STEP offers little with respect to integrated building data models. However, the GARM (General AEC Reference Model) approach, offers a set of concepts on a high abstraction level that could prove to be very suitable for our integration purposes [27]. Other candidates that could prove useful can be found in [28] and [29].

The latter two use a more straightforward and rather fixed abstraction hierarchy.

SCOPE OF THE DATA MODEL

It is apparent that the 'scope' of the data model determines the level of design functionality that can be provided. Scope is tentatively defined here as:

- —the subspace of the integration space that is covered.
- —the semantics represented in the conceptual model.

As design-meaning and design-context of the data is increasingly represented, the number of entities and especially the number of relation types will grow rapidly. Moreover the number of integrity constraints on the data will increase rapidly, whereas inferred knowledge from the data is indispensable to provide intelligent design assistance. It goes without saying that providing intelligent design support requires much more than data integration alone.

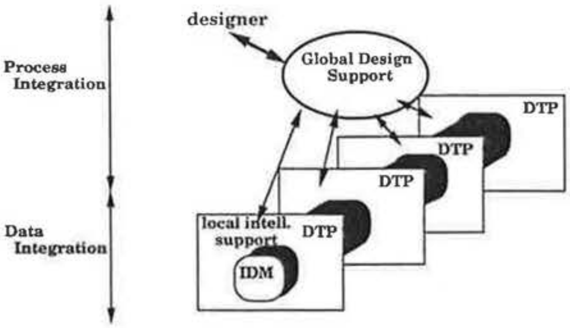

Fig. 5. Levels of integration in an IIBDS.

The added layer (Fig. 5) refers to what we will call 'process integration'. Process integration deals primarily with the design meaning, purpose and context of the exchanged data and the way design functions are called upon and controlled through a dialogue between IIBDS and designer.

The design process layer should help the designer-team in conflict-resolving and negotiation, thus steering and supervising the design process as a whole. Evidently, data analysis, as mentioned before must be augmented by design process modelling, while paying equally attention to the dynamics and time dependencies of these functions as well as decision flow through time. An approach to decision flow control is offered by the GRAI method [30].

DATA EXCHANGE IMPLEMENTATIONS

It should be noted that we have confined the discussion up till now to the conceptual level of the data exchange. Implementing the actual exchange requires us to make important choices with regard to:

- —the functional level of exchange (explained below)
- —the required transfer efficiency
- —the implementation effort we are willing to put in

whereas the communication hardware in relation to the 'proximity' of the exchanging actors (i.e. software applications) pose important constraints. A simple diagram (Fig. 6) helps to explain this. Three 'distance' levels with respect to exchange implementations can roughly be distinguished:

level 0: between software modules of one application; this level is considered to be outside of the scope of the conceptual data model as no separate functions of these software modules are specified and hence no separate 'view' is supported. The data sharing is accomplished through conventional parameter passing mechanisms.

level 1: between 'nearby' actors through a shared data area or 'work form'. The data area will normally contain a schema definition part and a data part. Interfaces are required to map the data in the data area to the internal data structures. Typically this is the kind of exchange used between

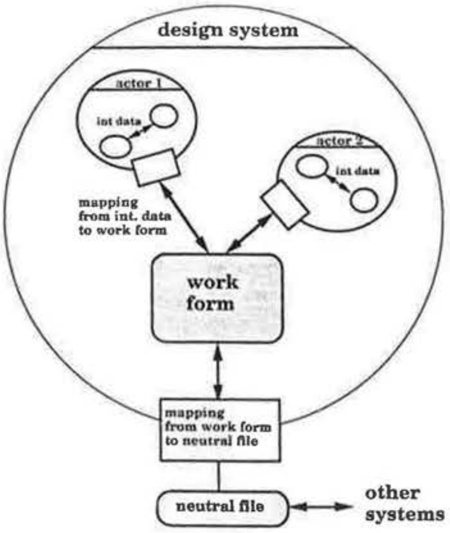

Fig. 6. Exchange solution depending on 'distance'.

tightly coupled or closely related applications inside an IIBDS.
level 2: between 'remote' actors through a ASCII neutral file format (STEP file). Interfaces are required to map the schema and data in the STEP file to the shared data area, or alternatively directly to the internal data structure. Typically, this kind of exchange is used in loosely coupled applications in an IIBDS, or indeed between two IIBDSs.

