Aspects of new CAAD environments

Richard Junge
Professor for CAAD Technical University Munich

Abstract

This paper views on a new generation of CAAD systems from two angles. First the architect view as user describing shortcomings in today's systems and articulates some representative examples of desirables. These are merely stemming from the fact that today's CAD is limited to processing geometrical objects. Design however is more than drawing. The second part presents some basic concepts for the new CAAD. Some of these concepts are already implemented in such a new system others are still in a conceptual phase.

1. Introduction

There are many approaches and concepts for new architectural design computing solutions existing in the research community. Some few examples mentioned are design grammar, artificial intelligence in design, cased based design, etc. Some of them are aiming at more general, some at more partial solutions. These concepts are on one hand not interconnected in a natural way but on the other hand, because of the fact that they all provide interesting aspects, it is of interest to look for an approach that would allow to tie them together, integrate them in one system.

When looking back in the history of design system, i.e., systems that are not pure tools for creating and manipulating geometry objects, the starting point can be found in a generation of systems like GLIDE and BDS [Richens, 1978] and many others. The next period is defined by a first generation of expert systems [Landsdown, 1983], [Wager, 1984]. Both approaches are using data models. Today one would say they where using Building Product Models.

A certain degree of consensus, e.g., that an integrated building design system has to be structured around building parts and not around geometry objects was already reached in the early years of building product model research [Eastman, 1978]. Nevertheless it took more then a dozen further years until the structures of building product models have become clearer.

The starting point for the development of Building Product Models seems to be the GARM (General AEC Reference Model) [Gielingh, 1988] that was developed for the STEP project. The GARM provides a first basic concept by introducing the 'Product

Definition Unit' (PDU) and its subtypes 'functional unit' and 'technical solution' of the GARM [Gielingh, 1988]. The Building Systems Model [Turner, 1990] shows a top down strategy to model a building using functional systems as, e.g., enclosure, structural, mechanical, etc. and their entities. The Neutrabas project [Neutrabas, 1991] is one of those that started to deal with the question of entities, belonging to more than one system.

That leads directly to the question how a building product model that would cover the needs of all parties involved has to be structured. Today the belief that there is no way leading to a homogeneous single central building product model is accepted nearly unanimous. In a number of current projects one finds various terms used for strategies dividing the universe of discourse into partial models. Terms used to describe such partial models are 'topical model' [van Nederveen and Tolman, 1992]. The ESPRIT project ATLAS has 'view type models' [ATLAS, 1994] and ESPRIT project COMBI uses 'partial models' and 'application models' [COMBI, 1994]. Although the solutions are slightly different a way has been found that leads to practical building product models. Currently, the discussion focuses on the question, how to connect these partial models?

The discussion is on strategies where these partial models should be compatible with each other. The idea is that there is a "central" part of the modeling domain that would be shared by all partial models [Luiten et al., 1991]. These central parts are often called kernel or core models. The core model concept is an underlying principle of the framework for Building/ Construction development in STEP, that is the B/C Application Protocol Planning Project [Junge and Storer, 1993]. This framework structures AP developments for Building/ Construction into families, such as building services, architectural system, structural system, etc.. These families are integrated by family cores. The Building/ Construction Core provides integration among these families. There is also the idea of an AEC core to integrate among all disciplines in the AEC domain. The Building/ Construction Core is currently under development as Part 106 in STEP [Tolman and Wix, 1995].

