Modeling and integrating design data from experts in a CAAD–environment

Peter C. Lockemann, Jutta Mülle, Rose Sturm
Institut für Programmstrukturen und Datenorganisation, Universität Karlsruhe, Germany

Volkmar Hovestadt
Institut für Industrielle Bauproduktion, Universität Karlsruhe, Germany

ABSTRACT: Computer support is strongest for routine tasks and weakest for creative activities. The ArchE project at the University of Karlsruhe attempts to overcome the weakness by developing special object-oriented database techniques in order to support human architects and other experts joining the building process. The main challenge is to replace the traditional requirement for rigid structural and procedural solutions by one that initially allows for much weaker formalizations and subsequently for their stepwise tightening. The paper will introduce a corresponding modeling approach.

1 INTRODUCTION

Like CAD in general, computer aided architectural design (CAAD) should be able to rely on an integrated design environment. An important factor on the way to integration is the varying degree of formalization along the design life cycle. In the early phases of building design the system has to support creatively working architects who will tolerate only a loose framework. These phases are characterized by inconsistencies which have to be tolerated, incomplete and inconclusive data, and, hence, unstable structures of the data, that can hardly be used as schema information in a database system. As the design process evolves and an increasing number of experts becomes involved, the data become much more clearly structured, the knowledge become more concrete and more rules are proven to be valid. Moreover, the design process is not a linear task but is organized in an iterative way characterized by trial–and–error, so that it should be possible to flexibly move back and forward along the degrees of formalization.

In today's design environments most experts are working on their own. They only communicate by blueprints, so that conflicts cannot be deleted automatically and the experts concerned cannot be notified. Communication can be improved by a more integrated design methodology, the so-called "Integral Planning". The goal of Integral Planning is to involve all experts as early as possible so that a high degree of parallelism will be achieved. Therefore, integration and early conflict detection and location between different experts is necessary and is one of the reasons for computer support.

The ArchE project ([2, 4]) at the University of Karlsruhe attempts to provide this kind of support by developing special object-oriented database techniques for the use by human architects and other experts joining the building process. The main challenge is to replace the traditional requirement for rigid structural and procedural solutions by one that initially allows for much weaker formalizations and subsequently for their stepwise tightening.

The central idea is a 11-dimensional design space called A4-space ([3]). The A4-space allows a flexible representation of loosely structured data in the early design phases. Moreover, the dimensions are typically needed in every design decision, and this is also true for later design phases, where the design information structure becomes more elaborate. With this property we get the chance to overcome the main problem of integration efforts, i.e. to get an integrated product model. In essence the A4-space provides the framework for an integrated product model accompanying the entire design process.

The remainder of the report is organized as follows: First, we give a brief overview of the concept. Then, the realization based on an object–oriented database system is shown. Finally, an example demonstrates the use of the concept.

2 CONCEPT

The main idea behind our concept is the extraction of a data structure involving the data common to all design decisions. This data kernel is equally available to all experts and functions as a basic structure to detect conflicts between the expert tools. The choice of our data kernel is strongly influenced by the A4–design space ([3]). The basic assumption behind the A4–space is that the architectural design takes place in an 11–dimensional design space, and all design decisions can be organized within it. The first three dimensions represent the *geometrical* attributes of a design decision. Each design decision is usually characterized by its spatial extent. The three geometrical dimensions describe the extent of the design decision as a bounding box around the design object. Further dimensions describe the *temporal aspects* of the design decision, e.g. the time of validity with respect to the building's life cycle, and the time the decision is valid in the design database. Besides these continuous dimensions there is a variety of discrete dimensions, e.g., the *user* being responsible for the design decision, the *alternative* (or version) of the design object, the *resolution*, and the *size* (or scale) used to investigate the design object. The schematic description of the dimensions is organized as a so–called "container schema". Instances of the container schema are called containers. A container together with the schema forms a container model and corresponds exactly to one design decision. Each container may be visualized as a hypersurface in the A4–space, because not all dimensions are mandatory.

Besides the (container) data as the aggregation of all common design decisions, there exists information, that describes design decisions local to a specific expert tool. The goal of our project is to allow the integration of expert tools at arbitrary points in time into the design process. Therefore, the common data kernel, i.e. a container model must, for integration purposes, be augmented by a concept to represent the expert tool specific data. This is achieved by dividing the design data of an expert tool into two parts, a part belonging to the container model, and a second owned exclusively by the expert tool (its special model). Take as examples of the latter the maximum velocity of circulation of a pipe, or the material properties of a pillar. The two parts together form a partial model (of the expert tool). The union of the container model and the partial models constitute the product model of a design. Figure 1 shows the relationships between the various models.

