Design Case Retrieval by Generic Representations

Citation for published version (APA):

Achten, H. H. (2000). Design Case Retrieval by Generic Representations. In J. S. Gero (Ed.), *Artificial Intelligence in Design '00* (pp. 373-392). Kluwer Academic Publishers.

Document status and date:

Published: 01/01/2000

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

DESIGN CASE RETRIEVAL BY GENERIC REPRESENTATIONS

HENRI H. ACHTEN

Eindhoven University of Technology

Faculty of Architecture, Building and Planning

Design Systems

The Netherlands

Abstract. Case-Based Reasoning and Case-Based Design have been proposed to utilize knowledge of previous design solutions to understand or solve current design problems. Case retrieval is often performed on the basis of verbal indexing systems, whereas in design the use of graphic representations is predominant. Case retrieval therefore, requires verbalization from the designer, which can interrupt the flow of thought during design. This paper aims to demonstrate how the theoretical work on generic representations in architectural design can form a graphic indexing scheme for cases that are graphic representations. The paper outlines three techniques: a decision-tree for graphic unit recognition, a query composition algorithm, and a look-up table for case retrieval. These three techniques are worked out in the context of a Case-Based Design Aid System.

1. Introduction

Routine design as a category is reasonably well covered both in research and existing CAAD applications. Innovative and creative design however, still face significant problems in design support. One approach of tackling design problems is by means of Case-Based Design (CBD). In CBD a previous design episode or case that resembles the current design problem is retrieved from a case base and (slightly) adapted to fit the problem. In this way, CBD resembles the cognitive approach of designers and problem solvers who use previous design solutions when generating new solutions. This is in fact the

basic cognitive model underlying both Case Based Reasoning (CBR) and CBD.

A main assumption both in case retrieval and case adaptation is that previous cases that resemble the current problem also bear the basic solution principles. The relatively complete solution then needs adaptation to fit the current problem. Issues in CBD therefore, are: problem definition, assembling an indexed description, matching the indexed description to the case base, retrieve matching cases, case adaptation, evaluation of the adapted case in the design problem, and storing the new case in the case base.

CBD is not as knowledge-intensive as conventional techniques in Artificial Intelligence such as rule-based systems, production systems, and expert systems that require more elaborated encoded knowledge structures and procedures. It relies on the structure present in the whole solutions of the case base and presupposes that the relations that made up the good solution in the past can be kept intact for the new design problem. The paradigm is well established and has seen much research in the past years under varying assumptions. Examples are: Kolodner (1993), Lee and Lee (1995), Oxman (1995), Tan et al. (1995), Maher and Pu (1997), Chiu and Chi (1997), de Groot et al. (1999), Mallory-Hill (1998), Heylighen (1998), Grassi et al. (1999). It has yielded a number of applications such as: CADRE, SEED, FABEL, ARCHIE and ARCHIE-2, IDIOM, PRECEDENT, EDAT, and DYNAMO.

Cases are retrieved through an index system. The indexing determines the features via which a case can be found. Index systems typically are text based. This means that finding the case requires verbalizing the design problem, which stands in contrast with the graphic way of working through sketching. Lacking an explicit vocabulary for describing drawings, it is hard to verbalize thoughts that are formed through sketches. Furthermore, reasoning in design, in particular in the formative early stages, is often done on the basis of sketches (Verstijnen 1997, McFadzean 1999). Architects often associate during design via the images at hand through recollection of images of other projects.

Given the abundance of graphic material that makes up the knowledge base of an architect, it makes sense to propose that graphic representations can function as a trigger for case indexing and retrieval. This means that the following questions need to be addressed: how can a graphic representation be interpreted; how can such an interpretation be translated into relevant questions for the designer; how can it serve as input for a query that retrieves relevant cases; do cases need to be presented as graphic representations only; how will information be presented?

The work on graphic units and generic representations that has been done at Eindhoven University of Technology provides a way in understanding

graphic representations in architectural design. It offers a methodology for elucidating the design decisions that a designer deals with when working on a particular graphic representation. This is done through identifying pervasive elements (graphic units) in the drawing that have general accepted meaning. By means of graphic units, graphic representations have been classified into generic representations. Sets of generic representations therefore, constitute an accessible case base of graphic representations.