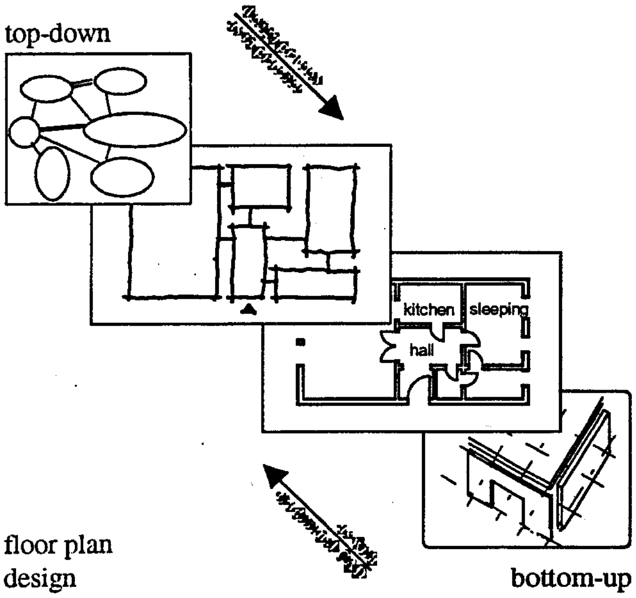
A question is, however, could a Building Product Model be the long term backbone for integrating the various concepts for intelligent architectural design tools into integrated environment.

2. Wish list, or how architects work

There is a clear mismatch between the way architects do their design and the way traditional CAD forces them to work. This mismatch certainly stems from the fact that todays CAD brings the skills of drafting to the computer and that is also true for 3D CAD. The centerpiece of these systems is the ability to create and manipulate geometry objects that simply are labeled with names. A label, for instance, wall or beam does not make more difference to the computer than that these two strings differ. They are both still pure geometric objects. An architect or engineer however when designing does not think in geometric objects but in objects of his design task. A computer could become a design assistant only when he would know about design objects and not only about drafting objects. For a new CAAD environment it is the first prerequisite that the system would behave like a design assistant and not as a drafting tool. In the following are some examples of how architects work

2.1. Aspect of continuous refinement

The design process starts in top down manner. The starting point is a building program or design brief. This is the first definition of functions to be fulfilled from the design. These functions will be physically fulfilled in rooms and groups of rooms. The first steps are diagrams showing the functional relationships and adjacencies. In these sketches' rooms have a symbolic geometrical representation. Next steps are dealing with the overall layout of the building and the ordering of the room layout following the diagrams. Now rooms start to have a geometry as known from preliminary design floor plans, but their boundaries are still in a symbolic form; lines without any differentiation. The differentiation into boundaries becoming physical walls or non physical boundaries is made in the next step as are the first decisions about the wall types. Especially that work continuos with dimensions, material properties, etc., until the design is mature for construction. This is of course not a one way process, there are loops again and again.

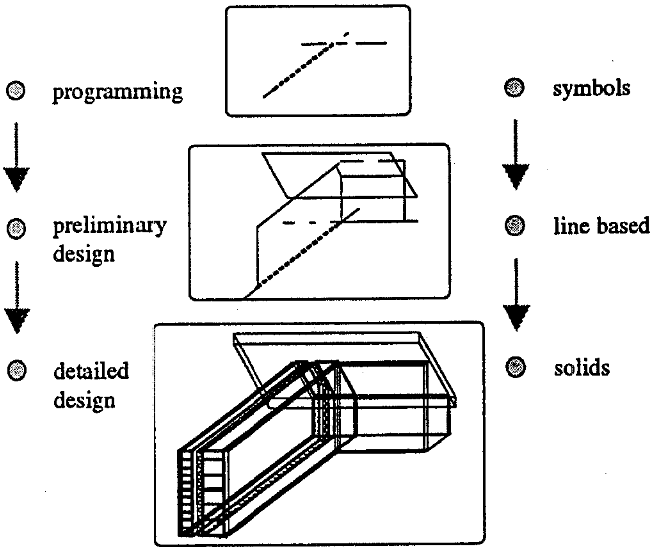


Aspect of continuous refinement

This process of continuos refinement starting from first conceptual ideas with only vague representations possible to more and more precise definitions of the design is a top down approach. This is in contradiction to the bottom up approach of todays CAD that forces the designer to start with defining walls, where his intention is a layout of functions physically fulfilled in rooms.

2.2. Aspect of multiple geometric representation

During the above described process of gradual definition of the design its representation and presentation have many expressions. It starts with simple symbols in early phases and goes to very precise ones in late phases. The representation and presentation are not only dependent from phases in the design process but also from engineering disciplines. There are many valid representations of a design object coexisting at the same phase or time. For a certain task in structural analysis, for example, it may be appropriate to have only the center line of gravity and not the maybe known 3D definition of a strut. The representation of design objects is dependent from phases and disciplines, from the context in which they are used.

