

"CONCEPTUAL MODELLING OF BUILDINGS"; Seminar, october 1988, Lund.

Proposed paper : " A generic approach to the modelling of buildings, focussing on the representation of productstructure".

Albert A.J. Pols, professor of planning and design, department of Civil Engineering, Technical University of Delft, the Netherlands.

Keywords : modelling of buildings, product-structure.

Summary.

An approach to the modelling of buildings is presented, focussing on the invariables in the representation of the building as a complex object, composed of spaces, assemblies of building elements and materials. The general informationarchitecture includes: functional and technical specification, the determination of form and dimensions, determination of productstructure, cost and performance estimation and the transformation of productstructure into building operations.

The emphasis is on the representation of the productstructure of buildings from different viewpoints and in different phases. Within the productstructure a distinction is made between spatial structure and "material structure", the way materials, components and assemblies are combined into the endproduct as a whole. This leads to a semantic network with various hierarchies of abstraction, allowing for an integration of top-down and bottom-up approaches in various stages of the design process and a combination of generic and specific representations of components, assemblies and productstructure.

Related issues of datastructuring and -modelling and variational and combinatorial design are discussed on the basis of a prototypical application: an assembly consisting of doors and windows, their frames, wall openings, wall and loadbearing structure. The application involves a hybrid object-oriented DBMS, Tornado.



1. Datamodelling perspectives on the "real world" of building design and construction.

The product life cycle approach, integrating object- and project data and the various processes over the entire life cycle, provides a general framework for determining the information needs of building design and construction. Various stages in product development and "viewpoints" (users and applications) have to be taken into consideration in datamodelling. Information on the "real world" of building design and construction is represented in terms of uniquely identifiable entities (concrete or abstract objects) with certain properties (attributes, attribute-values) and relationships between them. The relationships may be one-to-one, one-to-many and many-to-many. Relationships can be derived from other relationships. The value-set of attributes, i.e. the height and width of a certain type of door or beam, may be restricted to a predefined domain.

The datamodel may include:

- static properties (objects, their properties and relationships);
- dynamic properties (such as operations on objects, operation properties and relationships amongst operations);
- integrity constraints over objects (valid database states) and (valid state transitions).

Integrity constraints are used to define static and dynamic application properties which cannot be conveniently expressed in the datamodel. Some constraints are inherent to the data model (inherent constraints); others are defined explicitly (explicit constraints) or are logical consequences of other constraints (implicit constraints). Examples of explicit constraints are: names, object-structures, specific relationships and assertions over objects and their properties. Integrity constraints include valid states (static) and transformations (dynamic). The hierarchical and network models have some inherent constraints: owner-member relationships in a hierarchy for instance. Tuples in the relational model must be unique within a tuple. The implementation of a conceptual database scheme is restricted by the limitations of available DBMS- technologies. [Brod 84].

The datamodel has to fulfill multiple demands, in different stages of design and construction, from various users and applications. The various partial representations nevertheless refer to one and the same object, the datamodel of which is defined interactively (in a series of queries and transformations) as the design proceeds. Integration into a conceptual model of the different "external views" is required, a representation scheme which fulfills general criteria, such as: validity, completeness, consistency and non-redundancy. The conceptual datamodel must allow the user to define his personal view, eliminating all unnecessary detail, and ensure over all users and applications a valid and consistent representation. Datamodelling in an interactive design environment requires subschema integration, multiple view management and dynamic restructuring. The specific requirements of an engineering database differ significantly from administrative applications: the ability to handle versions and variants of objects, complex abstraction hierarchies and identification and classification schemes, nested and bundled relationships, dynamic associations of geometric and non-geometric data (Enc 83). Abstraction hierarchies suitable for modelling building design data can be obtained through abstraction ("class"- or "type"- concept; property inheritance) and aggregation (compound entity-classes). A "tree" as an entity-class may refer to the generalization of all trees (oak,- birch trees etc) or the aggregation of stem, branches, leaves and roots.

