Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte
Band: 68 (1993)

Artikel: A memory organization for heterogeneous design knowledge
Autor: Wang, Weiyuan / Gero, John
DOI: https://doi.org/10.5169/seals-51846

A Memory Organization for Heterogeneous Design Knowledge

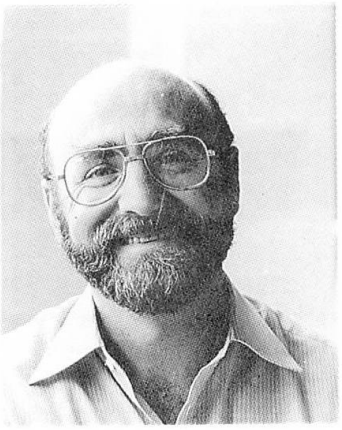
Organisation des données pour des connaissances hétérogènes en matière de projet

Datenverwaltung für heterogenes Entwurfswissen

Weiyuan WANG
Doctoral Student
University of Sydney
Sydney, Australia

John GERO
Prof. of Design Science
University of Sydney
Sydney, Australia

Weiyuan Wang received his BS degree in Computer Eng. from the Chongqing Univ., China, in 1983. Currently a doctoral student, his research interests include machine learning in design, knowledge acquisition and representation.



John Gero holds degrees in engineering, building science and architecture. He has been a visiting professor at Universities in the USA, UK and France. His current research is in artifical intelligence in design.

SUMMARY

Design prototypes and design cases are two disparate approaches to the representation of design knowledge. Design prototypes are a schema for representing compiled knowledge while design cases are a schema for representing situated knowledge. This research attempts to integrate both knowledge in a single system. A generalization-based memory organization for heterogeneous design knowledge is presented in this paper. Machine learning techniques are used to generalize knowledge-rich design schemas which function as both a knowledge representation schema for storing generic design cases and a memory organization facility for indexing design cases and design prototypes.

RÉSUMÉ

Les prototypes de projets et les études de cas spécifiques sont deux approches incompatibles pour représenter les connaissances en matière de projets: la première transmet un savoir par assemblage, tandis que la seconde procure un savoir en fonction de la situation. L'objectif de la recherche consiste à intégrer les deux domaines de connaissances en un système unique. Les auteurs présentent une organisation de données basée sur la généralisation pour les connaissances hétérogènes en matière de projets. Des moyens didactiques mécaniques servent à condenser des schémas de projets riches en informations, qui satisfont aussi bien à une fonction de stockage des cas d'études considérés comme spécifiques, qu'à une fonction d'organisation de données.

ZUSAMMENFASSUNG

Typenentwürfe und Entwurfsstudien sind zwei inkompatible Arten, Entwurfswissen darzustellen: Erstere vermitteln zusammengetragenes Wissen, während letztere situationsbedingte Erkenntnisse wiedergeben. Die Forschung versucht beide Teilbereiche in ein System zu integrieren. Als Beitrag dazu wird eine auf Verallgemeinerung bauende Datenorganisation für heterogenes Entwurfswissen vorgestellt. Unter Nutzung maschineller Lernfähigkeit werden informationsreiche Entwurfsschemata kondensiert, die sowohl eine Funktion der Ablage erdachter Entwurfsfälle als auch der Datenorganisation erfüllen, indem auf Entwurfsstudien und Typenentwürfe verwiesen wird.

1. INTRODUCTION

Design is complex and a designer usually utilizes several types of knowledge in solving a design problem. The design knowledge can be general or specific, formal logic or heuristic rules, generalized from experience or situated in design cases, etc. To integrate different types of knowledge to create more powerful systems is an important issue in the current research on knowledge-based systems. Although the integration of different types of knowledge or knowledge representations has the same importance as the integration of reasoning paradigms in the development of such systems, most research has focused on the integration of different reasoning paradigms while there has been little on the organization of heterogeneous design knowledge. This paper presents one approach to the memory organization for heterogeneous design knowledge and addresses the relevant issues.