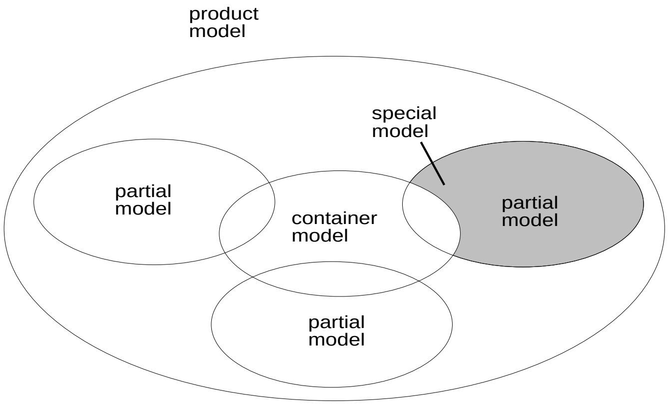

Figure 1: Relationships between product, partial, and special models

How do we now use this concept to meet the requirements of the building design process introduced above? The latitudes of formalizations in the A4–space are achieved in five ways. First, not all attributes have to carry values. Hence, design decisions are not necessarily represented as points in space, but instead as surfaces. As an example we consider the design of an object, that is first represented by the architect with the help of an ellipsis (see figure 2). The geometrical position of the object in the plan is known, but not the concrete specification of its type. Second, the interpretation of values of some attributes may depend on the values of other attributes, and hence may vary as the design evolves. The geometric representation on a lower scale of resolution, for example, is typically described by its bounding box giving only a rough outline of the design object. On a higher scale of resolution the geometry represents the exact contour of the object. Third, with every change in attribute values a new container is created so that the succession of design decisions forms a trajectory in space, allowing ease of monitoring and correcting of earlier decisions. If, e.g., in figure 2B a design decision is to interchange the living area and the working area, a new container will be created to reflect the new situation. The old containers are preserved but marked as old with the aid of a time stamp. This makes it possible to retrace earlier design decisions. We could view the container of a design process to lie on a trajectory in the design space. Fourth, containers are embedded into relationships with containers on the same trajectory or elsewhere in space by formulating consistency constraints. A great number of relationships of containers can be formulated, that are involved in the design in a very flexible and dynamic manner. With the container concept, most relationships can be expressed as overlaps within the design space. E.g., in figure 2 the container describing the whole building (A) overlaps with the containers describing the separate areas within the building (B). The relationships can be determined by computing the overlapping containers in this part of the design space.

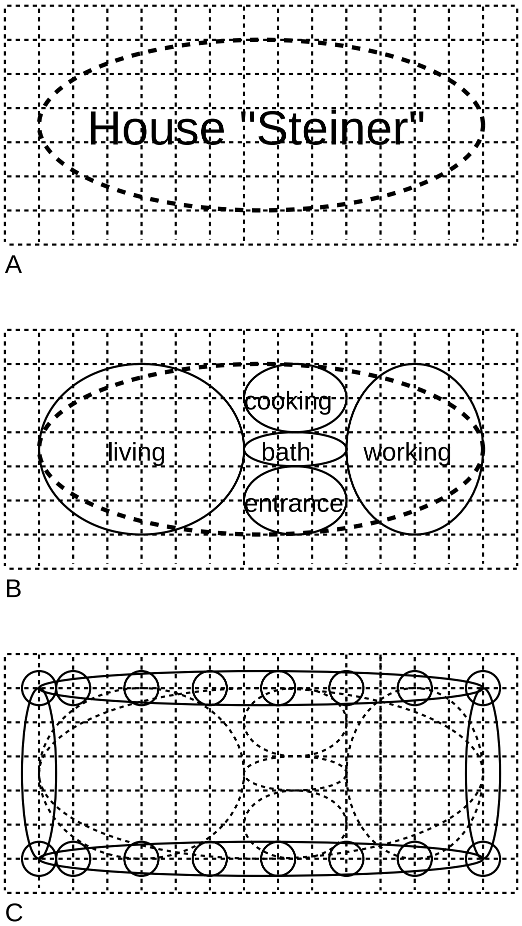

Figure 2: Modeling functional areas of a building in the A4-design space

The container model represents a common data kernel for all experts and is the basis for communication between them. In principle, each expert tool may work with any currently new container. Whenever the containers of two different experts collide, cooperation is needed in some way. At this point automatic mechanisms for conflict detection are needed, that, e.g., notify the two experts concerned.

