# Product Model: a Basis for Next Generation CAAD

*Richard Junge*, Edith Ammermann*, Thomas Liebich**

## Introduction

Even at the very beginning of the use of computers in architecture there had already been the vision to use the new medium for more than just copying the draught man's way of work [1]. Thus, the search for such a computer assisted or even computer automated design process, either for the more artistic or for the more constructive part of the architectural design process, has lasted as long as the history of CAD in architecture itself. This search follows different concepts, which are, with more or less intenseness, AI approaches. The more artistic and creative aspects are to be solved by, e.g., Shape Grammar or Case Based Reasoning, the more constructive aspects by Expert Systems. These are approaches on a very high and demanding level. It is still not clear, how far the progress will lead to usability in practice. A more pragmatic but surely arduous approach is to built next generation CAAD system on the basis of a product model. The product model, however, will be a sound foundation for using case based or expert system approaches in a further step of the development.

The NextCAAD project follows this approach. The model defines and describes the architectural design objects, their relations, dependencies and constraints. This will be done in a stepwise manner. A stepwise approach is also followed for the handling of dependencies and constraints. In a first phase the user will have to set the rules. In next phases small rule based modules will be taken over as far as they are proved useful.

## Limitation of today's CAD systems

Today's existing CAD systems process geometrical objects all together. A building object can only exist by the means of one favoured geometric representation. All the other descriptions, such as non-geometric attributes, are regarded as peripheral ones of supplementary nature only. A mismatch in regard of the real design process is the lack of support to multiple representation. In real architectural and engineering design process the representation of design objects is context dependent. A beam can be represented by a continuous or a dotted line, two parallel lines, a CSG or a photo realistic picture. Which one is the right one, the best one? The consequence is that objects have to be independent from their representation. Another limitation of today's CAD is owing to its data storing mechanism. Now CAD is often regarded as a database problem and the evolution of CAD-systems can be characterised by the storing technique, which has developed from only storing geometry in a file to storing so called attached attributes in that file as well to using RDBMS for storing those attributes.

The objective of the NextCAAD project is to develop a CAAD system, that is not primarily oriented towards the graphical and geometrical processing as current systems are. The basic idea is the architectural design object considered as an abstract entity. Architectural design objects can be rooms, spaces, building components of different complexity, grids, etc. Parts of the objects' definition are dealing with its real physical shape, physical properties, others with its representation, which is always an abstraction of the real design object.

This approach, among other results, will be a basis for enabling the penetration of CAAD into early phases of architectural design and a continuos use of the results through all phases from the early design stages to the late construction phase. It will be the basis for non architectural systems in the area of building engineering, what enable them to use the information in an appropriate representation or view. The design object approach will not only lead to a new generation of CAAD systems in the restrictive sense of the term, but to a new basis for integration, or better, cooperative systems.

## Conceptual idea

The conceptual idea for the software technology is, among others, the application of a total and consistent object orientation. What does that mean? Firstly, it does not simply mean using C++ and nevertheless remaining to deal with geometrical objects or their assemblies, which are declared to be building elements by attaching strings (as element names). It does mean applying all characteristics of object oriented analysis, design and implementation [2] to the development of CAD software. In particular it requires a common understanding of the object oriented way of thinking among all participants of the project.

The underlying conceptual ideas of NextCAAD can be described as follows:

- *design objects* versus shape objects
  The existence of an object is not based on the representation of its geometry on the screen respectively the object's definition in geometry data. This is the fundamental step, which creates the basis for a consistent application of the product model philosophy. An object is a more abstract entity that can be viewed from viewpoint and thus has different representations characterised by different sets of attributes [Fig. 1]
- Product modeling applied to building design.
  This is not the first project for building product modeling, even not the first in connection with CAD, but it is one of the first trying to use such a product model approach in a commercial development. An architectural CAD application is developed on top of autocad. Of course results and experiences made in previous or ongoing research projects are carefully studied.

- Definition of objects and classes.
  The first step in building the product model is to start with a choice of design objects, the entities in the model. The product modeling technique as well as the object oriented paradigm enable the model, and vice versa the CAD software, to be extendable by adding new entities.