In this paper, we demonstrate how graphic units can be used to index a given graphic representation, and how this can lead to case retrieval which shows the architect examples of other designers' drawings that deal with the same design problems. In this way, it is possible to establish a Case-Based Design Aid System (CBDAS) that provides the designer during the design process cases that bear relevance to the current design issues at hand. The work on the Fabel project (Coulon 1995; Schaaf and Voss 1995) is most related to the current paper. The perspective from generic representations forms the main difference, focussing on the early stages of design. The work of Gross et al. (1988) and Gross (1990; 1996) has been informative on the development of graphic units and generic representations.

2. Graphic Representations

Graphic representations vary to a great extent in their appearance. A survey of historical examples (Achten 1997, p. 15-21) yields the following observations:

- Under the assumption of a particular convention of depiction (plan, perspective, section, etc.) a graphic representation does not need extensive textual elaboration in order to produce a correct interpretation. This means that the constituent graphic elements of the graphic representation provide strong clues about the interpretation, and that these elements do not change very much over a long period of time (Medieval drawings for example, are still intelligible).
- In a graphic representation, the identified elements are not the most basic elements (vertices, lines, planes, etc.) but aggregates of these elements with a particular interpretation. For example, a closed polygonal shape with constant thickness and particular hatching indicates a wall, a closed filled-in circle indicates a column, or a set of lines and circles indicates a vault system.
- An architectural graphic representation that makes sense in a particular convention of depiction and encoding presents a feasible and well-balanced whole. In a well-constructed graphic

representation a number of design conflicts between elements are solved.

2.1. GRAPHIC UNITS AND GENERIC REPRESENTATIONS

On the basis of the observations stated above, it is proposed that it is possible to classify graphic representations on the basis of their constituent elements to determine what design decisions are involved in that particular graphic representation. A constituent element is called a graphic unit. It is a set of graphic entities that has a specific meaning such as grid, axis, or shape. This combination of form and interpretation is crucial to the concept (Figure 1). Each interpretation has different implications in a design context.

Figure 1: The set of lines (left) can be a grid, a contour in grid, or a schematic subdivision. The particular interpretation means different things in an architectural context.

Graphic representations that have the same graphic units fall under the same generic representation, however varied their appearance may be. Graphic units and generic representations have been identified in an analysis of over 200 graphic representations (Achten 1997a).

24 Graphic units have been identified, such as (1) *simple contour*, (2) *contour*, (3) *measurement device*, and (4) *specified form*. It has been found that up to four graphic units constitute generic representations. 50 Generic representations have been identified, such as (15) contour in grid, (28) element vocabulary in grid, and (41) schematic subdivision in grid and refinement grid (see Achten (1997a; 1997b; 1998) for an extensive discussion of graphic units and generic representations).

2.2. SUCCESSIVE GRAPHIC UNITS

In a design process, one can identify sequences of design decisions. For example: before the particular length of a building wing is decided upon in a design, the decision has been taken that the shape of the building actually constitutes a number of wings. In terms of graphic units this means that the

graphic unit: (2) *contour* (a shape with no particular dimensions), is established before the graphic unit: (4) *specified form* (a contour with particular dimensions). The *specified form* forms the precondition for (5) *elaborated structural contour* (a shape with particular dimension whose edge has been detailed). Therefore, it is possible to define the sequence of successive graphic units: (2) → (4) → (5).

This principle of moving from less specific to more specific can be used to delineate successions of graphic units as they become more developed. In this way, 12 sequences of successive graphic units are identified (Table 1).

Table 1: Successive graphic units. A node is a graphic unit. → Indicates a more specified graphic unit.

The names of the successive graphic units are:

(2) *Contour* → (4) *specified form* → (5) *elaborated structural contour*.

(1) *Simple contour* → (4) *specified form*.

(2) *Contour* → (6) *Complementary contours*.

(7) *Function symbols* → (8) *Zone* → (21) *Functional space* → (19) *Element vocabulary*.

(19) *Element vocabulary* → (24) *Combinatorial element vocabulary*.

(7) *Function symbols* → (24) *Combinatorial element vocabulary*.

(11) *Modular field* → (16) *grid* → (12) *refinement grid* → (17) *tartan grid* → (18) *structural tartan grid*.

(19) *Element vocabulary* → (20) *structural element vocabulary*.

(3) *Measurement device* → (23) *proportion system*.

(10) *Schematic subdivision* → (22) *partitioning system*.

(13) *Schematic axial system* → (15) *axial system*.

(26) *Circulation scheme* → (27) *circulation*.

Successive graphic units are used when matching generic representations need to be found in a CBD context (Section 3.3).