Levels 1 and 2 address exchanges between separate actors, i.e. exchanging fully recognized and completely defined views of the design object. It is important to note that logically, all exchanges take place through the complete and common conceptual model, yet the actually shared data could be stored in distributed fashion or it could be decided that only subsets of the actual data (e.g. only the data that is actually mutually relevant) is sent between two actors. In that case the conceptual data model must contain references to storage locations whereas in the latter case, it must contain a subset mechanism or entities to support the extraction of subsets.

The alternative would be to exchange subschema's of the conceptual data model in the STEP-file, thus explicitly defining the subsets in each transmission. In a recent ESPRIT project IMPPACT [31], one has realized efficient control over distributed data by adding a message-extension to EXPRESS. By this approach one has among others circumvented the necessity of storing all integrity constraints centrally in the global data model. Messages are issued to distribute this type of integrity control to local actors. Apart from the level of coupling of the software actors, we can distinguish the following implementation performance levels of the data exchange. The 4 levels below are defined in STEP standardization work in progress [36]:

level 1: neutral file transfer (STEP-file)
level 2: 'work form' with an access method (data area in main memory)
level 3: database management system (DBMS); access methods are provided by a standard query language.
level 4: combined data and knowledge base; providing 'intelligent' access and control of the shared data (e.g. constraints, rules, inference queries).

It is obvious that any attempt to provide process integration would require level 4. Many challenging issues remain to be solved for finding efficient implementations of distributed data and knowledge bases in future IIBDS's. Blackboard architectures seem to get a lot of attention at the moment.

5. KEY-ISSUES IN BUILDING DATA MODELLING

As explained above, the scope of the data modelling effort is determined to a large extent by the views we want to integrate. Application-specific interfaces must then be developed to take care of the actual mapping between the resulting integrated data model entities and application entities. Other functional requirements would further guide implementation choices, such as whether to exchange through a neutral file (static interface) or through a shared database access (dynamic interface). Some of the topics that guide the development will be discussed below.

5.1. *Framework and modelling paradigms*

In general, any data model development effort will be guided by the following considerations [38]:

- —the intended scope of the central model; i.e. what data is exchanged through the central model, and what data is considered to be application-specific and thus 'private' to an actor.
- —what top-down concepts and abstraction mechanisms are supported by the conceptual building model, e.g.
  - ● generalization-specialization: in order to reduce data redundancy we need a mechanism to 'describe' entities on different levels of specialization, e.g. employing a generic ('an air duct'), specific ('a PVC-airduct with diameter 0.15 m'), occurrence ('the PVC-airduct, diameter 0.15, length 6.00 m between joint A and Joint B') hierarchy.
  - ● aggregation-characterization: entity-description should support 'aspect-of' relationships, thus enabling an aspect-oriented (e.g. color, strength, cost) view of the product.
  - ● decomposition-composition: the building model should support 'is part of' relations, thus enabling views on different levels of detail.
  - ● life-cycle stages: a building description goes through life cycle stages, that could be classified according to some process-oriented categorization of specific decision-points in time.

  As integration across life cycle stages is one of the prime targets of integration, the data model must support life cycle views and their relationships.
- —the availability of a general framework to harmonize ongoing data modelling efforts. In fact the ISO-STEP

effort performs a major role in this respect; a recent STEP-document defines the following layers [39]:

- definition: all data-aspects of a product, other than its shape-representation. This layer in fact contains three sub-layers, dealing with context-definition, product-definition and property definition. *Example: for a duct system, its component definitions and their relations would be on this level.*
- shape representation: all data-aspects of the shape of the product. *Example: representation of the shape of a duct, e.g. entity 'cross-section' with attributes 'cylindrical' and radius-value.*
- shape presentation: all data aspects of how the shape is presented. *Example: the duct-cross section could be presented by either center point and radius or through 3 points on the cylinder-boundary.*

It should be noted that the presentation layer is strictly speaking not considered part of the general framework of STEP.

A few observations are in order to highlight some of the challenges that one faces in the development of complete product models:

—no complete building data model will become available in any foreseeable future. It is therefore important to build in guarantees for future extensions in ongoing developments. Support of above mentioned concepts is one of them.