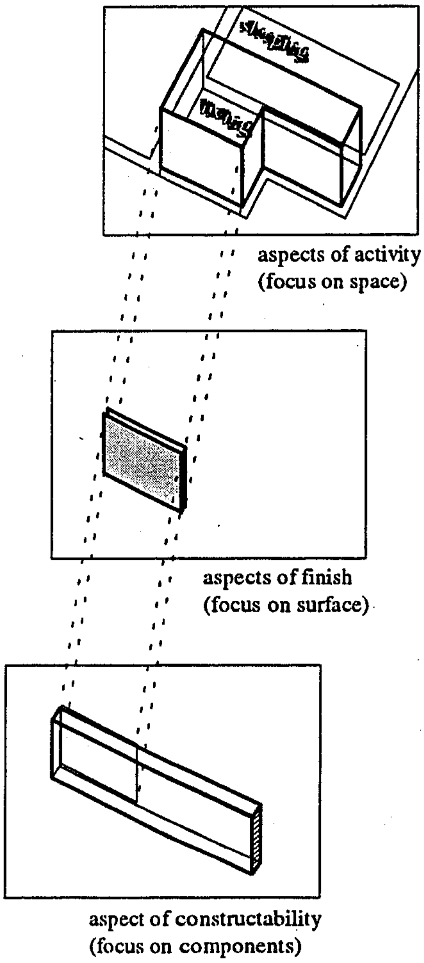


Aspect of multiple geometric representations

Traditional CAD provides only inflexible means to allow for multiple representations, such as sophisticated layer and macro techniques.

2.3. Aspect of continuous specification of the design object

So far only the specification of the design objects form or Gestallt was mentioned, which covers only one branch of the total task. An architect thinks during his design process, which seems to be merely geometry oriented, in a second, parallel stream. This stream is dealing with all the non geometric properties of the design object and these properties are, compared with the Gestallt properties, almost the majority. Also these properties are defined in a stepwise manner during the design process. For instance for defining a bathroom dimensions one has to specify the bathtub's size (taken from a producer's catalogue), the kind of wall finishes, e.g., glued tiles and type of joint between tub and wall. All these specifications are made but they are not stored in the drawing and not in the CAD data base. They are almost lost, unretrievable. This information that has to be reconstructed later for the BoQ for example.



Aspect of dynamic shift of focus

In todays computer assisted design environments from this point of view two shortcomings are existing, which seem to be inherent in the current technology. First, despite all integration efforts there is an almost unbridged gap between CAD and the non geometrical side of the design tasks. Second, today's specification systems allow only for very structured input mostly in a relatively complete stage of the specification. They are not designed for the stepwise mostly unstructured manner that is characteristic for design tasks. What is needed is a system to support the continuos specification flow, able to follow the almost unsorted pieces of specification as they emerge during the design process. That is a missing link to the natural design process.

2.4. Aspect of dynamic shift of focus

During the design task the designer's interest is constantly shifting between various aspects of the design. So for instance at one moment he might be working one the functions to be fulfilled in the building, in a specific room and the relating spatial needs and adjacencies. In the next moment he might shift to the architectural appearance of the spatial arrangement. That might especially be the surfaces, or more precise the finishes of the space boundaries, their structure color, etc.. The next focus might be the bounding wall or wall core itself.

That is of course not a problem as long as all these three design focusses are only parts of the geometrical objects with the label "Wall". The question is what are the implications if these three are objects of their own, existing in separate systems as they are doing in the designers real world.

2.5. Aspect of communication, integration

The design is not done by only one designer, e.g., architect, alone, it is done by a design team, which for the purpose of a coordinated design ('coordinated' for avoiding 'integrated') has to have a very intense communication. All design tasks off design team members are highly interrelated and dependent on each other. The basis for this communication between humans is a to a certain degree common understanding, for example, between architect and structural engineer about a load bearing concrete wall. They have to implicitly know the implications of that fact for their own design task. In both knowledge domains there has to be an overlap covering their mutual discipline views. Or in other words, their conceptual models have to have a solution for the different views that they have on one design object. That is the basis for their communication on a semantically very high level [Junge, 1991].