3. Graphic Units and Generic Representations in a Case-Based Design Aid System

Further elaboration of the theoretical work in this paper is done in the context of a hypothetical Case-Based Design Aid System (CBDAS). The hypothetical system has the following components:

1. Interface for generating design drawings.
2. Establishing graphic units in the design drawing.
3. Composing a query for the index of generic representations.
4. Retrieving the appropriate set of generic representations.
5. The database of generic representations.
6. Presenting the retrieved set of generic representations and cases.

This outline serves to provide a framework of reference rather than to discuss in extensive detail all parts of such a system. Most emphasis will be on parts 2, 3, and 4.

3.1. SYSTEM PART 1: INTERFACE FOR GENERATING DESIGN DRAWINGS

Graphic representations are the medium through which the CBDAS must operate. Currently, sketching tools receive much attention in research, and they are used for a varied number of applications (see for example Qian and Gross 1999, Stellingwerff 1999, Richens 1999, and Trinder 1999). The particular operation of a sketching tool is not the main focus of this paper, so we will have to assume that a sketch or draw functionality can be implemented. This can either be on a CAD-like vector based platform, such as AutoCAD, or a paint-like pixel-based platform such as PhotoShop.

3.2. SYSTEM PART 2: ESTABLISHING GRAPHIC UNITS IN THE DESIGN DRAWING

In the drawing environment, the designer works on his designs. At some point, the CBDAS must read the drawing, recognize the graphic units that make up the drawing, and address the case base.

Graphic unit recognition can either be fully automated or designer-assisted. Automated graphic unit recognition (through neural networks or shape grammars) is beyond the scope of the research work. The designer-assisted approach can be either straightforward (ask the designer to identify graphic units), or assist the designer in identifying. The first approach assumes the designer to be knowledgeable about the theoretical work underlying the system, which usually is not the case. Furthermore, such actions are prone to mistakes and they distract from the design work. The second approach guides the designer in a limited number of steps through a decision tree to determine which graphic unit is worked with at the moment.

Ultimately, automated graphic unit recognition needs to be implemented in a CBDAS. We have chosen the designer-assisted decision-tree approach for the following reasons:

- Before automation, the various ambiguities and decision points in identification of graphic units need to be clearly defined. The decision-tree can form a starting point for this.
- It enables faster implementation in a design prototype that will form ‘proof of concept’ for the current notions, in particular, if cases retrieved on the basis of matching graphic units make sense in a design context.

3.2.1. A Decision-Tree Based on Graphic Units

The decision-tree is a question-answer mechanism that leads to identification of a graphic unit in a drawing. The nodes are not bifurcal in order to decrease the number of questions that need to be answered. The process starts when the designer prompts the system for case retrieval. Each pass through the tree identifies one graphic unit (Figure 2).

Figure 2: The decision-tree for identifying graphic units in a graphic representation.

Each node in the decision-tree either is a question or an identified graphic unit. The questions are identified with capital letters A, B, C, etc. The graphic units are identified with their numbers. At each decision point,

a schematic drawing clarifies the decision that has to be taken. Textual clarification is also available, but for brevity's sake this is not included. Below follows the list of questions in the decision-tree.

A. Is it a graphic or symbolic element?

- If graphic element, go to B.
- If symbolic element, graphic unit is (7) function symbols.

B. Is it a closed shape or a set of one or more lines?

- If a closed shape, go to C.
- If a set of one or more lines, go to D.

C. Is it a single shape or multiple shapes?

- If multiple shapes, go to E.
- If single shape, go to F.

D. Is it a coordinating system or not?

- If coordinating system, go to G.
- If not a coordinating system, go to H.

E. Is it spaces or building elements?

- If spaces, go to I.
- If building elements, go to J.

F. Is it a Platonic shape, a more irregular shape, or a shape that is specified?

- If Platonic shape, graphic unit is (1) simple contour.
- If irregular shape, graphic unit is (2) contour.
- If specified shape, graphic unit is (4) specified form.

G. Is it a zone, grid, or proportion system?

- If zone, graphic unit is (8) zone.
- If grid, go to K.
- If proportion system, graphic unit is (23) proportion system.

H. Is it for measurement, subdivision, composition, or circulation?

- If measurement, graphic unit is (3) measurement device.
- If subdivision, go to L.
- If composition, go to M.
- If circulation, go to N.

I. Is the contour elaborated and showing the structure, or are the contours complementary, or do the spaces have functional indications?

- If contour is elaborated, graphic unit is (5) elaborated structural contour.
- If the contours are complementary, graphic unit is (6) complementary contours.
- If the spaces have functional indications, graphic unit is (21) functional space.