—as different application views imply different levels of abstraction, it is of the utmost importance to support useful abstraction mechanisms. It is debatable whether characterization and decomposition supply enough richness in this respect. Often a particular abstraction is guided by an application-specific schematization and (physical) modelling approach. Further work in the area of modelling methodology based on a categorization of physical agents and a taxonomy of building behaviour should blend in with ongoing data modelling efforts. Among others, valuable input is expected from the EKS project [5].

—one must realize that important operational and user issues of the (implemented) central building model are more or less blocked out from present day attention, e.g. ownership, versioning, authorization etc. In fact, especially the general support along the life cycle stages-axis will entail a great deal of implied procedural knowledge dealing with these issues. Process knowledge captured in process and decision flow diagrams will thus have to supply essential support to the use of the building data. Although we see rapid progress in CIM projects, mostly dealing with fairly simple part production processes, application to the building industry seems to be some decades away.

—another important issue is the type of product modelling power one is able to supply to different types of users, i.e. 'Who can define what?' Taking an existing first shot at a building product model supplied for instance by an ongoing data modelling exercises, it does not take too much imagination to come up with an existing building that one would not be able to model according to this data model structure. This fact would make it quite unacceptable to the average designer, one suspects. So, why not take a broader view and only define the way we describe the product and not the product itself. This would imply that a major part the real modelling power, i.e. what the product IS and not just filling in a 'hard-wired' description format, is delivered to the designer. Obviously, for any meaningful exchange of data to occur, one would have to adhere to certain conventions known to both transmitting and receiving ends, which however would not have to be hardwired if one can find a way to exchange these conventions on the fly.

Finally, it is important to realize that any approach to building data modelling is to a large extent driven by the global objective, defined at the outset. Data modelling in a context, dominated by design-flow considerations would clearly demand a richer set of abstraction mechanisms (i.e. a richer data model) than those, merely dominated by data exchange considerations. Recent research in the area of a new building design-oriented data model (EDM) looks to be promising in this respect [40].

5.2. *Topology-geometry issues*

Since the earliest introduction of CAD systems, design has been dealt with as a visually driven exercise. In fact the emphasis on drawing capabilities of present CAD systems is still pre-eminent. Recent years have witnessed a quite natural evolution from presentation-based 'drawing' systems to more design-oriented systems enabling the addition of design semantics to a shape representation, the latter still being regarded as the backbone of the design process. Interestingly enough, growing requirements on capturing the semantics of a product are pushing the shape aspect even more to the background, as is reflected by the STEP-framework mentioned in the previous section. To elucidate this, let us reflect on the different roles that the shape-aspect can play:

—in present day CAD-systems: a shape-driven design approach; topology and geometry are used as the 'glue' by which entities are connected. A closer look reveals that this approach can be very restrictive and is in fact unable to support a top-down design process. Such a process requires that many relations (only some of which are topological by nature) can be applied on different refinement levels, even before some shape information is available.

—in future CAD-systems: relations are explicitated on different levels of refinement, topological relations (e.g. connectivity relations) are refined on lower levels adding geometry where appropriate. This implies that shape is (partly) evaluated from the semantical definition of the product. In other words, the product glue is in the product description itself, not in its shape.

As far as classical data models for shape representation are concerned we conclude that they do not as such supply us with enough power to support the second approach. Looking more closely at these data models (Fig. 7) we can make a few general observations [38]:

—Wire frame representation (WF Rep): the simplest representation, dealing only with topological Vertex and Edge entities. Geometry is nicely separated through adding Shape Type to edges and Coordinate