The extent to which these requirements can be met, in the conceptual model and its implementation, depends upon the capabilities to formalise semantic meaning and integrity constraints in the datamodel and the practical limitations of DBMS-technologies. The traditional hierarchical and network models do not meet requirements of data independence : separation of logical datastructure and physical implementation. Networkdatabases are too slow for information retrieval and are not suited for integrity checking and security control. The relational databases cannot handle the complex and dynamic design objects and relationships in engineering. Classical approaches to datamodelling were restricted to the representation of the possible content and structure of the data , separate from their semantic meaning.

2. Representation of product-structure.

In the course of the designprocess the model of the building-object evolves through a series of transformations. A structured set of functional specifications and bugettary constraints is first transformed into spatially structured functional areas and volumes. Concepts of form/shape and dimensions gradually materialise; the spatial and material structure of the building-object is further specified in conjunction with the representation of form and dimensions. The iterative determination of form , dimensions and product-structure results in the specification of the final product , reflecting more detailed functional specifications, and estimates of costs and expected performance on a variety of design criteria. Continuous cost/quality trade-offs are involved. The representation of product-structure is embedded in a more general information architecture for the representation of building objects and -processes in design and construction. (figures to). Control-loops have been omitted, presenting only a skeleton of information-entities and major transformations. The "product-structure" consists of interdependent, spatial and physical, structures.

The 3-D spatial object may contain spaces within spaces and can be assembled from - or decomposed into - more elementary geometric primitives: surfaces, lines and points. In the data-structures many-to-many relationships predominate (e.g line-surface, component-assembly). Generally a distinction is made between "integrated modelling" and partial representations derived from the model: a "view" or display. As a 3-dimensional object the building can be represented (in relative and world-coordinatesystems) as a composite of complementary volumes: a combination of interrelated solids and voids with topological, geometrical and non-geometrical properties. The latter refer to functional, physical and economic properties of the various entities and relationships amongst them ; the non-geometrical attributes may also include data on technology ,operations and resource-requirements. The product-specification at the component-level is not limited to dimensions and materialproperties and tolerances, but may also include assembly- and manufacturing -"recipes" (activities and resources ,times and costs) and maintenance-expectations .

The material productstructure of a building can generically be represented as an ordered set of assemblies of assemblies , each consisting of various component-types. Objects with common characteristics may be considered as a class of similar objects; through a process of generalization or abstraction an object-type or entity-class can be defined. "Column" for example, is a generalization of rectangular and circular columns with different dimensional and material properties, locations and orientations. The entity-class is an abstract object which on its turn can be regarded as an entity. Through specialization, the inverse of generalization, sub-components or specific components (instances, occurrences) can be derived. In representing a T-beam additional attributes have to be specified,

which are only common to T-beams. Next to the properties common to all individuals of a class, specific components may have auxiliary properties which can affect the integrity of the design at the assembly-level. Compound entity classes provide a common reference to several entityclasses, having data elements in common or common relationships to other entity-classes. A "floor" can be regarded as an aggregation of spaces,panels and walls. The inverse operation of aggregation is decomposition. The type-concept provides a conceptual basis for adaptive and parametric design. The aggregate-concept corresponds with assemblies involving combinatorial design of different component-types and their instances or occurrences.

The generic form of the productstructure can be represented by a network , handling both one-to-many and many-to-many relationships. The networkmodel is based on linked sets in a ring structure; each ring each constitutes a set and each set includes owner-member relations between entities. An object can belong to one or several sets. To describe many-to-many-relationships link entities can be introduced; each link entity is connected as member in two sets. Hierarchical structures can be implemented as a restricted network. The network, relational and semantic datamodels and databasetechnologies can be combined in a hybrid model. This combination is in some hybrid DBMS, such as TORNADO, even possible at the object level [Lil 86]. A single object can be part of a network-structure, contain a structured name and have one or more relational tables attached to it. Basic componenttypes, which are relevant from a specific viewpoint and at a particular stage of the design process may be combined into larger assemblies, accounting for functional, topological and geometrical relationships. To account for "emerging" properties at the assembly level which can not be derived from the constituent components, the assembly may be defined as a new logical entity. Assemblies of assemblies may be formed; sub-assemblies can be distinguished as well.