J. Is it a set of elements, structural elements, or rules for elements?

- If set of elements, graphic unit is (19) element vocabulary.
- If structural elements, graphic unit is (20) structural element vocabulary.
- If rules for elements, graphic unit is (24) combinatorial element vocabulary.

K. Is it a modular field, grid, refinement grid, tartan grid, or structural tartan grid?

- If modular field, graphic unit is (11) modular field.
- If grid, graphic unit is (16) grid.
- If refinement grid, graphic unit is (12) refinement grid.
- If tartan grid, graphic unit is (17) tartan grid.
- If structural tartan grid, graphic unit is (18) structural tartan grid.

L. Is it a schematic subdivision or a concrete partitioning?

- If schematic, graphic unit is (10) schematic subdivision.
- If concrete, graphic unit is (22) partitioning system.

M. Is it a schematic axial system, or a concrete axial system?

- If schematic, graphic unit is (13) schematic axial system.
- If concrete, graphic unit is (15) axial system.

N. Is it a schematic circulation scheme, or a concrete circulation system?

- If schematic, graphic unit is (26) circulation scheme.
- If concrete, graphic unit is (27) circulation.

Note that the questions first aim to isolate groups of graphic representations that have a meaningful resemblance (shapes, sets of lines, spaces, building elements, etc.), and then differentiate on the basis of their interpretation (grid, axial system, schematic subdivision, etc.) to derive the specific graphic unit. This method is not specific enough for automated graphic recognition, but as a question-answer mechanism for a designer it is adequate.

The decision-tree leads in a maximum of five questions to a graphic unit. This process needs to be reiterated until all the elements of the drawing have been identified, although a user may decide to have only part of the drawing interpreted.

3.3. SYSTEM PART 3: COMPOSING A QUERY FOR THE INDEX OF GENERIC REPRESENTATIONS

The use of system part 2 results in a list L of graphic units that have been identified in the graphic representation:

$$L_{gu} \{gu_1, gu_2, \dots, gu_n\}, n \in N.$$

This list is the basis for composing the query that will search the case base for matching generic representations. If only this list will be used, then the matching is a simple look-up whether the generic representation exists, and then retrieval of the associated graphic representations follows. However, drawings evolve throughout the design process, and this simple matching procedure could miss cases that are related but do not belong to the same generic representation. By means of successive graphic units it is possible to identify drawings that are closely related to the current graphic representation. Also, drawings that are less complex may be relevant. The query for case matching therefore, needs to be expanded.

3.3.1. Expanding the Query

As discussed in Section 2.2, graphic units can have successive graphic units. In a sequence of successive graphic units, a graphic unit resembles to some extent its predecessor or successor, as it becomes defined to a more or lesser degree. The relevance of this aspect for the CBDAS is the following.

Consider $L_{gu} \{gu_1, gu_2, \dots, gu_n\}$ and $M_{gu} \{gu_1, gu_x, \dots, gu_n\}$ where gu_x is a successive graphic unit of gu_2 . Then M_{gu} differs in one graphic unit from L_{gu} . The different graphic unit gu_x is a more or less defined variant of gu_2 . For example, gu_2 could be (13) *schematic axial system*, and gu_x could be (15) *axial system*. A case with an axial system can be informative for the designer, even when he is in the stage when considering an axial system only in a schematic form. M_{gu} therefore, bears much resemblance with L_{gu} , and should be included in the search for matching generic representations.

It is also possible to consider the cases where two elements of M_{gu} differ as successive graphic units with respect to L_{gu} , where there are three successive graphic units, etc.

Subsets of L_{gu} (where M_{gu} has a smaller number of graphic units) can be considered for matching as well. Since subsets resolve a lesser number of graphic units, they have reduced complexity with respect to the graphic representation under consideration. The relevance of subsets therefore, as matches in the query, will be less.

In order to take note of the differences between matches, sets of M_{gu} need to be weighted against L_{gu} . For this purpose, we have established a list of weighted successive graphic units (Table 2).

The weights have been established in the following manner. A *modular field* for example, does not differ much from a *grid*. Both consist of a system of lines that delineate where elements can be placed. A *refinement grid* however, adds an additional grid, thus increasing the complexity. Therefore, a succession from (11) *modular field*, to (16) *grid*, makes a lesser difference (weight 1) than a succession from (16) *grid*, to (12) *refinement grid* (weight 2).