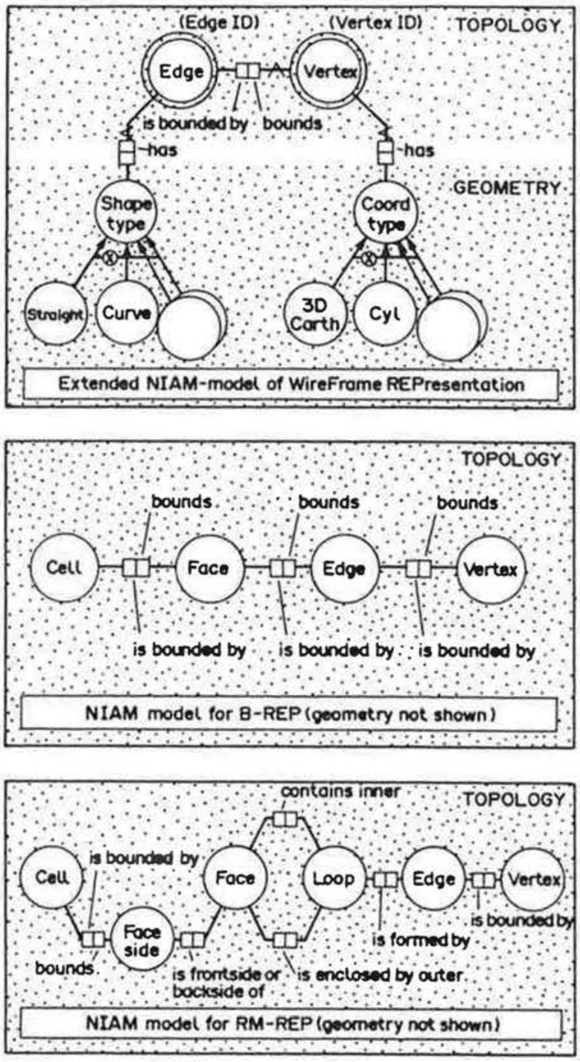

Fig. 7. Classical shape representation models in NIAM format.

Type to vertices. It needs little clarification that this model is inadequate for a general building shape representation as no knowledge about spaces (voids) and solid material can be stored.

—Boundary representation (B-rep): a popular solid model representation dealing with topological Solid-Face-Edge-Vertex entities, whereas geometry is added similarly to the WF-rep. Although much richer that WF-rep, B-rep is still too restricted for building modelling as it is unable to model internal spaces, which is not surprising in view of its solid model origin.

—Reference Model representation (RM-rep): based on B-rep with added Cell and Loop entities in order to adequately model spaces and holes (voids) in faces. Face Side is introduced to unambiguously determine the relation between two adjacent Cells. Again geometry is not shown; all depicted entities are topological, which thus act as the reference for the actual shape represented in geometry entities (this aspect explains its name Reference Model, i.e. a topological reference structure for the explicit geometrical shape). It must be noted that for building shape representations on a non-detailed level (e.g. regular spaces, flat walls) no extra geometry-entities are needed.

Although the RM-rep is quite adequate to model the shape of buildings it cannot be used as the spinal cord of the building product model. The main reasons for this will be summarized below:

—in the early design stages, one must be able to store information although there is as yet no explicit shape information.

—different abstraction levels should be supported by different shape representations. As we have seen in Section 3.1, the support of explicit abstractions is of key-importance to support different views of a building; it is quite obvious that these abstractions require different levels of shape representation detail, in order to support application models of a building. If this requirement is not fulfilled, the totally 'flat' shape description spans the range from global objects (room, wall) to the finest details (door knob) on one and the same level.

6. COMBINE PROJECT

Following the 1987-workshop mentioned in the Introduction, the initiative was taken to start a new R&D topic on IIBDS within the JOULE-programme of the CEC. Prior to the call for proposals, a pilot study [7] was carried out, which assessed the state of the art along the lines stated above and defined the new R&D topic in broad terms. The review process of the submitted proposals ended in 1989 and a new project labeled by the acronym COMBINE (Computer Models for the Building Industry in Europe) was set up, based on a small number of accepted proposals. The present COMBINE project will span a period of approximately two years, ending in the fall of 1992.

A set of prerequisites for performing an effective first step towards future IBBDS was set up in the COMBINE pilot study:

—a multi-disciplinary approach steered from the designer's or rather design process-viewpoint is to be regarded as essential

—attempts to actually build an IIBDS would be unrealistic at this moment (wasting premature implementation-efforts is a risk, not always fully recognized in many of the ongoing R&D of this kind)

—development of concepts for integrating (existing) tools in the building process is a prime target

—a development-approach must be established which is conceptual and open and adheres to available standards

—demonstration of these concepts in prototypes is a necessity

—future efforts can build upon developed integration concepts which must offer a standard way how to configure limited implementations (i.e. in terms of aspects, actors or stages of the design process) of actual IIBDS.