Two types of widely-discussed design knowledge are design prototypes and design cases. Design prototypes are a schema for representing compiled design knowledge while design cases are a schema for representing situated design knowledge. Design prototypes are efficient in generating and evaluating design alternatives because of their richness of knowledge while design cases are powerful in providing a good set of initial values for the instantiation of a design prototype [12]. Case-based design needs to be supported by domain knowledge such as qualitative models [3], decomposition models [8] and design prototypes [11]. On the other hand, prototype-based design can be enhanced by the design knowledge situated in design cases [10]. This research attempts to integrate both kinds of knowledge in a single system in order to complement the strengths of each. To connect heterogeneous knowledge and link multiple representations, we must have an efficient way to organize the memory. Since the knowledge is acquired with different methods and from different sources, we must also have a mechanism to maintain the consistency of memory. We have proposed an integrated learning model [13], which has been further developed into a generalization-based memory organization for heterogeneous design knowledge. A knowledge-rich design schema has been introduced which functions as both a knowledge representation schema for storing generic design cases and learned knowledge, and a memory organization facility for indexing design cases and linking them to related design prototypes. Machine learning techniques have been applied to generalize the schemas and organize them in a hierarchy. The knowledge in the schemas is in the form of attribute associations, default values, value ranges of attributes, qualitative and quantitative relations which have been suggested being useful for guiding the design refinement in prototype-based design systems [12] and the design adaptation in case-based design systems [3]. The indexing mechanism of the system provides a user with flexibility in both design case retrieval and design prototype retrieval. The overall relationships between generalized design schemas, design cases and predefined design prototypes are shown in Figure 1.

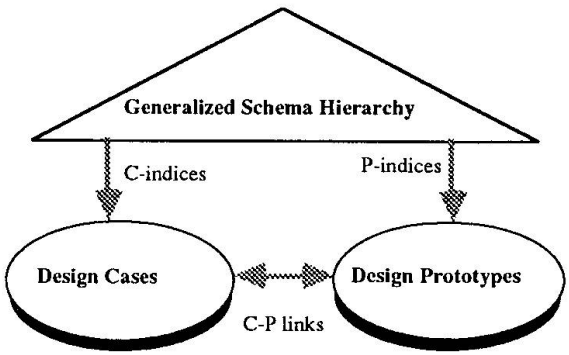


Figure 1. The relationships between generalized design schemas, design cases and predefined design prototypes. C-indices means case indices, P-indices means prototype indices, and C-P links means case-prototype links.

2. HETEROGENEOUS DESIGN KNOWLEDGE

In this section we discuss three types of design knowledge, predefined design prototypes, design cases, and computer-generalized design schemas.

2.1 Design prototypes

A design prototype is a conceptual schema for representing compiled design knowledge. The formal model of design prototype [2] is symbolically represented as follows:

$$P = (F, B, S, C, K)$$

where

P	a design prototype
F	a set of functional attributes
B	a set of behavioural attributes
S	a set of structural attributes
C	a set of contextual attributes
K	design knowledge

One representation of design prototype is frame-like. Each attribute may have type, unit, default value, and value range. For example, for attribute, horizontal-module of office buildings, the type is numeric, the unit is feet, the value range is 2 - 4, and the default value is 3.4. The types of attributes can be nominal, ordered, structured, numeric, set, and other design prototypes. Design knowledge can be if-then rules, qualitative and quantitative relations, and dependencies between variables.

A series of design prototypes, from specific to general, forms a design prototype memory. In a prototype-based design system, designing starts with function requirements from a client followed by the process of continually finding appropriate design prototypes using existing information as the index, deriving instances from them and synthesizing these instances to compose the desired artifact. The speed of prototype retrieval and the degree of relevance between new design problem and the retrieved design prototype have strong influences on the system efficiency.

2.2 Design Cases

A design case is represented as a set of features or attribute-value pairs which are classified into four categories of function, behaviour, structure and context. It is formally defined as follows:

$$\text{Design case} = (F, B, S, C)$$

where

F	a set of functional features
B	a set of behavioural features
S	a set of structural features
C	a set of contextual features

Like design prototypes, the type of an attribute here may be nominal, ordered, structured, numeric, set, and a particular design prototype.