The knowledge at the beginning of the design process is imprecise and vague, but becomes more and more concrete and detailed as the design evolves. Our observation is that at the beginning of the design the container model often is the only one in use. Later on, the special models of the design experts, become more elaborate and, hence, more important as the expert-specific information grows with the design process.

3 REALIZATION WITH A DATABASE SYSTEM

The data model presented above is implemented via an object–oriented database system. As usual we start by developing a conceptual model which subsequently is transformed into a logical object-oriented database schema. For the conceptual modeling step we used the object-oriented methodology OMT ([5]). The conceptual schema shown in Figure 3 follows the concept and has two main parts: The container schema and the special schemas of the experts. The container schema models the dimensions of the A4-space and additionally some descriptive attributes which are necessary for our control mechanism.

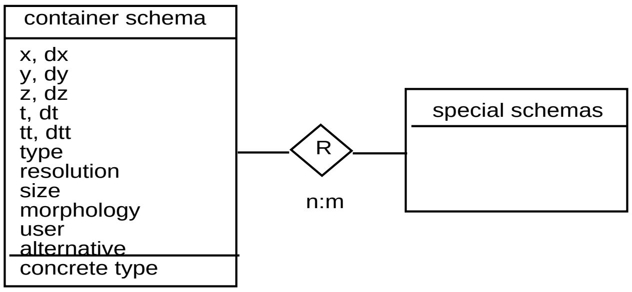

Figure 3: Conceptual schemas of containers and experts

The dimensions of the A4–space determine the attributes of the container type. There are two classes of attributes, one that refers to attributes with a continuous domain, and a second that contains attributes of discrete domain types. Continous attribute types are the geometrical dimensions, characterized by (x,dx), (y,dy), and (z,dz). X, y, and z describe the reference point of the bounding box, and dx, dy, and dz represent the vector of direction. There are two time dimensions: time and timetag, described with (t,dt) and (tt,dtt). Time gives the time interval during which the container is valid for that building, e.g. a window container is valid from the first input of the window object into the design until its disassembly. Timetag contains the time interval during which the container object resides without change within the design data. It allows to manage the history of design objects on a very fine level of granularity.

Attributes with a discrete domain type allow properties with a specific value. *Type* describes the class of design objects to which the container and the design decision behind it belongs. Example values are furniture or cold water pipes. The attribute *resolution* contains the level of detail to which the design object is represented. If the resolution is *by sketch* the bounding box represents only a rough outline of the design object. If we work in the phase of *detailed planning* the geometry represents the exact place of the design object. The attribute *size* reflects the scale of the actual planning, matching the usual scales in architectural design, e.g. 1:500, ..., 1:1. In a design environment supported by computers, these sizes reflect the information content which is shown within a design drawing. The attribute *user* refers to the expert who is responsible for the container object. *Morphology* describes the phenomenon that a planning may be described, in parallel or in sequential order, under different aspects. In our A4–space the attribute may have the values usage, supply, and development. The attribute *alternative* allows to denote different versions of a design object.

A special schema describes the special model of an expert. Consider a special tool MIDI ([1]) for the design of component–based steel buildings. The special schema is the representative of the data types used by the various MIDI restrictions. Figure 4 shows as an example a part of the MIDI schema. MIDI is a representative member typical for experts, that generate a large amount of (structural) knowledge which is ideally captured by an object–oriented special schema. Basically, our schema provides for a product model in the usual sense, with the novel aspect of separating special aspects (the special model) from common ones (the container model).