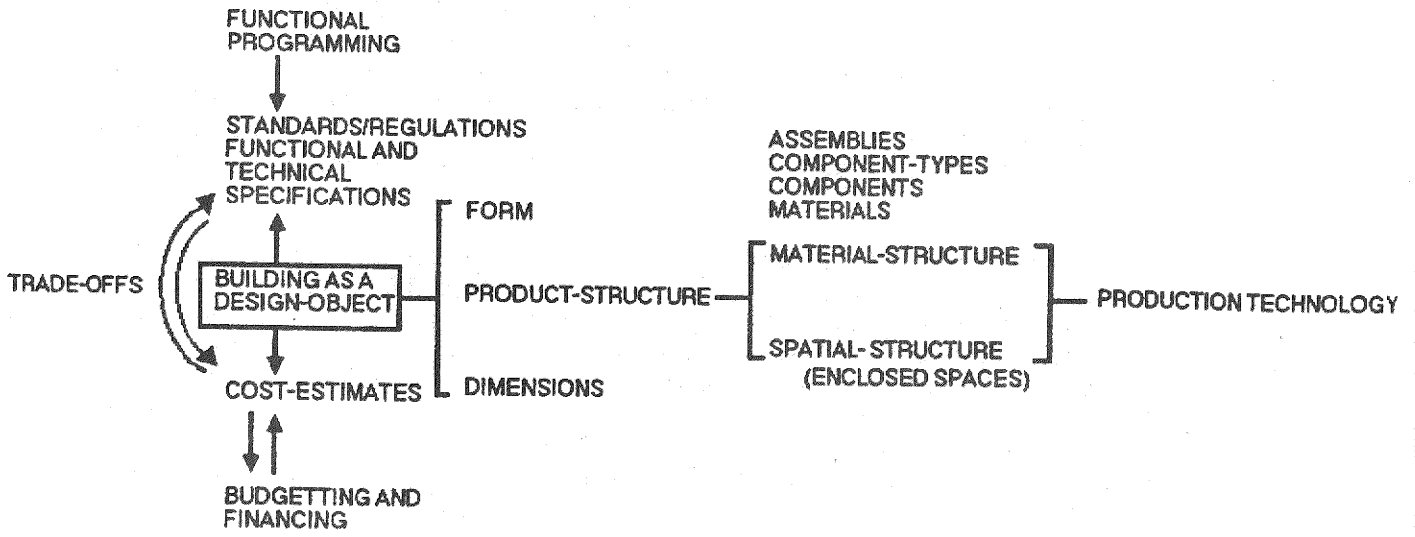
Integrity can be maintained in designing or adapting assemblies, composed of components belonging to various componenttypes, by imposing constraints upon:

- the allowable (range of) attribute-values;
- value combinations of different attribute-values of a component-type and specific components;
- the set of common attributes of a component-type;
- functional, topological, geometrical and physical relationships between components in an assembly or amongst different assemblies.

References

- [Brod 1984] On the Development of Data Models, in : M.L. Brodie, J. Mylopoulos, J.W Schmidt , " On Conceptual Modelling" ,Springer Verlag , 1984.
 [Enc 83] J. Encarnaco, E.G. Slechtendahl, " Computer Aided Design", Springer Verlag ,1983
 [Law 87] K.H. Law , M.K. Jouaneh, Data modelling for building design , in : Computing in Civil Engineering , pg 21-35, ASCE, 1987.
 [Lil,1986], Tornado Database Design Guide, METIS/AS , Horten, Norway

