System Architecture for Computer-aided Building Engineering *

Katy Börner and **Dietmar Janetzko**
University of Freiburg, Centre for Cognitive Science, 79098 Freiburg, FRG
katy | dietmar@cognition.iig.uni-freiburg.de

Building engineering is one of the keystones to economic competitiveness. As a consequence, computational models for planning and design are important research topics. To meet the expectation that computer-aid enables to produce better designs in shorter time, the tools built on these models need to be fully integrated into the workflow of architects. The paper presents the architecture of a planning and design support system for building engineering. The architecture is highly interactive. It is adaptive to the tasks to be tackled, the particularities of the domain, and the preferences of users. The huge amount of knowledge necessary to support planning and design is acquired automatically, i.e., without bordering the user to answer thousands of questions. Figure 1 sketches the system architecture inclusive the human–system interaction. Modules are represented by boxes. Input and output data are denoted by arrows.

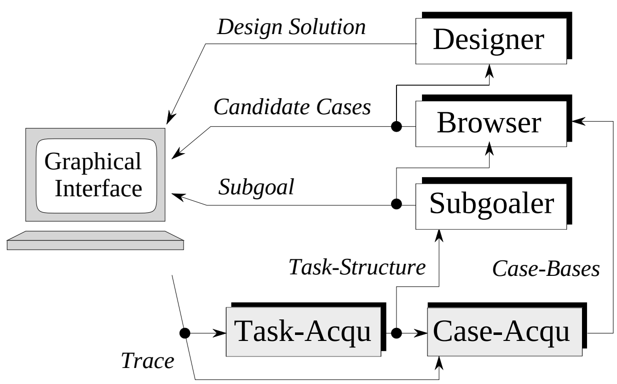

Figure 1: System architecture and human–system interaction

As graphical user interface we employ the hypermedia drawing environment DANCER developed in the Institut für Industrielle Bauproduktion (IFIB) at the University of Karlsruhe (Hovestadt 1993). While employing the system like a standard CAD drawing–tool the user creates new objects, resizes them, assigns labels to them, etc. Each of his actions is recorded in a *Trace* that provides the input to the support system. The system support, i.e., (1) guidance to select the next *Subgoal*, i.e., the next task to be tackled in design; (2) suggesting a set of *Candidate Cases* able to give some hint about how to solve the actual problem; and (3) generating an

*This research was supported by the German Ministry for Research and Technology (BMFT) within the joint project FABEL under contract no. 413-4001-01IW104. Project partners in FABEL are German National Research Center of Computer Science (GMD), Sankt Augustin, BSR Consulting GmbH, München, Technical University of Dresden, HTWK Leipzig, University of Freiburg, and University of Karlsruhe.

adapted *Design Solution* which fits the selected subgoal, is provided graphically, too. Each time the user is in a position to accept, modify, or reject suggestions. If the actual problem can't be solved by the system it is allocated to the user.

The basic knowledge representation used is A4 (Hovestadt 1993). Thus, every object occuring in architectural design may be represented by its geometrical attributes and its type attributes. The former represent placement and extension of objects. The latter refer to the object type (e.g., room, furniture etc.). CAD-like drawings correspond to snapshots of states in problem solving.

Knowledge acquisition for planning is done by ACT. The module extracts preferred state sequences, i.e., the *Task-Structure*, out of several user *Traces*. More exactly, it records predecessor relations between object types (e.g., room- outlets have to be designed before the furniture may be arranged etc.).

Knowledge acquisition for design is done by a module named CASE-CUTTER. During the design of a building architects frequently browse through old blueprints reflecting similar designs. According to our experience and to the literature as well (Goel 1989, Hinrichs 1992, Hua and Faltings 1993, Kolodner 1993) case-based reasoning seems to be the natural problem solving method. Aiming at a task-oriented user support (Janetzko, Börner, Jäschke and Strube 1994), the grainsize of cases corresponds to the grainsize of tasks. Thus the CASE-CUTTER uses the *Task-Structure* like a cookie-cutter to stencil cases out of user *Traces*. The cases are given in A4 representation, i.e., by attribute-value pairs. They are stored in task-dependent *Case-Bases*[1].

Subgoaling support is provided by the SUBGOALER that uses the *Task-Structure* to propose the next *Subgoal(s)* to be tackled.

Retrieval of candidate cases is possible by activating the BROWSER. Possible instantiations are retrieval modules like FAV, ASM, PIX (Voss, Bartsch-Spörl, Börner, Coulon, Dürschke, Gräther, Knauff, Linowski, Schaaf and Tammer 1994) which use appropriate distance or similarity measure to determine a set of *Candidate Cases* out of the *Case-Base*.

Adapted design solutions are generated by the DESIGNER module. The module extracts prototypes out of a set of *Candidate Cases*. The prototype itself acts as a holding form for typical object topologies. The modifications applied to the concrete cases may be used inversely to generate valid variants of the prototype (see also (Gero 1990)). This approach to structural similarity assessment and adaptation has been implemented in a module called SYN* (Börner 1994a, Börner 1994b).

The detailed description of the assumptions and approaches underlying the presented system architecture will be given in an extended version of the paper. Here, we aimed at an intuitive feeling of how much interactive and adaptive computer support is possible by 'hard' methods for 'soft' domains like building engineering.

[1]Different design steps, i.e. tasks, require different design strategies. For example, while return air accesses will be connected by using the shortest path, connections for fresh air accesses take curved tracks to achieve noise reduction. We assume, that objects of identical type are grouped together to constitute a state. The task to be supported by case-based reasoning is the creation of new objects based on objects already designed.

References

Börner, K. (1994a). Structural similarity as guidance in case-based design, *in* S. Wess, K.-D. Althoff and M. M. Richter (eds), *Topics in Case-Based Reasoning Selected Papers from the First European Workshop on Case-Based Reasoning (EWCBR-93)*, Vol. 837 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 197–208.

Börner, K. (1994b). Towards formalizations in case-based reasoning for synthesis, *AAAI-94 Workshop on Case-Based Reasoning*, pp. 177–181.

Gero, J. (1990). Design prototypes: a knowledge representation schema for design, *AI Magazine* **11**(4): 26–36.

Goel, A. K. (1989). *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving*, PhD thesis, Ohio State University.

Hinrichs, T. R. (1992). *Problem solving in open worlds: A case study in design*, Lawrence Erlbaum Associates.

Hovestadt, L. (1993). A4 – digital building – extensive computer support for the design, construction, and management of buildings, *CAAD Futures '93, Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures*, North-Holland, pp. 405–422.

Hua, K. and Faltings, B. (1993). Exploring case-based building design – CADRE, *AI EDAM* **7(2)**: 135–143.

Janetzko, D., Börner, K., Jäschke, O. and Strube, G. (1994). Task-oriented knowledge acquisition and reasoning for design support systems, *First European Conference on Cognitive Science in Industry*, pp. 153–184.

Kolodner, J. L. (1993). *Case-based reasoning*, Morgan Kaufmann.

Voss, A., Bartsch-Spörl, B., Börner, K., Coulon, C. H., Dürschke, H., Gräther, W., Knauff, M., Linowski, B., Schaaf, J. and Tammer, E. C. (1994). Retrieval of similar layouts – about a very hybrid approach in FABEL, *in* J. Gero and F. Sudweeks (eds), *AI in Design'94*, Kluwer Academic Publishers, Dordrecht, pp. 625–640.