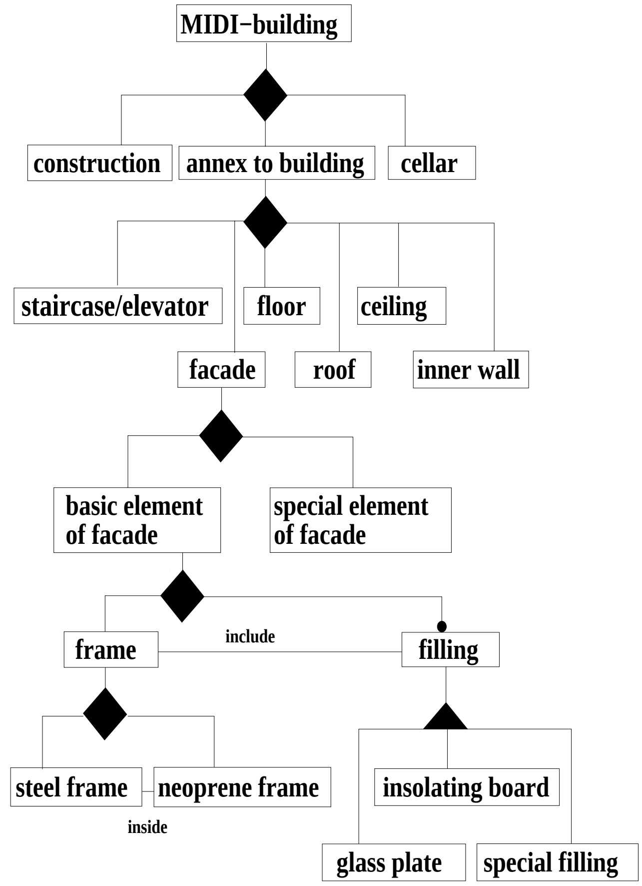

Figure 4: Conceptual schema of a part of MIDI

Figure 5 shows a bounding box integrated into a design. The design goal of this example is to build a building with the MIDI toolkit. The figure shows a first plan of the facade, which is represented by a rough bounding box. Figure 6 represents the container for the facade element.

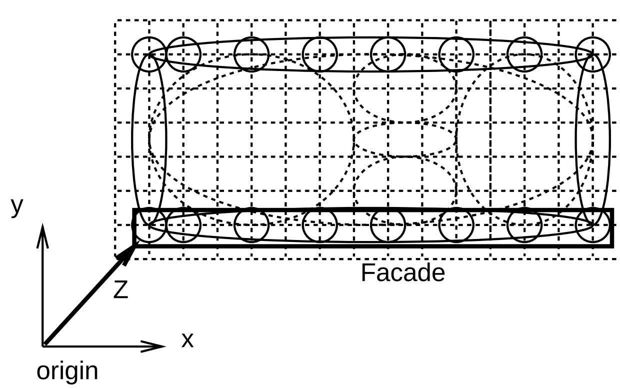

Figure 5: Bounding box representation of a facade element

The relationship between container model and the special model may be n–to–m, because one container will usually employ several special objects, and a special design object may also wish to communicate on several levels of granularity.

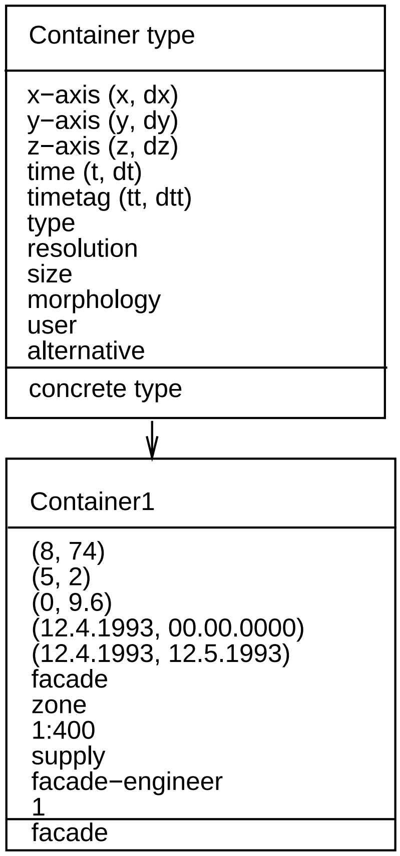

Figure 6: Container representation of a facade element

Therefore, a separate relationship object is used to relate a container model to a special model.

