Case-Based Design in the SEED System

Ulrich Flemming
Carnegie Mellon/Building Industry
Computer-Aided Design Consortium (CBCC)
Carnegie Mellon University
Pittsburgh, PA 15213
ujf@globe.edrc.cmu.edu

SEED is a software environment to support the early phases in building design currently under development. The overall architecture of the system reflects an open-ended division of the early design stage into specific *tasks* or *phases*, each of which is characterized by the *type of problem* it addresses and the *type of solution* it generates, and is supported by SEED through a specific *module*. The three modules planned for the first system prototype deal, respectively, with architectural programming, schematic layout design, and schematic 3-dimensional configuration design. The form generation operators to be used in these modules were developed in connection with two multi-generational research efforts: the LOOS and the ABLOOS systems [Flemming et al., 1988; Coyne and Flemming, 1990] and the GENESIS system [Heisserman, 1991]. Among the capabilities offered by SEED is the automatic storage of solutions generated with the system as *cases* and their retrieval for reuse in a similar problem situation. SEED responds to practical demands requiring an approach to case-based design that differs from those known from the literature [e.g., Domeshek and Kolodner, 1992; Hua et al., 1992].

The reuse capability of SEED aims at aiding designers in two crucial ways: (i) access to a large memory of past solutions that is likely to contain instances designers will *not* recall on their own; and (ii) the fast derivation of an initial solution that is immediately available for editing and modifications under either the system's or the designer's control. Past solutions may be reused in any module, and the problem addressed in a module may be decomposed into a hierarchy of (sub)problems. Case retrieval is available in SEED across hierarchical problem decomposition levels within a module and across modules. The case memory of SEED accumulates as a side-effect of a firm's normal design activities without additional efforts. If a designer requests a solution to be stored as a case, an index is computed based on the current problem statement and stored with the associated solution to support retrieval.

References

[Coyne and Flemming, 1990] R. F. Coyne and U. Flemming, "Planning in Design Synthesis: Abstraction-Based LOOS," in *Artificial Intelligence in Engineering V, Vol. 1: Design (Proc. Fifth International Conference, Boston, MA)*, J. Gero, editor, Springer, New York, 1990, 91-111

[Domeshek and Kolodner, 1992] E. A. Domeshek and J. L. Kolodner, "A Case-Based Design Aid for Architecture", in *Artificial Intelligence in Design '92*, J. Gero, editor, Kluwer Academic Publishers, Boston, MA, 1992, 497-516

[Flemming et al., 1988] U. Flemming, R. Coyne, T. Glavin, and M. Rychener, "A Generative Expert System for the Design of Building Layouts - Version 2," in *Artificial Intelligence in Engineering: Design (Proc. Third International Conference, Palo Alto, CA)*, J. Gero, editor, Elsevier, New York, 1988, 445-464

[Heisserman, 1991] J. Heisserman, *Generative Geometric Design and Boundary Solid Grammars*. PhD dissertation, Dept. of Architecture, Carnegie Mellon University, Pittsburgh, PA, 1991

[Hua et al., 1992] K. Hua, I. Smith, B. Faltings, S. Shih, and G. Schmitt, "Adaptation of Spatial Design Cases", in *Artificial Intelligence in Design '92*, J. Gero, editor, Kluwer Academic Publishers, Boston, MA, 1992, 559-575