2.3 Design schemas

The memory system will automatically generalize knowledge-rich design schemas from design cases. We call them knowledge-rich design schemas because they contain more information than usual generic design cases. A generalized design schema consists of common features, generalized features, generalized knowledge (qualitative and quantitative relations), and typology (indices to design cases, more specific schemas, more general schema, and related design prototypes) (Figure 2).

Generalized Design Schema:
- link to more general design schema:
- common features:
- generalized features:
- generalized knowledge:
  - qualitative relations:
  - quantitative relations:
- index or discrimination nets:
  - link to design cases
  - link to more specific design schemas:
  - link to related design prototypes

Figure 2. The representation of a generalized design schema

Common features are those shared by all cases under the schema and each of generalized features has a value range and often a default value. By quantitative relations here we mean empirical formulae among numeric attributes (e.g., for a given set of examples of office buildings, the system has learned that maintenance-cost-per-year is approximately equal to $0.1 \times$ total-cost-of-construction). Qualitative relations among numeric attributes are represented as empirical, proportional relations (e.g., for a given set of examples, number-of-building-stories$^+ \rightarrow$ cost-per-efficient-area$^+$; building-performance-level$^+ \rightarrow$ ratio-of-net-area-to-gross-area$^-$; the last relation says that the rise of building performance level may lead to the reduction of ratio of net area to gross area). These relations can be used as heuristics to guide the design adaptation in case-based reasoning and the instance refinement in prototype-based design, or aid a designer in making decision at preliminary design stage.

3. THE MEMORY ORGANIZATION

3.1 The General Strategy

In general, machine learning techniques are used to create knowledge-rich design schemas which function as both a knowledge representation schema for storing generic design cases and design knowledge learned from design cases, and a memory organization facility for linking design cases to related design prototypes as shown above in Figure 1.

The design schemas are further classified into F-schema, B-schema, S-schema and C-schema. F-schema is a schema indexed with function descriptions and its creation is dependent on a cluster of functional features; similarly, B-schema is a schema indexed with behaviour descriptions, S-schema is a schema indexed with structure descriptions, and C-schema is indexed with the description of design context. Each of classes of schemas forms separate hierarchies, and in a whole they construct an efficient memory that stores and retrieves design cases, generalizes more general knowledge from design cases, and associates relevant design cases, design schemas and design prototypes together (Figure 3).

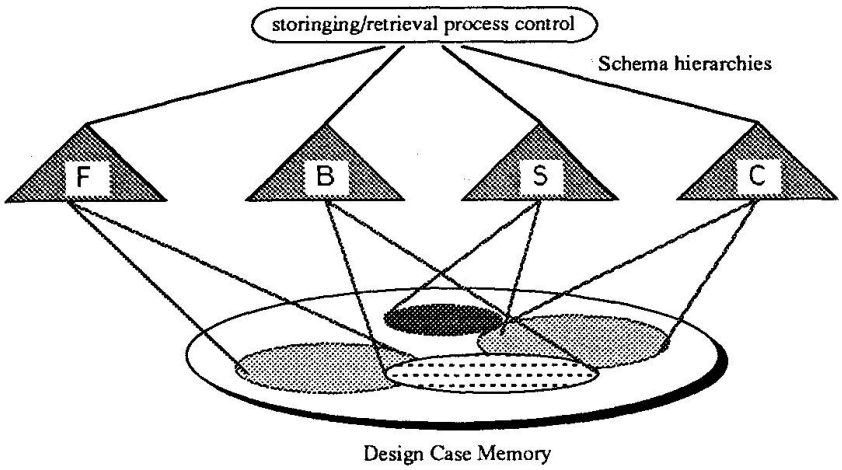


Figure 3. Design cases can be retrieved with each of four schema hierarchies of function, F, behaviour, B, structure, S, context, C, or any combination of them.