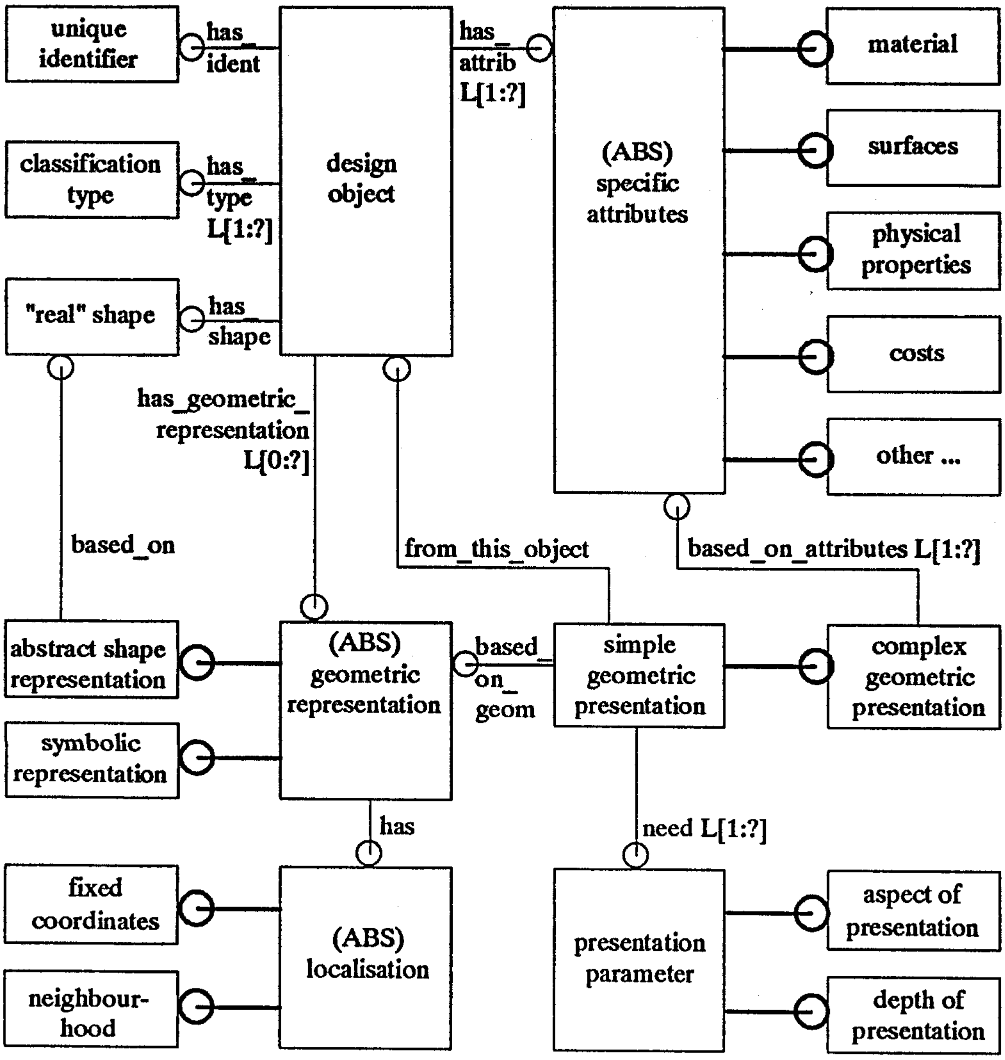

Fig. 1 The root entity "design object" with its attributes and representations

- Definition of relations and attributes.
  The entities form a kind of framework that has to be completed by defining the relations between the entities. This is an important step. Entities will know about their interrelations or dependencies. All entities of a building are dependent on or related to others. One will find examples in the architect's terminology, where he or she wants a design object to be connected to a second one by being central, rectangular or flush or all together. Some of these are to be set by the user as a first step, others follow

rules that could be processed in future versions. The attributes and the range of their values that are in scope of the project have to be defined. This is only a small section of all possible attributes of a certain entity. Here one can imagine the link between one partial building product model, e.g., an architectural model and other models from other domains, e.g., HVAC. On the other hand the relations are kept in order to keep consistency among design objects after modifications of a subset. The violation of some design rules can now be detected during a session, and the system can either ask to user for a decision or call a constraint propagation mechanism to solve the problem.

- Runtime objects (C++) and object oriented data bases.
  It surely needs no further explanations that C++ objects will be handled during runtime. Due to the fact that Autodesk does not currently provide an interface to an object oriented data base, the obvious task of intensively using such a data base for storing the product model related data had to be postponed after some experiments were done.