A subset of L_{gu} is considered to deviate more than a set of M_{gu} where two graphic units have been substituted by a successive graphic unit. The maximum weight in the case of two successive graphic units is 4 (2+2). Therefore, a weight factor of 5 is set for each graphic unit of L_{gu} that is omitted.

Table 2. Weighted successive graphic units. Nodes are graphic units, weights are set on the vertices. The vertices are directed to indicate more elaborated graphic units.

3.3.2. Finding and Rank-Ordering All Queries

All possible combinations of L_{gu} with successive graphic units that will form query Q, can be generated by the following method:

1. For each gu_j in L_{gu} , substitute with its predecessor, successor, or itself. A predecessor s_j is a graphic unit in the table of successive graphic units that precedes gu_j . A successor s_j is a graphic unit in the table of successive graphic units that succeeds gu_j . Each single substitution in L_{gu} forms a new list of graphic units $(s_1, s_2, \dots, s_n)$.
If gu_j has been substituted by a predecessor or successor, add the weight stated in the table of weighted successive graphic units to the current weight (initial value is 0). The total weight W determines the rank-order of the new combination. Add the tuple $\{(s_1, s_2, \dots, s_n), W\}$ to Q (initial set is empty).
2. For each subset of L_{gu} repeat step 1.
3. Rank-order the found list Q by increasing weight.

The result is stored in Q, which is the query that will be used in system part 4 for case retrieval.

3.3.3. An Example of a Query Result

Suppose in a graphic representation (Figure 3) the following graphic units have been identified: (10) schematic subdivision, (13) schematic axial system, and (6) complementary contours.

Figure 3. Example of a graphic representation with three graphic units. The orthogonal and cross-shape are (6) complementary contours, the bold lines are (13) schematic axial system, and the thin lines are (10) schematic subdivision.

L_{gu} then is: $\{10,13,6\}$.

Using Table 2, we find that graphic unit (10) *schematic subdivision*, has no predecessor, and one successor: (22) *partitioning system*, with weight 1. Graphic unit (13) *schematic axial system* has no predecessor, and one successor: (15) *axial system*, with weight 1. Graphic unit (6) *complementary contours*, has one predecessor: (2) *contour*, with weight 2, and no successor. Together with the subtraction of one or two elements from L_{gu} , this is everything necessary to generate Q .

Q then, rank-ordered on increasing weights, is:

$\{(10,13,6),0\}, \{(22,13,6),1\}, \{(10,15,6),1\}, \{(10,13,2),2\},$
 $\{(22,15,6),2\}, \{(10,15,2),3\}, \{(22,13,2),3\}, \{(22,15,2),4\}, \{(10,13),5\},$
 $\{(10,6),5\}, \{(13,6),5\}, \{(10,15),6\}, \{(22,13),6\}, \{(22,6),6\}, \{(15,6),6\},$
 $\{(22,15),7\}, \{(10,2),7\}, \{(13,2),7\}, \{(22,2),8\}, \{(15,2),8\}, \{(10),10\},$
 $\{(13),10\}, \{(6),10\}, \{(22),11\}, \{(15),11\}, \{(2),12\}$

The rank-order indicates the potential match in query Q . Weight 0 indicates a complete match. Queries with weight 1 and 2 are partial matches that deviate in one graphic unit which is a successive graphic unit. Higher weights indicate more differences with the graphic representation at hand.

3.4. SYSTEM PART 4: RETRIEVING THE APPROPRIATE SET OF GENERIC REPRESENTATIONS

The approach of generic representations has the major advantage that aspects on which the cases are stored (graphic units) are either present in the graphic representation or not. There is no problem with range of values or

limit values on which matching has to occur. The process aspect of design has been addressed via successive graphic units and subsets in the query.

Generic representations are listed as their set of graphic units (Table 3). Query Q matches its lists against Table 3 to retrieve all cases. Since Q already is rank-ordered, the match also is rank-ordered in relevance.