The last prerequisite is rather ambitious, as it requires the COMBINE-project to come up with sufficient integration concepts to allow for distributed development of components which eventually can be integrated into general design systems. Based on the set of requirements that were presented in the introductory section on the one hand and the selected R&D proposals from research institutes from European community countries on the other, the COMBINE project (funded in part by the CEC) was defined. The actual work started mid 1990 and will be concluded around the fall of 1992. The project set-up and other aspects will now be presented in some detail.

Fifteen partners from eight different countries participate in the project:

> Building Research Establishment, UK (Parand), Comité Scientifique et Technique des Industries Climatiques, France (Hoffmann), University College Galway, Ireland (Monaghan), Université de Liege, Belgium (Dupagne), Statens Byggeforskningsinst., Denmark (Christensen), Fraunhofer Institut für Bauphysik, Germany (Erhorn), University College Dublin, Ireland (Lewis), University of Ulster, UK (Norton), University of Edinburgh, UK (Tweed), Centre Scientifique et Technique du Bâtiment, France (Dubois, Poyet), TNO-Bouw, The Netherlands (Plokker), University of Newcastle upon Tyne, UK (Wiltshire) and Technical Research Center, Finland (Björk).

Task-leaders are mentioned in parenthesis. The project is coordinated by Delft University of Technology (Augenbroe).

General objective

COMBINE will perform a first step towards future intelligent integrated building design systems (IIBDS). Within its JOULE context the emphasis will be on energy performance related aspects, as addressed by a set of existing and new BPE (Building Performance Evaluation) tools. Other tools will address typical architectural design tasks in the early design stages, e.g. sketch design of inner spaces-layout. The project aims to provide a conceptual basis for future integration developments and demonstration of these concepts through a number of limited prototypes. The integration focus will be on data integration whereas limited design intelligence will be provided locally by the design tools.

Remarks:

—Although actual developments of a full-blown IIBDS is not pursued, a set of concepts leading to their development must result.

—Prototypes will serve as a proof of concept.

—All development must reflect the designer's viewpoint, design practice will be involved in the actual definition of the prototype.

—A demo design-session with the prototype-IIBDS will be developed by adding a (necessarily very restrictive) design process on top of the prototype.

—Data integration will be accomplished through a common conceptual data model and conformance it to the STEP-standard (EXPRESS, STEP-file). All exchanges will logically take place through a so-called Integrated Data Model (IDM). The actual exchange will probably be a blend of exchanges on level 1, 2 and 3.

—'Local' intelligence within the design-tools focuses on the mapping between IDM and local aspect model and the integration of their design-meaning.

—Present description formats [32] for maintaining a library of BPE-tools are refined and used for formal description and storage of the BPE-tools employed.

With reference to the last two remarks it must be emphasized that major breakthroughs are expected from ongoing developments of a new generation of flexible object-oriented simulation tools such as SPANK [6], EKS [5], IDA [33] and ZOOM [37]. These environments will eventually offer 'automatic configuration' capabilities which make them ideal candidates for automatic adaptation to arbitrary requests defined by their design contexts and handled through an intelligent IDM-interface.

Global architecture

The IIBDS prototype will consist of a set of design tool prototypes (DTP), logically shared around the common conceptual data model (Fig. 8). The Application Interface executes the mapping between the IDM and the aspect model of the design tool. The six DTP's that will be developed address the following tasks:

DTP-1: Construction design of external building elements.
DTP-2: HVAC-design.
DTP-3: Dimensioning and functional organization of inner spaces.
DTP-4: Input generation for a thermal simulation tool in the late design stage.
DTP-5: 'L-T method' in the early design stage (spreadsheet-based energy analysis).
DTP-6: Energy-economic design based on the RATAS building model [29].
DTP-7: Geometric Modeller.