**Some guidance to methodology**

Before the project actually started, some requirements had been set up, and in order to meet them some basic concepts had been chosen. Most of them, at least in principle, were proved to be the right choice. In some cases, however, the concepts were too advanced for immediate commercial realisation. It is expected that time will work for these currently postponed concepts. At least these delays are partly connected to the current kind of object oriented storing technique provided by autocad, which is the first implementation basis for the NextCAD project. These problems are momentarily not on the critical path of the project. Developments at autodesk, at some data base developers and others are pulling into the desired direction. Some of the concepts, however, have been proven to be so stable that they could form a kind of guideline. These are the following:

- Paying attention and observing relations to STEP especially to [3]:
  - shape -- topology and geometry (AP 42)
  - explicit draughting (AP 201)
  - associative draughting (AP 202)
  - explicit shape representation (AP 225)

  It seems to become obvious that STEP (officially: ISO 10303) is simply the emerging standard for data exchange and for data sharing in the architectural / engineering domain. The latter will be even much more important in future. Besides its role in standardization STEP methodology is an attractive focus point for many object oriented developments. It makes sense not trying to make own developments or to use techniques that are not more or less connected to the STEP-World.

- Modeling and documentation in EXPRESS-G / EXPRESS
  EXPRESS is the data definition language of STEP [4]. It is also the most used language for the purpose of data definition respectively product modelling. It is a very important advantage hat there are a lot of tools are available in the environment of STEP / EXPRESS. For reasons of easiness of readability for human beings a graphical method of displaying product models is of high importance. This graphical capability of EXPRESS is EXPRESS-G.

- Processing rules and where clauses from EXPRESS schemes for maintaining consistency of the models.
  As indicated, one of the objectives of the project is to bring "intelligence" to Computer Aided Architectural Design. One of the possibilities to facilitate this is to define rules for relations, dependencies between entities and to define the range of attribute values by using where clauses. By doing so, the current state of entities can be controlled.

- Generating C++ classes, structures of relational data bases and object oriented data definition structures from EXPRESS schema entity classes,
  A product model written in EXPRESS contains all definitions of the data: their semantic, relations and structure in a computer readable form. It is obvious that there must be way of an automated interpretation, conversion or interfacing to other computer usable forms. And in fact the availability of such tools is a real progress in application of CASE. The behavioural aspects of objects can not be described in EXPRESS, they have to be added by appropriate CASE tools.

- Realisation of the total CAD functionality as methods on the objects.
  The role of the CAD software itself will be transformed step by step. In addition to the traditional role focused on generating and manipulating geometry it now has to fulfill functions concerning the design objects, which are as mentioned not geometric object from origin.

## Status of modeling work

The following model parts or schemes have been realized or they are currently under development. (Fig. 2 gives a condensed overview about the description of openings):

- grid
- building components (first samples, such as wall, column, beam, foundation, etc.)
- geometry subset for location and basic parameter
- spaces
- catalogues

## Conclusions

The project was conceived as a long term development with a long term concept from beginning, nevertheless there where lessons learned during the last one and a half year. Some of these were expected more or less. In any case some conclusions can be written down:

- The definition of a building product model that fulfills the requirements from practice is a very ambitious task, it is complex as well as work consuming. (And the project is still concentrated on architectural domain)