"CIM"-ARCHITECTURE FOR COMPUTER INTEGRATED CONSTRUCTION



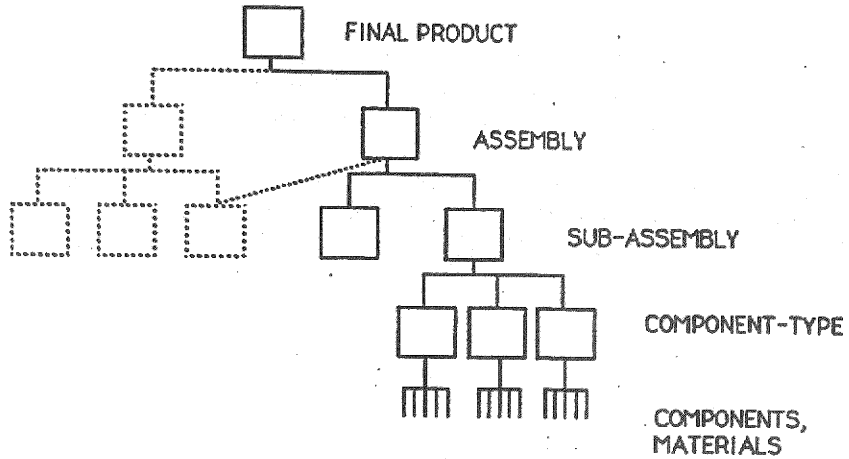
VARIANTS IN PRODUCT-DEVELOPMENT

"PRODUCT-FAMILY"

TOP-DOWN



SPECIFIC BUILDING



BOTTOM-UP



ADAPTATION TO SPECIFIC FUNCTION, CLIENT, SITE, ETC.

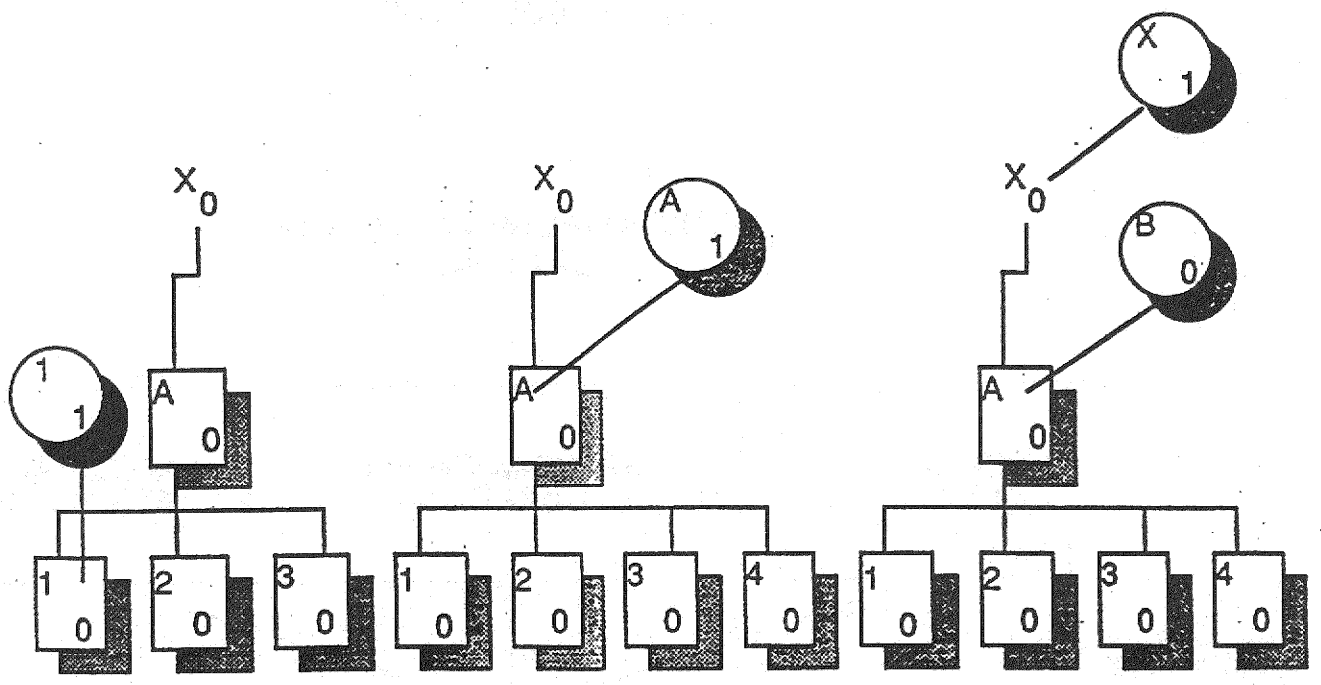
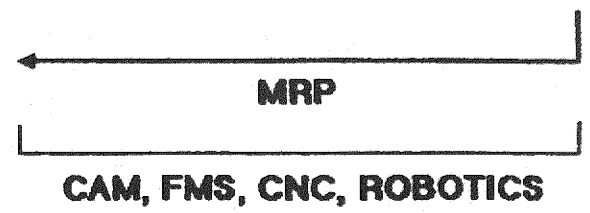
CAE-CIM DESIGN AND PRODUCTION