3.2 Main Processes

Two main processes of memory organization suggested in Wang and Gero [13] are conceptual schema hierarchy formation and numeric relation discovery, Figure 4. The techniques used in the former are influenced by the generalization-based memory techniques applied in the UNIMEM [5]; the techniques used in the latter are similar to those presented in ABACUS[1] and BACON [4]. The rest of this section will describe these processes.

Input —> Conceptual Schema Hierarchy Formation —> Numeric Relation Discovery —> Output
(Design Cases) (Schema Hierarchy)

Figure 4. Two main processes of the memory organization: conceptual schema hierarchy formation and numeric relation discovery.

Process of conceptual schema hierarchy formation

1. Receive new design case;
2. Let Feature-list = Functional attributes of new design case;
3. Let most-specific-schemas = results of SEARCH (root-of-F-hierarchy, Feature -list, design-case);
4. For each of nodes in the paths of search, call QUALITATIVE-DISCOVERY process;
5. For each of Most-specific-schemas, do

   5a. If the similarity between new design case and one indexed under the schema is higher than a predefined threshold,

   create a new schema,
   remove the case index from the original schema,
   create index from original schema to new schema,
   create index from new schema to the design cases,
   generalize default values and value ranges of attributes;

   If any feature in the schema has a prototype name as its value,
   create the index from the schema to that prototype,
   call a process CONSISTENCY-MAINTAIN(Design-case, prototype-name)
   to update the related prototype;

5b. If no generalization can be done,
add new case into the schema,
update the index to design cases
update the default values and value ranges of attributes.

The system incrementally incorporates design cases into a hierarchy by this process. Various indices will be created. The SEARCH process is similar to that in UNIMEM except that only the indexing of a schema and Feature-list will be considered in matching. The process QUALITATIVE-DISCOVERY will create or evaluate qualitative relations and may call a quantitative discovery process. Both processes are described below. The process CONSISTENCY-MAINTAIN will check related design prototype and may modify default value or value range of an attribute. B-schema hierarchy, S-schema hierarchy and C-schema hierarchy are created and updated with the same process with minor changes as in steps 2 and 3. Part of a generalized schema hierarchy and its relations to design case and design prototypes is graphically described in Figure 5.

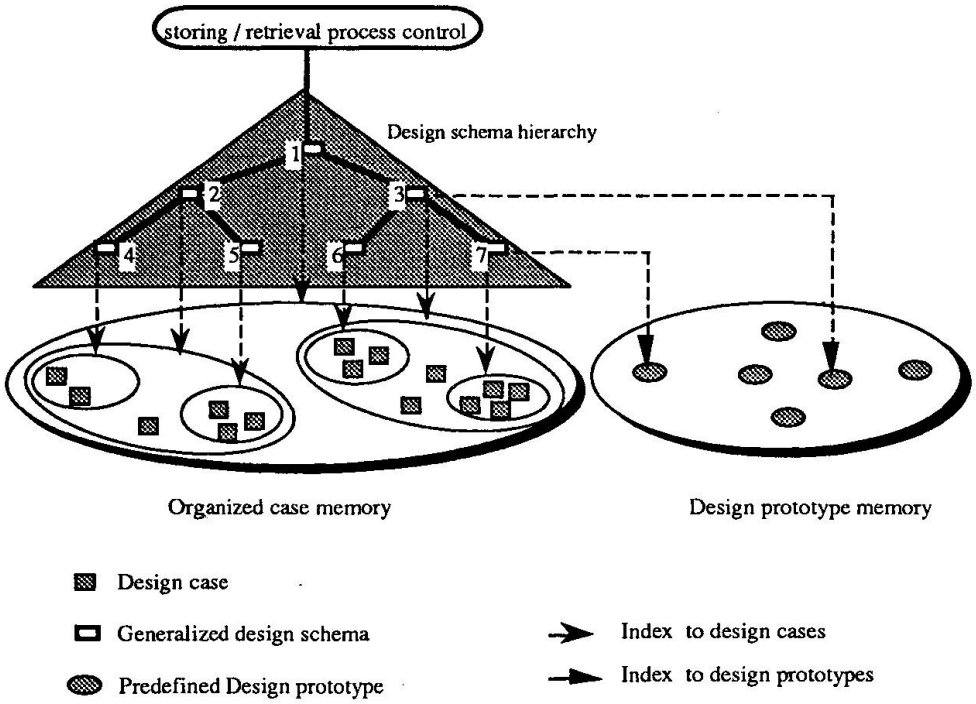


Figure 5. Part of detailed representation of generalized schema hierarchy and its relations to design cases and design prototypes.