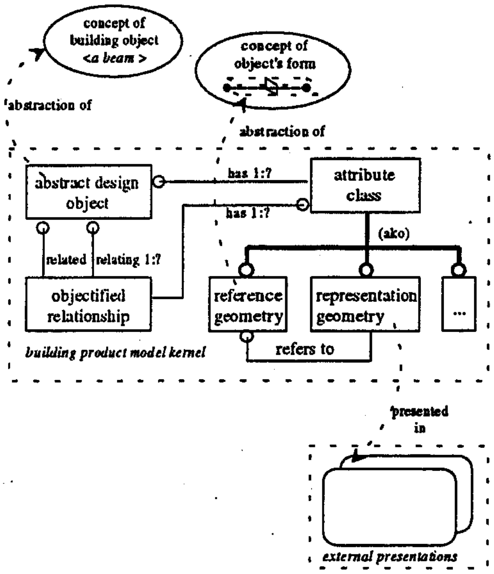
A design computation environment, able to assist the design team, has to provide a means to communicate on a comparatively as high semantical level.

3. Some concepts for a BPM based CAAD system

The product modeling approach is considered as a sound foundation for developments that should lead to data exchange, data integration and to new design applications, such as future CAAD systems [Junge and Liebich, 1994]. Both the new applications and the data exchange and integration facilities should be enabled to deal with semantically meaningful elements rather that with geometric primitives.

3.1. Abstract design object

The basic construct of the product modelling work, the abstract design object, is independent of its geometrical representation. An abstract design object is the highest abstraction of any architectural object, for example; building component, space, grid, or catalogue item. Each of these architectural objects is later regarded as a specialization of this root object. It is described at different levels of complexity by the definition of its attribute classes, that is by its name and, if available, by its geometrical and physical properties, relationships, dependencies and constraints. Thus, the abstract design object allows the equal treatment of form descriptions and specifications, e.g., material, costs, building instructions, or performance attributes.

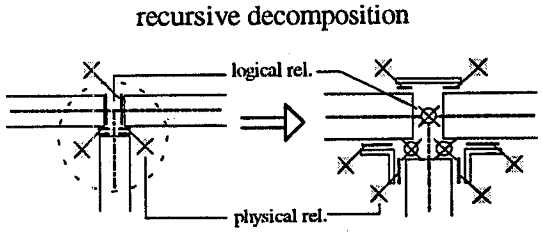


the definition of the abstract design object

The design object has a logical and a physical state, whereas the logical state refers to abstract definitions, such as the reference geometry or the logical relationships, such as connected_by, bounded_by, or adjacent_to relationships. Those definitions establish the network of relationships among the design objects, and they influence but not directly describe the form of design objects. The form of a wall being connected to other walls at its ends is geometrically defined by an axis and a cross section. The polyhedra forms a particular three-dimensional representation, that is developed from the reference geometry and the functionality of the connected_by relationship. The physical state refers to the detailed development of the representation form, such as the representation geometry. Thus, the logical and the physical states can partly be compared with the GARM approach, that distinguishes between the functional unit and the technical solution of design objects.