4 WORKING WITH THE MODEL

This section will give an overview on how to work with the modeling concept presented, on the one hand from the designer's point of view and on the other hand from the system's point of view. The designer should not be aware of the database system, but simply access the system by the usual interface familiar to him/her. Nontheless s/he will profit from the database characteristics, e.g. the inviolability, durability, and consistency of the data. The designer's view comprises the partial models corresponding to the expert tools. Figure 7 gives a coarse overview of the system architecture of the ArchE prototype. The figure illustrates the way in wich the data of the designers are transformed to the basic database structures. The architectural database interface divides the information povided by the designer at the user interface of the design system, e.g. from the MIDI expert planer, into two parts: information for the corresponding containers, and the objects belonging to the special model. Concurrently with the splitting of the data, the system creates the relationship–information between containers and special model instances, that is also held in the database, too. On reverse the data from the database, i.e. the containers and the instances of the special model, will be coalesced under the control of the architectural database interface. In this way the designer will always obtain a unique expert–specific (or partial model) view on his/her data.

Further on, for each creation and change of an object the system can check if a collision with other expert models takes place. Because all data needed for the check are stored as containers the system has to determine the containers in the database which overlap with the input container. This form of access to the containers is very typical and, hence, frequent in the environment, so that n–dimensional access paths (shown in the system architecture) are used for special access support.

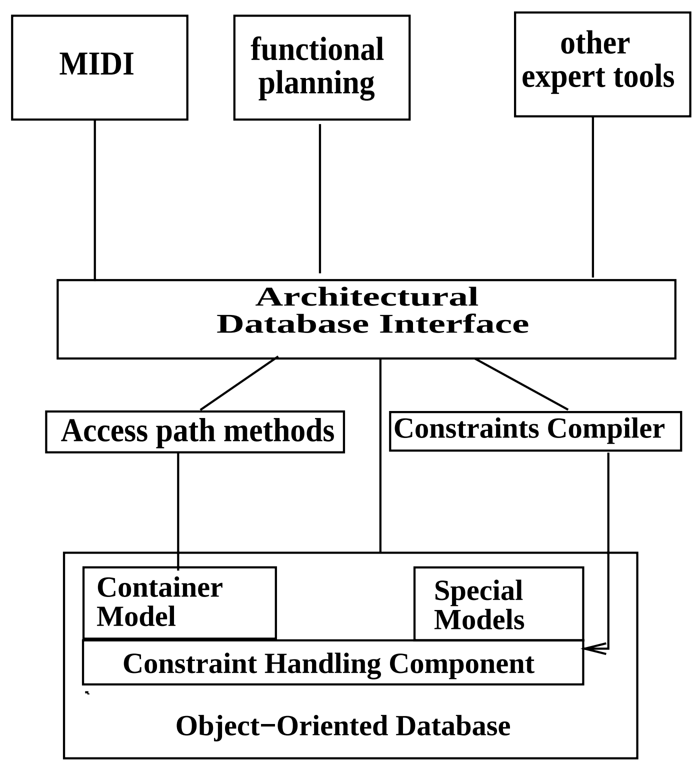

Figure 7: System architecture of the ArchE prototype

5 CONCLUSION

With the method completed and the object-oriented implementions in place we are presently testing our approach using two expert design tools, the component–based steel system MIDI and the installation planning tool Armilla. In particular, by integrating a constraint definition and handling component into the design database system, we will be in a position to evaluate our premise that consistency constraints are a particularly powerful mechanism for controlling the latitudes in design decisions needed during the more creative design phases.

References

[1] *midi 1000, Tragwerk Planungsgrundlagen.* USM bausysteme Haller.

[2] F. Haller, K. Friedrichs, V. Hovestadt, P. C. Lockemann, J. A. M"ulle, and R. Sturm. The design navigator. In *Proc. 5th Int. Conf. Computing in Civil and Building Engineering, Anaheim, California*, 1993.

[3] L. Hovestadt. *A4 - Digitales Bauen - Ein Modell f"ur die weitgehende Computerunterst"utzung von Entwurf, Konstruktion und Betrieb von Geb"auden.* Fortschrittsberichte VDI, Reihe 20 Rechnerunterst"utzte Verfahren Bd. 120, D"usseldorf, Dissertation, Institut f"ur Industrielle Bauproduktion, Universit"at Karlsruhe, 1994.

[4] L. Hovestadt, V. Hovestadt, J. A. M"ulle, and R. Sturm. ArchE – Entwicklung einer Datenbankunterst"utzten Architektur–Entwurfsumgebung. Ein Anforderungsbericht. Interner Bericht Nr. 23/94, Fakult"at f"ur Informatik, Universit"at Karlsruhe, Mai 1994.

[5] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson. *Object-Oriented Modeling and Design.* Prentice-Hall, 1991.