RADICAL INNOVATION
 - PRODUCT
 - COMPONENT
 - MATERIALS
 - PRODUCTION TECHNOLOGY

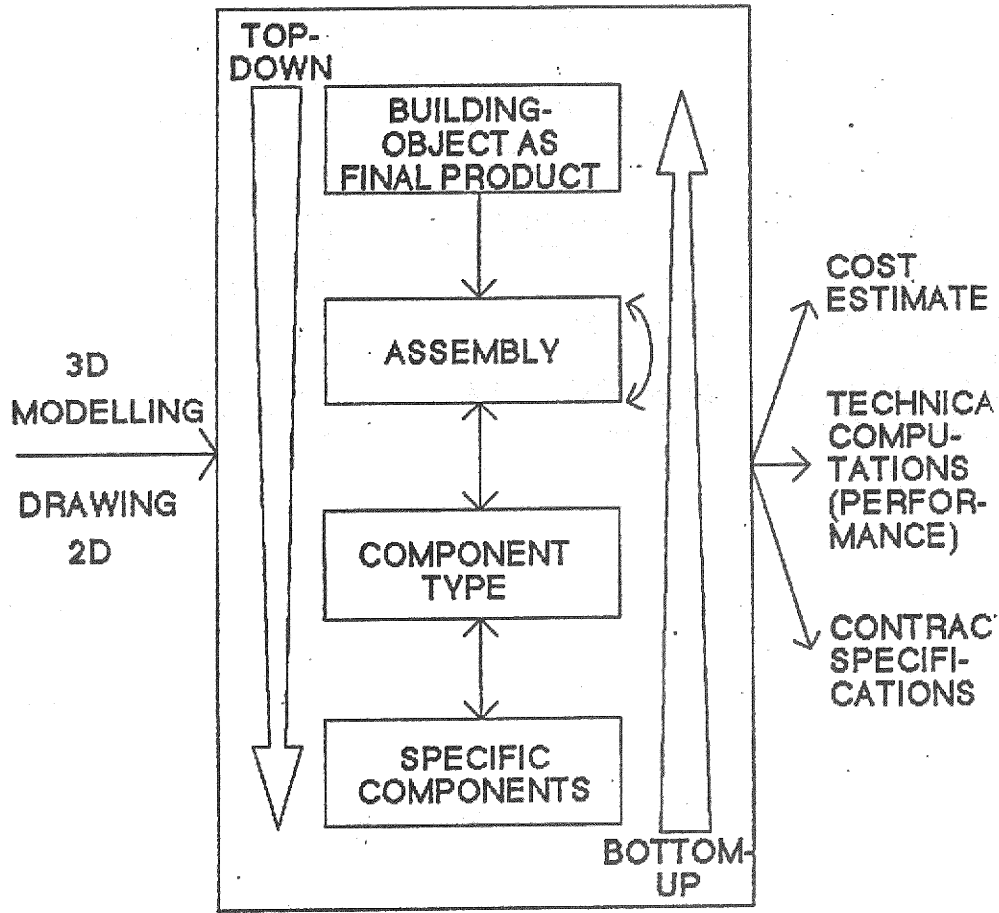
COMBINATION + ADAPTATION NEW ASSEMBLIES
 COMBINATORIAL DESIGN

ADAPTATION OF EXISTING COMPONENTS
 PARAMETRIC DESIGN

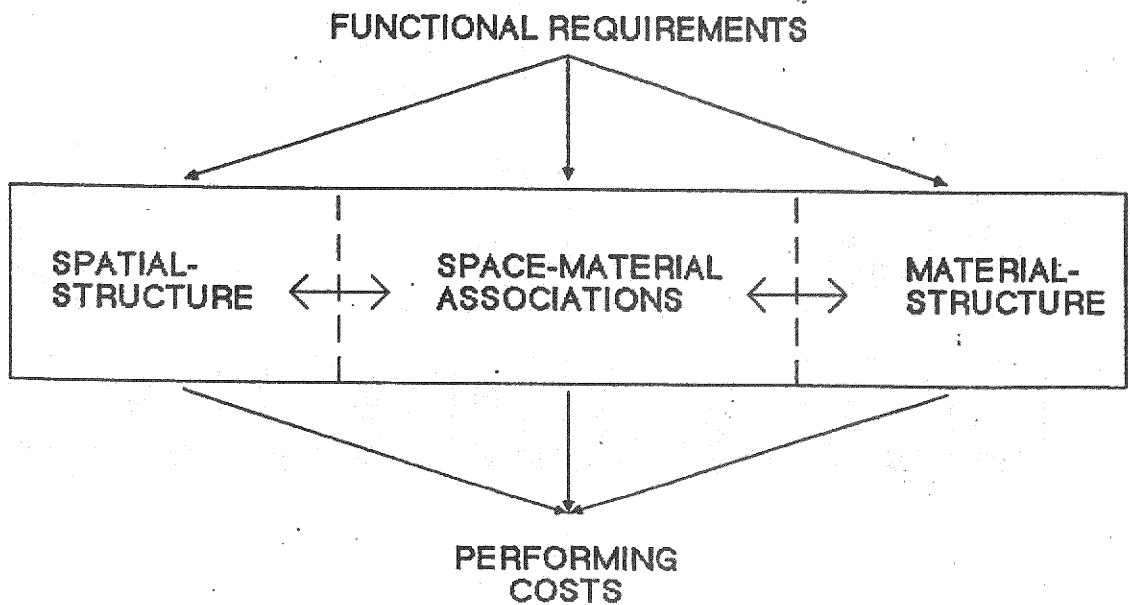
STANDARD OR CATALOGUE PRODUCTS