3.2. Objectified relationships

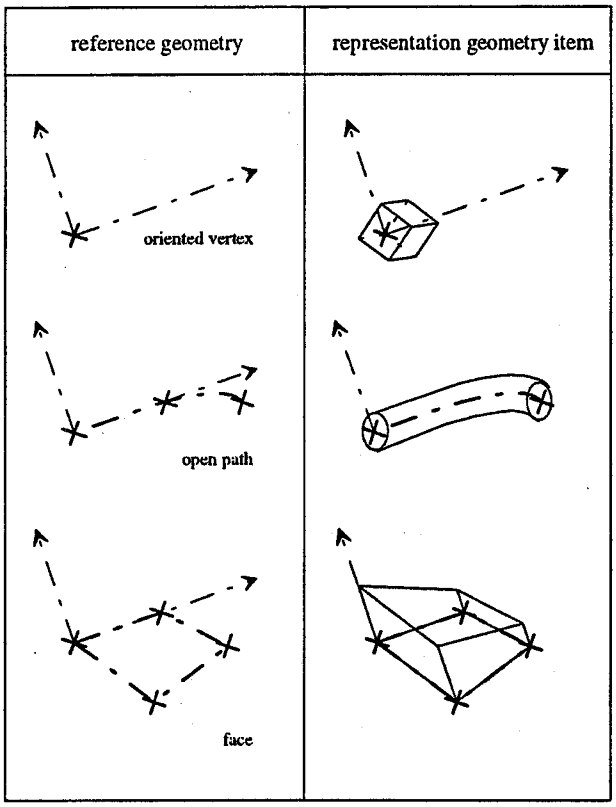
Beside the design objects the objectified relationships are considered as root objects as well. Objectified relationships deal with different kinds of interrelationships between design objects, such as dependencies, connections, or aggregations. They are objects by their own rights. The objectified relationship is described by attribute classes to keep the specific properties. The support of a beam on a footing is a relationship having its own attributes, e.g., the type (free or restraint) and the bearing pressure. Objectified relationship has a logical and a physical state as well. The comparison to GARM applies particularly to the linkage relationship, as a logical relationship can be further decomposed into several physical relationships. This process can be recursive, if any of the physical relationships acts as a source of a new logical linkage. At the logical state the bounded_by relationship points to a space and defines a virtual space boundary, with a connection to the space behind. At the physical state, however, this relationship is objectified into a space enclosing element, being a view type object of the building element that actually forms the physical boundary, such as a wall.



recursive decomposition of logical and physical relationships

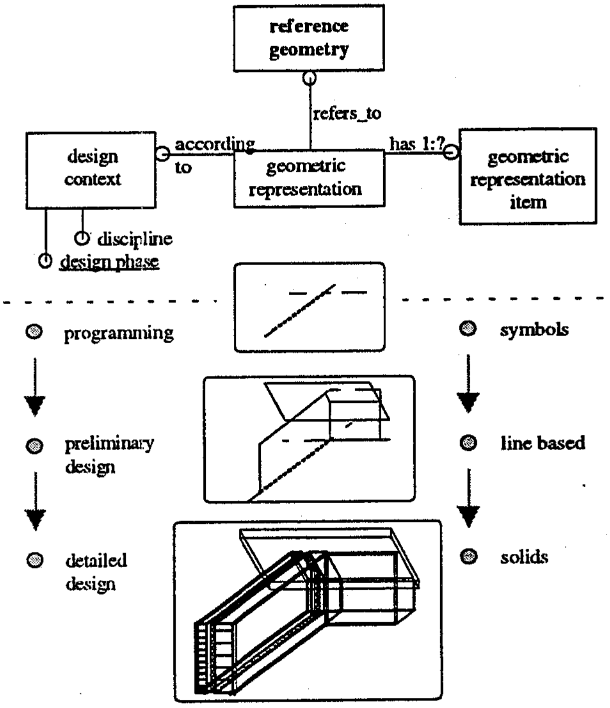
3.3. Reference geometry

A prerequisite for the logical description of design objects is the reference geometry. The reference geometry contains the basis geometric definition of a design object and places it into the spatial context of a building. It can be considered as an implicit geometric definition, as it defines a geometric reference item to which either a parametrized description or an explicit geometric representation item refers [Figure 3]. Whenever a design object, e.g., a building element or space, is placed into the coordinate system of the building project, exactly one reference geometry item is created. Its type, however, can change during the design object's life cycle. The sum of reference geometry of spaces builds the skeleton topology of the building. The Relational Model Topology of COMBINE follows a similar approach [Combine II, 1995].



different types of reference geometries

The reference geometry is the point, to which all different geometric representations refer. It is either possible to develop a reference geometry under a given context into geometric representation items, or to connect any explicit graphic or geometric representation form to the reference geometry. In an application, the geometric representation can now change from vague and fuzzy symbols to sketch-like two-dimensional geometry and again to exact two- and three-dimensional geometric entities. All those geometric representations are valid design object descriptions, as they refer to the reference geometry and thus to the same location within the spatial context of the building.