The numeric relation discovery itself consists of two processes: qualitative relation discovery QUALITATIVE-DISCOVERY and quantitative relation discovery QUANTITATIVE-DISCOVERY. Quantitative relation between two attributes are discovered by the following process:

Process of qualitative relation discovery between X and Y

1. As design cases are sequentially incorporated into the memory,
check the directions of their value changes in last two cases;
if both go in the same direction (increase or decrease),
let counter P increase by 1;
if one variable increases and another decreases,

let counter N increase by 1,

2. If P / (P + N) > predefined threshold T1,
   hypothesize a relation $X^+ \rightarrow Y^+$;
3. If N / (P + N) > threshold T1,
   hypothesize a relation $X^+ \rightarrow Y^-$;
4. If a hypothesis has already created,
   both P / (P + N) and N / (P + N) < predefined threshold T2,
   propagate the hypothesis into sub-schemas except one that current case belongs to,
   remove the hypothesis from the original schema;
5. If P + N > predefined threshold T3,
   call a process QUANTITATIVE-DISCOVERY.

The value of P/(P+N) or N/(P+N) is regarded as the confidence of hypothesis. When it becomes smaller than the predefined threshold T2, the hypothetical relation is disproved and deleted; on the other hand, when the number of cases (P+N) becomes larger than predefined threshold T3, quantitative discovery process QUANTITATIVE-DISCOVERY will be called, which is described as follows.

Process of qualitative relation discovery between X and Y

1. If $X^+ \rightarrow Y^+$,
   create new variable X/Y, let C equal to the value of X/Y;
   if C remains constant or its deviation is smaller than predefined threshold T4,
   hypothesize an equation X/Y = C;
2. If $X^+ \rightarrow Y^-$,
   create new variable X×Y, let C equal to the value of X×Y;
   if C remains constant or its deviation is smaller than predefined threshold T4,
   hypothesize an equation X×Y = C;
3. If a hypothesis has already been created,
   update the deviation of C;
   if the deviation of C becomes larger than the predefined threshold T4,
   delete the hypothesis.

As described in the above processes, for each of schemas which contains more than two design cases, the system starts searching for qualitative relations, while the search for quantitative relation will be carried out only if a qualitative relation has been found and the number of supporting cases has become equal to or larger than a pre-defined threshold. Since the generalized relations are assumed as a kind of heuristics, not as scientific laws, they may be approximate but still useful. The computational cost of relation discovery is high. To control the combinatorial explosion, the system may only search the potential relations between those variables which have dependency relations with each other. Design prototypes provide such information. The underlying assumption is that it is relatively easy for experts to tell if there exists a dependency between two variables but it is much more difficult for them to describe mathematically the relations between them.

As we have seen from the above processes, once a relation is hypothesized, it can be incrementally evaluated as new design cases come. If a quantitative relation at a high-level schema does not cover a new case, the system will not simply throw it away, but propagate it into sub-nodes which the new case does not belong to (in effect it just reduces the application scope) and search for a qualitative relation at original node.

Process of memory consistency maintenance

When a schema is created or updated, if there is an index from a schema to a design prototype, the related design prototype(s) will be checked by the process of CONSISTENCY-MAINTAIN. The value range

of a variable in design prototypes will be expanded if it does not cover the range of the variable in the generalized schema. This gives the memory the capability of self-adaptation.