7. FOLLOW-UP

With the focus at the present effort being on data integration i.e. a set of BPE-tools shared around an IDM, with some local intelligent design support, several extension-directions for future follow-ups can be identified:

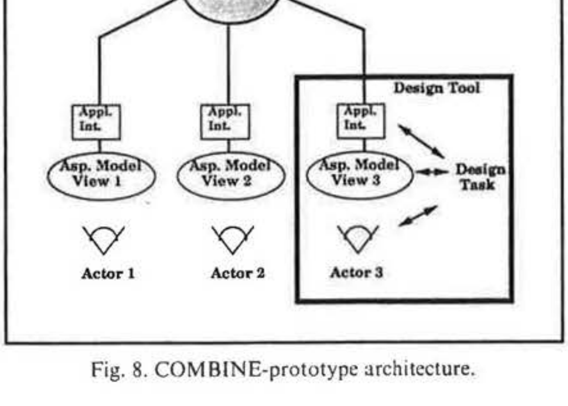

Fig. 8. COMBINE-prototype architecture.

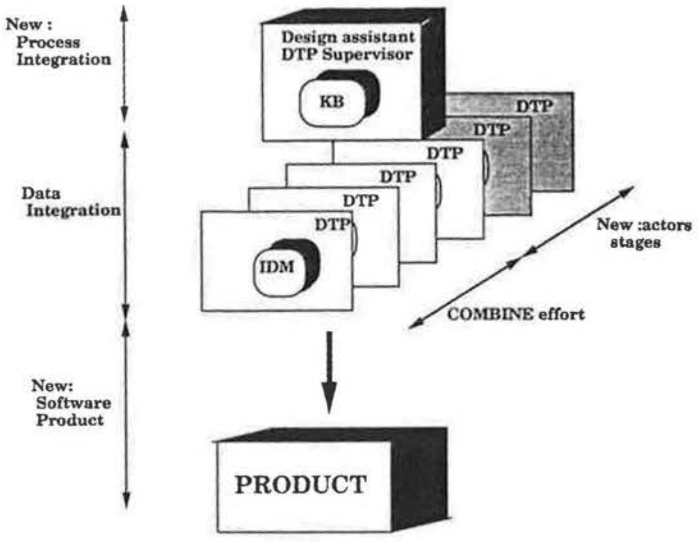

Fig. 9. Possible future extensions.

(i) extensions in the horizontal integration plane, i.e. adding more 'actors' and covering more design stages.

(ii) 'vertical' process integration, through the addition of a knowledge layer around a knowledge base (KB) on top of the DTP's.

(iii) 'vertical' IT-development towards a software product for the design office.

The three directions are sketched in Fig. 9.

It needs little clarification that an attempt at building a full-blown IIBDS requires an extension in all three directions 1, 2, 3 and probably in that order.

REFERENCES

1. D. van Hattem (ed.), Proceedings, Workshop on future building energy modelling, Commission of the European Communities, Ispra (1987).
2. J. A. Clarke, The future of building energy modelling in the U.K. Report to the Building Subcommittee (1987).
3. F. C. Winkelmann, Advances in building energy simulation in North-America. *Energy and Buildings*, **10**, 161–173 (1988).
4. T. J. Wiltshire and A. J. Wright. Advances in building energy simulation in the U.K. The SERC Programme. *Energy and Buildings*, **10**, 175–183 (1988).
5. J. A. Clarke *et al.*, An object-oriented approach to building performance modelling. *Proceedings USER-1 Conference, Ostend* (1988).
6. W. F. Buhl *et al.*, Object oriented programming, equation-based submodels and system-reduction in SPANK. *Proceedings Building Simulation* '89, *Vancouver* (1989).
7. G. Augenbroe and L. Laret, COMBINE pilot study report to the Commission of the European Communities CEC-DGXII, Brussels (1988).
8. F. Tolman, Introductory course in product modelling of constructions (in Dutch). Delft University of Technology (1989).
9. J. S. Gero (ed.), Expert systems in computer aided design. *Proceedings, IFIP WG* 5.2 *Working Conference, Sydney* (1987).
10. J. S. Gero (ed.), Knowledge engineering in computer aided design. *Proceedings, IFIP WG* 5.2 *Working Conference, Budapest* (1985).
11. Y. E. Kalay (ed.), Evaluating and predicting design performance. 3*rd Int. SUNY Buffalo CAD Symposium* (1990).
12. A. C. Tweed, Describing design tasks. *Proc. Eighth Israel Conference on CAD/CAM and robotics*, Tel Aviv (1986).
13. A. C. Tweed and A. Bijl, *MOLE: a reasonable logic for design? Intelligent CAD systems* 2: *Implementation Issues*, P. J. W. ten Hagen *et al.* (eds), Springer-Verlag, Berlin (1988).
14. J. Pohl and A. Chapman, An expert design generator. *Arch. Sci. Rev.*, **31**, 75–86 (1988).
15. S. J. Fenves, *et al.*, Integrated software environment for building design and construction. *Computer Aided Design*, **22**, 27–36 (1990).
16. G. Carrara, Y. E. Kalay and G. Novembri, A computational framework for supporting creative architectural design. 3*rd SUNY Buffalo CAD Symposium* (1990).
17. J. A. Turner, A systems approach to the conceptual modelling of buildings. Proceedings, *CIB Seminar on Conceptual Modelling of Buildings, Lund* (1988).
18. K. A. Reed, Product modelling of buildings for data exchange standards; from IGES to STEP/PDES and beyond. *Proceedings, CIB seminar on Conceptual Modelling of Buildings*, Lund (1988).
19. B. M. Smith, Product Data Exchange; the PDES project-status and objectives. NIST, US Dept. of Commerce (1989).