different context dependent geometric representations

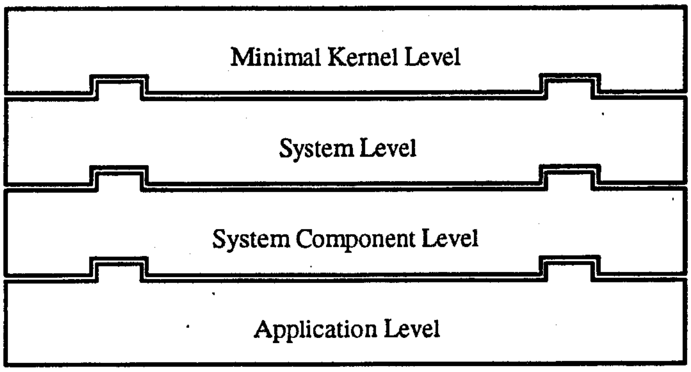
3.4. Building product model decomposition

The abstract design object, the objectified relationship, and the reference geometry are fundamental concepts for defining a building product model applied to design applications. Another prerequisite is the proper structuring of the building product model.The layered approach is certainly the favored way to decompose the whole modeling universe into levels of specific information contents. According to the core model philosophy, the following structure is chosen:

- minimal kernel level
- system level
- system component level
- application level

SPECIALIZATION
EXPLANATION/DEFINITION

It is the task of the minimal kernel to establish a common agreement for all models, that are being incrementally defined. The minimal kernel establishes the basic description of the root objects, such as the design object and the objectified relationship. Those objects are further specified in partial data models at the system and system component level. Another task of the minimal kernel is to keep track of the links between the different partial data models.



layered structure of the modelling framework

At the next level, the different systems are described, e.g., structural system, enclosing system, spatial system and distribution system. Those partial models for systems contain the core description, that applies to all components of the system. The third level comprises schemas for the components, that play their major role in the according partial model for systems. Those components are fully described in terms of their attributes and relationships. As the definition of system components, such as spaces, significantly varies during the life cycle, a further decomposition into life cycle stage dependent partial models for system components might be appropriate. The forth layer of application models uses modeling constructs from the layers above and adds the application specific functionality. The application layer should also allow to further expand the design object hierarchy to create more detailed and user specific subtypes of design objects.

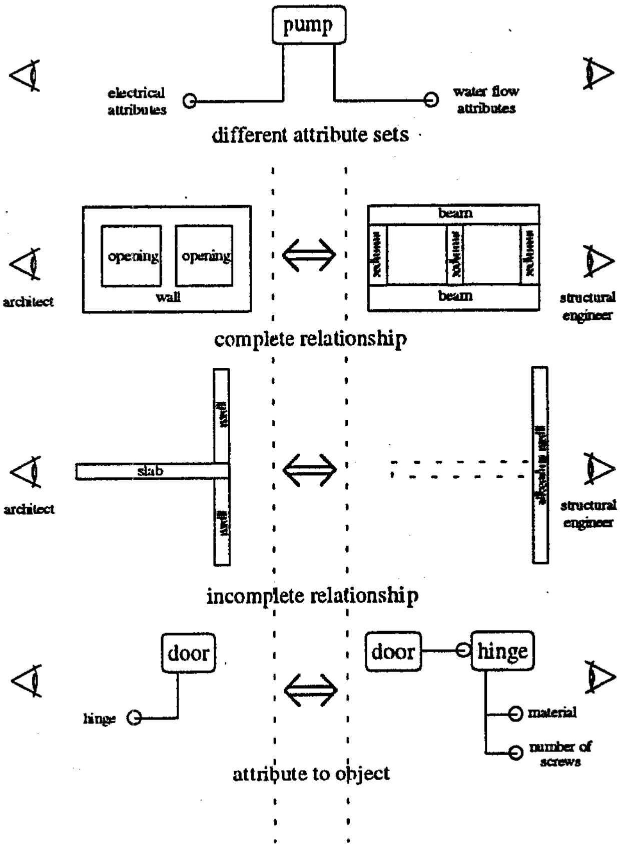
A problem arises with the multidisciplinary nature of many system components, playing a role in several partial models for systems. Each partial model manifests a different way to cope with the specific views on physically identical design objects. This reflects the human way, as the various participants of the design process have different conceptualizations of the design object. There are several relationships between the different abstractions, based on different views onto the physically identical design object:

a) Different attribute sets of the design object are kept only once in that partial model, where the design object plays its major role.