4. RELATED WORK

Using machine learning techniques to organize case memory and acquire knowledge from design cases was investigated by Reich et al. [9]. In their system, Bridger, only two types of knowledge were considered, design cases and generalized design schemas. Neither qualitative nor quantitative relations are generalized in the schemas. However, it is possible to insert qualitative and quantitative discovery processes in the process of concept formation and generalize more knowledge in the schemas. Design prototypes may also be integrated into the system. The combined application of concept learning and quantitative discovery in design was also examined by Maher and Li [7, 6]. Although a similar two-stage strategy was applied in their system, they utilized a probabilistic association network and classical numerical methods, regression analysis and dimensional analysis, and approached concept learning and quantitative discovery in a non-incremental fashion. Their system, unlike our model presented above, only focused on automatic design knowledge acquisition from design cases and neither indexing nor pre-defined design prototypes needed to be considered. Case memory organization and combined application of design cases and qualitative models was explored by Goel et al. [3]. In their system, Archie, design cases are indexed by a different categories of features, such as design specifications, design solutions, design outcomes, design critiques, potential problems, and adaptation strategies.

5. CONCLUSION

We have presented a machine learning approach to the memory organization for heterogeneous design knowledge in form of design cases, design prototypes and design schemas. The integrated application of concept formation and qualitative and quantitative discovery techniques in design is original. To combine design cases and design prototypes for more powerful knowledge-based design systems, their memory organization needs to be further explored. The efficiency of qualitative and quantitative relation discovery in an incremental fashion needs to be empirically analysed. What should be in a design case and what should be in a design prototype need to be further justified in the area of their integrated applications.

REFERENCES

1. Falkenhainer, B. and Michalski, R. S. (1986). Integrating quantitative and qualitative discovery: the ABACUS system, *Machine Learning* **1**: 367-402.
2. Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design, *AI Magazine* **11**(4): 27-36.
3. Goel, A. K., Kolodner, J. L., Pearce, M., Billington, R. and Zimring, C. (1991). Towards a case-based tool for aiding conceptual design problem solving, *Proceedings Case-Based Reasoning Workshop*, Washington, DC.
4. Langley, P., Simon, H. A., Bradshaw, G. L. and Zytkow, J. M. (1987). *Scientific Discovery*, MIT Press, Cambridge.
5. Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM, *Machine Learning*, 2: 103-138.
6. Maher, M.L. (1992). Automated knowledge acquisition of preliminary design concepts, *ASCE 8th Conference on Computing in Civil Engineering*, Dallas, Texas, pp. 975-982.
7. Maher, M. L. and Li, H. (1992). Automatically learning preliminary design knowledge from design examples, *Microcomputers in Civil Engineering* 7: 73-80.

8. Maher, M. L. and Zhang, D.M. (1991). CADSYN: using case and decomposition knowledge for design synthesis, *in* J. S. Gero (ed.) *Artificial Intelligence in Design*, Butterworth-Heinemann, Oxford, pp. 137-150.
9. Reich, Y. and Fenves S. J. (1991). The formation and use of abstract concepts in design, *in* D. H. Fisher, M. J. Passani and P. Langley (eds), *Computational Approaches to Concept Formation*, Morgan Kaufmann, San Mateo, CA.
10. Rosenman, M. A., Gero, J. S. and Oxman, R. E. (1991). What's in a case: the use of case bases, knowledge bases and databases in design, *in* G. N. Schmitt (ed.), *CAAD Futures'91*, ETH, Zurich, pp.263-277.
11. Sun, D. and Gero, J. S. (1991). Representing design experience through design cases: the use of case-based reasoning in design, *in* G. Woodbury (ed.), *The Technology of Design*, ANZAScA, University of Adelaide, Adelaide, pp.235-244.
12. Tham, K. W. (1991). A model of routine design using design prototypes, *PhD Thesis*, Department of Architectural and Design Science, University of Sydney, Sydney.
13. Wang, W. and Gero, S. J. (1991). A model for the organisation of design prototype and design case memories, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design, Preprints of the Workshop of IJCAI-1991, Sydney*, University of Sydney, Australia, pp.301-307.