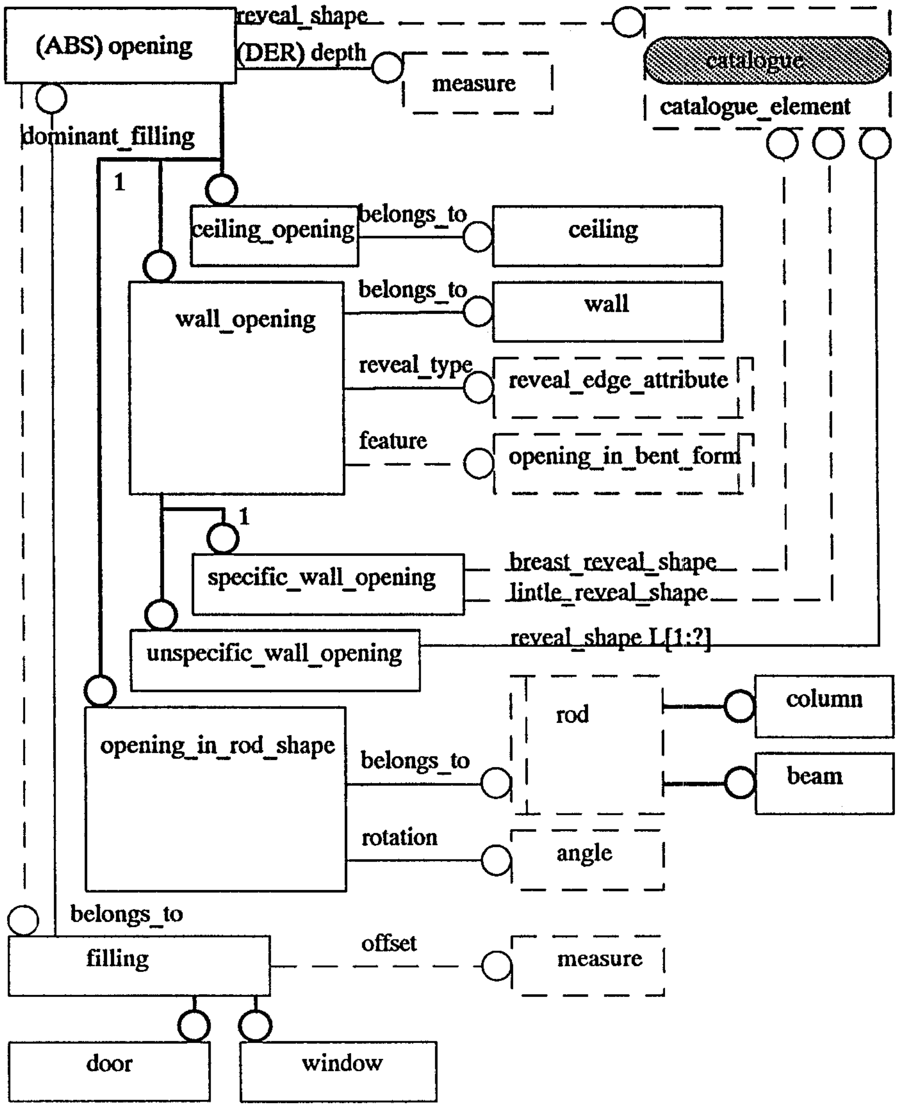

Fig. 2 the entity "opening" and its subtypes

- EXPRESS has been proved to be a sufficient as well as effective language for object modeling.

- It is pragmatically and practically to refer to STEP geometry and topology for the purposes of modeling "gestalt". But it is recommended to intermediately use more condensed aggregations of what STEP is providing until today. They have to be more specific to building design objects and should avoid "deep structures" of links in order to fulfill CAD specific demands on runtime performance for commonly available hardware platforms.

- Object oriented development of CAD systems allows new objects respectively classes to be integrated relatively easy into new versions of the software. This allows such a system respectively the design objects it is using to grow stepwise.

- Object orientation beyond graphics means: capturing of so called real world objects, inclusively their definitions and their relations. That has irresistibly to lead sooner or later to the adjustment of different individual concepts.

The first phase of NextCAAD project is now considered to be a successful step into the direction of intelligent CAAD. During the next phases the emphasis will be put more on incorporating intelligent modules, which will base on the object oriented description of design objects, provided by the underlying product model.

## References

[1] N. Negroponte, 1970. *The Architectural Machine: Toward a more Human Environment*, MIT Press, Cambridge (Mass)

[2] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, 1991. *Object-Oriented Modeling and Design.* Prentice-Hall : Englewood Cliffs

[3] ISO 10303 Product Data Representation and Exchange
Part 42 *Integrated Resources: Geometric and Topological Representation,* Part 201 *Explicit Draughting,* Part 202 *Associative Draughting,* Part 225 *Structural building elements using explicit shape representation*

[4] ISO DIS 10303 Product Data Representation and Exchange - Part 11, 1992. *The EXPRESS Language Reference Manual*

* Richard Junge, Edith Ammermann, Thomas Liebich work at CAB, Osterwaldstr. 10, D-80805 München, Tel. +44/ 89/ 369030, Fax +44/ 89/ 363801, e-mail junge@cab-muenchen.de