b) Different design objects refer to the same physically identical design object, but they have distinct abstraction in two or more different partial models. The relationship between them can be a one-to-one or a many-to-one relationship. The differences, made within the partial models, can require:

- a complete relationship, i.e., the design object within the first partial model is fully decomposed into many design objects within the second partial model
- a incomplete relationship, i.e., the design object within the first partial model is also decomposed into sections of other design objects within the second partial model
- an attribute to object transformation, i.e., the same information item is described as an attribute of another design object within the first partial model, whereas it is an own object within the second partial model

Those view type objects can dynamically evolve as type changes of the original design object during the design process. If the original object should be kept, since it reflects a specific view or it manifests the design history, a relationship between the two view type or version objects has to be established.

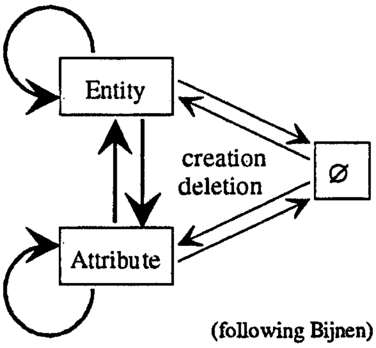


example of different views onto physically identical design objects]

3.5. Product model integration

The establishment of a collaboration between diverse application or partial models requires a model translation, both at conceptual and at implementation level. The translation process needs the input of technical knowledge to relate the semantic description of the source object to the description of the target object. Thus, the conceptual model is necessary to allow the modeler to specify the model correspondence on a conceptual level as well. Another advantage is that changes can be made at the specification level, and the actual converter is then automatically generated.

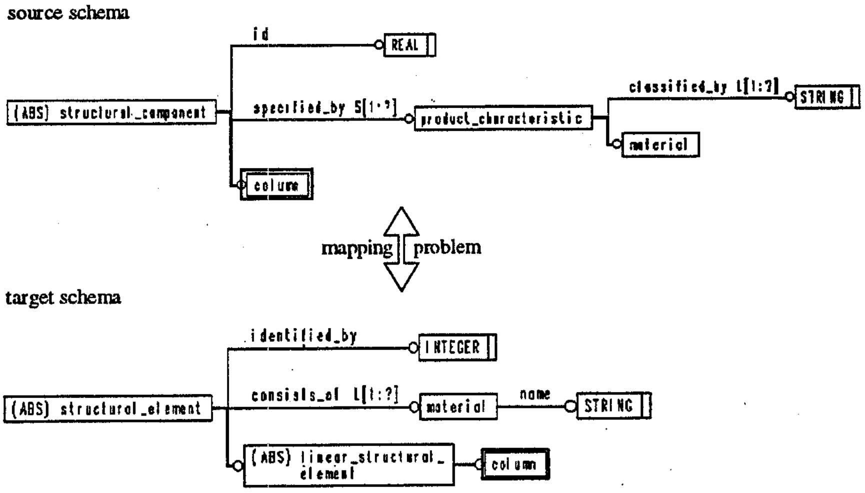
There are several formal translation languages being currently under development. Most of them accept two schemas, source and target usually specified using EXPRESS, and establish the transformation process either by imperative statements or by rules, e.g., in languages based on Lisp or Prolog derivatives. Those conceptual translation languages have to cope with converting mapping primitives (entities and attributes) in any combination and with any possible cardinalities. They should provide for translations in one or both directions. Functions and constraints, associated with entities, should also be converted in a more advanced stage.



