Using Multiple Sources of Knowledge to Generate Design Decompositions

David C. Brown & Jingwen Liu

Computer Science Department
Worcester Polytechnic Institute, Worcester, MA 01609, USA.
(508) 831-5618
dcb@cs.wpi.edu

Introduction

Design expert systems can be characterized by the type of compiled knowledge used. Compiled knowledge is efficient for design problem-solving with a bounded set of design problems, but often leads to failure in slightly different design problem situations. Design decomposition knowledge is often compiled into design systems. Design specification changes often impose problem decomposition changes. Generating decomposition knowledge for a slightly different design problem is difficult.

Our research is about the compilation of design decomposition knowledge for design expert systems (Liu & Brown 1992). Our goal is to decompose design problems using a variety of types of knowledge. We generate design decomposition knowledge using contributions from object knowledge, functional knowledge, design cases, design heuristics, general problem-solving knowledge and domain knowledge.

Design object knowledge (i.e., topology and geometry) can provide descriptions about components, attributes and their relationships. Combining them with requirements produces evidence for decomposition. Functional knowledge can also provide evidence. Design cases help by providing the decomposition of all or part of a similar design problem.

Knowledge Compilation is a type of learning in which existing knowledge is converted into new forms, with the intent to improve problem-solving efficiency (Brown 1990). The compilation of decomposition knowledge is the first step toward the compilation of DSPL design plans (Brown & Chandrasekaran 1989), providing problem decomposition and subproblem ordering knowledge.

We want decomposition mechanisms that use as much existing knowledge as possible, that aggregate the evidence for different decompositions, and which use techniques that we can imagine a human designer using.

The Approach

We use a tree structure to represent the decomposition. Each node of the tree contains a problem description, a list of competing decomposition hypotheses, and links to subproblems, knowledge sources, etcetera.

Knowledge extraction and decomposition factor generation: Given a problem description, the various types of relevant knowledge are deduced and retrieved. Reasoning methods are applied to the different types of knowledge to generate a set of decomposition factors. A *decomposition factor* (df) is a suggestion about how to partition a design object, a component, or a set of attributes. All the dfs generated are organized into an hierarchy, classified by the entities to which the factors are applied (e.g., attributes, or components).

Hypothesis proposal and refinement: Decomposition factors reflect different aspects and levels of interactions among the components. Many decomposition factors will be generated. We combine mutually compatible decomposition factors into a *df-group*. Based on the df-groups generated, a heuristic "composer" is used to produce a set of problem decomposition *hypotheses*. These are ranked by the strength of their supporting decomposition factors and by other heuristics.

Hypothesis confirmation: Rough estimates are made of the decomposability of each of the subproblems for the highest ranked hypothesis. The hypothesis is accepted only when the estimate is below some limit. Otherwise, we modify the hypothesis, or try an alternative. The whole decomposition process will repeat for each subproblem until primary design problems are reached, or until no decomposition exists. A *primary design problem* is a routine problem with the solution directly available.

Summary

We have outlined a mechanism for compiling decomposition knowledge from a variety of types of knowledge. Currently we are implementing an experimental prototype system. We expect to obtain detailed results in late 1992.

References

Brown, D. C. 1990. Compilation: The Hidden Dimension of Design Systems. *Intelligent CAD, III.* (Eds.) H. Yoshikawa & F. Arbab, North-Holland.

Brown, D. C., & Chandrasekaran, B. 1989. *Design Problem Solving: Knowledge Structures and Control Strategies.* Morgan Kaufmann.

Liu, J., & Brown, D. C. 1992. The Generation of Decomposition Knowledge for Near Routine Design Problems. *Appl. of AI in Eng. VII.* (Eds.) D. E. Grierson, G. Rzevski & R. A. Adey, Elsevier Applied Science, pp. 825-833.