20. F. Tolman and W. Gielingh. STEP planning model. ISO-TC184/SC4-WG1 Document (1989).
21. ISO-TC184/SC4: External Representation of Product Definition Data. ISO Status Report (1989).
22. ISO: External representation of product definition data (STEP). ISO-DP 102030 (1989).
23. W. F. Danner. A proposed integration framework for STEP. NIST, US Dept. of Commerce (1990).
24. Information modelling manual. IDEF1-Extended (IDEFIX). ICAM, D. Appleton Company (1985).
25. D. Thomson, Nyssen Information Analysis Method (NIAM). Contral Data (1986).
26. D. Schenk, Express information modelling language reference manual N386, ISO TC184/SC4/WG1 (1989).
27. W. F. Gielingh. General AEC reference model (GARM). TNO Report BI-88-150, Delft (1988).
28. J. A. Turner, AEC building systems models. ISO TC1184/SC4/WG1 Working paper (1988).
29. B-C. Björk, Issues in the development of a building product model. Int. Workshop on computer building representation, Chexbres (1989).
30. G. Domeingts, Use of the GRAI method for the design of an advanced manufacturing system. *Proc. 6th Int. Conf. Flexible Manufacturing Systems, Torino* (1987).
31. A. Meier, IMPPACT-Improvements on integration by a feature approach. Product and process modelling move closer together. *Conference Proceedings ESPRIT '90, Kluwer Ac. Publ.* (1990).
32. L. Laret and A. M. Dubois, Modelisation et simulation de fonctionnement thermique des batiments et des equipements energetique associes. CSTB Report MGL/88-1031/NB (1988).
33. P. Sahlin, MODSIM, a program for dynamical modelling and simulation of continuous systems. Inst. of Applied Mathematics, Stockholm (1988).
34. J. L. Bonin *et al.*, Coupling analysis in building thermal simulation: the ZOOM program Proceedings International Energy Society Conference, Hamburg (1987).
35. ESPRIT-Computer Integrated Manufacturing, Results and progress of selected projects in 1990. CEC DG XIII, Brussels (1990).
36. J. Altemueller, PDES/STEP implementation levels, ISO TC184/SC4/WG1, doc. N282 (1988).
37. T. Tomiyama and H. Yoshikawa, Extended General design theory. CWI report, no. CS-R8604, Centrum voor Wiskunde en Informatica, Amsterdam (1986).
38. F. P. Tolman, W. F. Gielingh, P. Kuiper, P. H. Willems and H. M. Böhms, Four Years of Product Modelling, collected papers. TNO report BI-89-140, Delft (1989).
39. W. F. Danner and Y. Yang, Generic Product data resources for STEP. NIST, US Dept. of Commerce. NISTIR (1991).
40. C. M. Eastman, A. H. Bond and S. C. Chase, Application and Evaluation of an Engineering Data Model. *Res. Eng. Des.* **2**, 185–207 (1991).