mapping primitives

The complexity of even simple translation problems is shown in the following example. An entity 'column' within the architectural tool has to be converted into the different abstraction of column within the structural engineering tool. This requires the translation of the unique identification, the material information, and the different geometric descriptions, whereas only the first two are now described and given in the schema descriptions. The translation of the identification needs a mechanism for the handling of synonyms, since the attributes have different names. The second translation problem comprises the conditional conversion and merges a set, containing embodied subsets, into a single set. Only those specified_by of the column in the source schema, which refer to a material, have to be converted into consists_of of the column in the target schema. As the source entity material is described by a list of strings, it has to be translated into many target entities material, being described by a single string.

The decomposition of the data integration framework into small and easily manageable schemas already reflects the different views of design actors. Model translation can be used to realize data integration within such a heterogeneous model world. The differentiated model structure, however, also provides a background for collaboration, that goes behind the bulk exchange of design information. At a next stage, the design actors should be enabled to collaborate by the frequent exchange of distinct information on request, realized by the transformation of required data between the according partial models.



EXPRESS-G diagrams of the translation example

4. Conclusions

Design is more than drawing. As stated in the wish list there are shortcomings in today's CAAD systems. These systems are limited to processing geometrical objects. From watching the evolution of the software, the new versions each year, one can discover that the innovation curve is leveling off, while at the same time there some demands for new features that are very arduous or even not to fulfill. It seems that todays CAAD systems have reached a point where a paradigm shift is inevitable. At many places, not only in research, one can observe activities leading to new directions in CAAD.

CAAD really is momentarily in an exceptional phase, at the verge of a new generation based on a new paradigm. It is of importance that developments will not go half-hearted into the new direction rather to build completely and consequently on the paradigm.

Bibliography

ATLAS, [1994]
ESPRIT 7280 project: various deliverables

Bijnen, A [1994]
Operation mapping or how to get the right data?
First ECPPM Conference, A.A. Balkema publishers, Rotterdam 1995

COMBI, [1994]
ESPRIT 6909 project: deliverable A2

COMBINE II, [1995]
EC joule project. final report

Eastman, C., [1978]
The representation of design problems and maintenance of their structure.
Latcombe (ed.), Application of AI and PR to CAD, Elsevier, North-Holland

Gielingh, W., [1988]
General AEC Reference Model, ISO TC 184/SC4/WG1 DOC N.3.2.2.1

Junge, R., [1991]
Integration by Communication.
1st international symposium Buillding System's Automation - Integration, Conference Paper, Univ. of Wisconsin, Madison

Junge, R. and Liebich, Th., [1994]
Produktmodellierung im Architekturbereich.
Abstracts of IKM, HAB Weimar

Junge, R. and Storer, G., [1993]
B/C Application Protocol Planning Project. ISO TC 184/SC4/WG3/T12 AEC Working paper

Luiten, B., Luijten, B., Willems, P., Kuiper, P. and Tolman, F.P., [1991]
Development and Implementation of Multilayered Project Models.

Mangin, J-C., Kohler, N. and Brau, J. (edts), Computer Building Representation for Integration, Pre-Proceedings of the second international workshop, Ecole de polytechnique federale de Lausanne, Lausanne,

Landsdown, J. [1983]
Expert Systems: Their impact on the construction industry
Report to RIBA Conference fund, London 1983

NEUTRABAS, [1991]
ESPRIT 2010 project: Outfitting Information Model, Deliverable 4.2.2

Richens, P. [1978]
The Oxsys system for the design of buildings.
3rd International conference on Computer in Engineering and Building Design CAD78.,
Ipc science and technology press, UK

Tolman, F.P. and Wix, J., [1995]
ISO TC 184/SC4/WG3/T12 AEC, Part 106 Working paper

Turner, J., [1990]
Building Systems Model. ISO TC 184/SC4/WG1 Working paper

van Nederveen, G.A. and Tolman, F.P., [1992]
Modeling multiple views on buildings. Automation in Constuction, vol 1, 1992 nr. 3

Wager, D. [1984]
Expert systems and the construction industry.
CICA, Construction Industry Computing Association, Cambridge UK