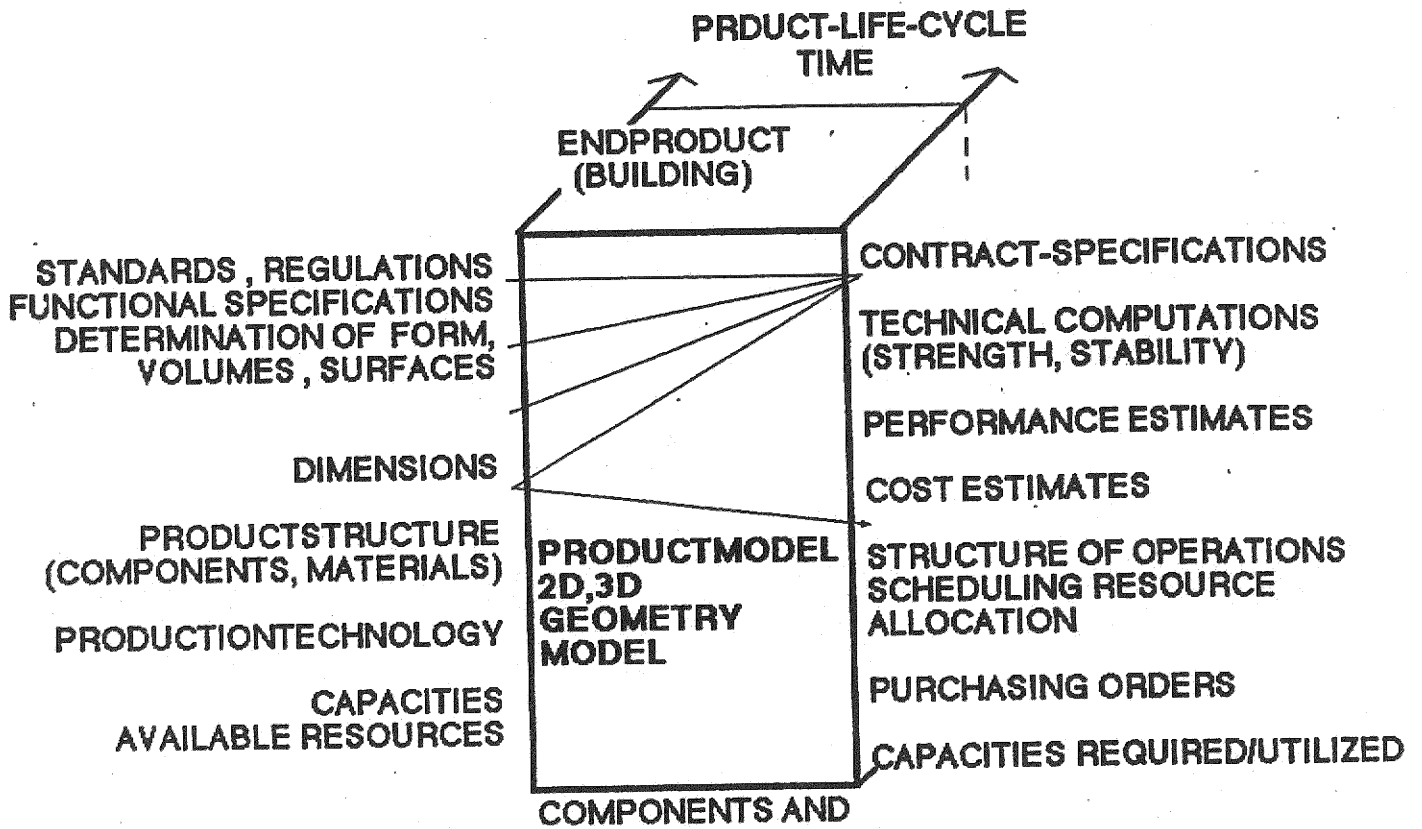


FUNCTIONAL REQUIREMENTS



TECHNICAL SOLUTIONS





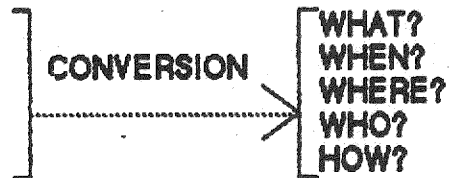
BUILDING PROCESS : IN SITU FABRICATION
 ASSEMBLY OF PREFABRICATED COMPONENTS
 MATERIALS SUPPLY AND MATERIALS HANDLING

CURRENT SITUATION

SOURCE OF INFORMATION

INFORMATION NEEDS

DRAWING
 CONTRACT SPECIFICATIONS
 PRODUCT DOCUMENTATION
 NORMS, STANDARDS
 SIMILAR PROJECTS



WHAT?
 WHEN?
 WHERE?
 WHO?
 HOW?

NEEDS FOR INFORMATION MANAGEMENT:

OBJECT INFORMATION

- + PROCESSES : ACTIVITIES, TASKS, JOBS
- + PROJECT ORGANISATION AND STATUS
- + PROJECT RESOURCES
- + ENVIRONMENT

PROCESSES AS TRANSFORMATIONS AND QUERIES OF A PRODUCT MODEL