Table 3: Matching table for graphic units and generic representations

Generic representation		Graphic units	
(1) Simple contour			1
(2) Combination of contours			1
(3) Complementary contours			6
(4) Modular field			11
(5) Proportion system			23
(6) Multiple grids			16
(7) Functional spaces			21
(8) Schematic axial system			13
(9) Schematic subdivision			10
(10) Elaborated structural contour			5
(11) Element vocabulary			19
(12) Combinatorial element vocabulary			24
(13) Circulation scheme			26
(14) Proportion system in contour	2		23
(15) Contour in grid	2		16
(16) Zone in specified form	4		8
(17) Function symbols in combination of contours	2		7
(18) Axial system in specified form	4		15
(19) Schematic subdivision in grid	10		16
(20) Schematic subdivision with function symbols	7		10
(21) Schematic subdivision in contour	2		10
(22) Partitioning system in contour	2		22
(23) Specified elaborated structural contour	3		5
(24) Elaborated structural contour in grid	5		16
(25) Elaborated structural contour in complementary contours	5		6
(26) Elaborated structural contour and axial system	5		15
(27) Elaborated structural contour and function symbols	5		7
(28) Element vocabulary in grid	16		19
(29) Element vocabulary in multiple grids	16		19
(30) Combinatorial element vocabulary in grid	16		24
(31) Combinatorial element vocabulary in specified form	3	4	24

(32) Circulation in contour	2	27
(33) Circulation scheme in elaborated structural contour	5	26
(34) Structural element vocabulary in structural tartan grid	18	20
(35) Proportion system in elaborated structural contour in tartan grid	5	17
(36) Zone in contour in grid	2	8
(37) Axial system in contour in grid	2	15
(38) Axial system in contour in tartan grid	2	15
(39) Axial system in specified form in structural tartan grid	4	15
(40) Schematic subdivision in grid and refinement grid	10	12
(41) Schematic subdivision and schematic axial system in contour	2	10
(42) Elaborated structural contour and function symbols and axial system	5	7
(43) Element vocabulary in zone and contour	2	8
(44) Circulation in contour in grid	2	16
(45) Structural element vocabulary in contour in modular field	2	11
(46) Structural element vocabulary in structural tartan grid and refinement grid	4	12
(47) Structural element vocabulary in axial system in contour	2	15
(48) Element vocabulary and function symbols and grid in specified form	4	7
(49) Schematic subdivision in zone in contour with function symbols	2	7
(50) Structural element vocabulary in axial system in contour in grid	2	15

3.4.1. An Example of a Retrieval Result

In the example given in Section 3.3.3, where L_{gu} is $\{10,13,6\}$, the result of the retrieval is the following list of generic representations:

1. (41) Schematic subdivision and schematic axial system in contour, weight 2, graphic units (2,10,13).
2. (21) Schematic subdivision in contour, weight 7, graphic units (2, 10).
3. (22) Partitioning system in contour, weight 8, graphic units (2,22).
4. (9) Schematic subdivision, weight 10, graphic units (10).
5. (8) Schematic axial system, weight 10, graphic units (13).

6. (3) Complementary contours, weight 10, graphic units (6).

There is no complete match in the resulting match. The first hit deviates with respect to the graphic unit *complementary contours* that was found in the graphic representation. It has become *contour* in the generic representation. The survey from Achten (1997a) yields the following graphic representations belonging to the first two generic representations found in the match (Figure 4 and Figure 5):

Figure 4. Case retrieval of graphic representations belonging to (41) Schematic subdivision and schematic axial system in contour. Redrawn from Durand (1804), *Parties*, Planche 4 (left) and Planche 5 (right). The '+' and '-' are simplified notations for axes.

Figure 5. Case retrieval of graphic representations belonging to (21) Schematic subdivision in contour. From Tzonis (1986), p. 31 (left) and Wittkower (1973), p. 73 (right). The partial match omits the schematic axial system.

3.5. SYSTEM PART 5: THE CASE BASE OF GENERIC REPRESENTATIONS

Each generic representation that is found in the retrieval procedure is a category of a group of graphic representations that share common design decisions. The generic representation forms the index through which all particular graphic representations belonging to it are retrieved and presented to the user.

The cases that belong to the generic representation depict different designs. Since they belong to the same generic representation, they share the design decisions that the designer is working on in his particular design. This information is stored with the generic representation.

The case base stores the particular graphic representations separately from the generic representation, with an index to the generic representation. The information on the graphic representations can be in any format, as at the moment case adaptation is not an option.

In the example, there is no complete match for $L_{gu} \{10,13,6\}$. Since the constitutive graphic units are known, and the case base is built on graphic units, the particular graphic representation of the example can now be added to the case base, thus making it available for the next time.

3.6. SYSTEM PART 6: PRESENTING THE SET OF GENERIC REPRESENTATIONS AND CASES

After zero, one, or more generic representations have been found to match the set of graphic units in the design drawing, they are presented to the designer. The presentation shows the list of all found generic representations, as well as the graphic representations that belong to a selected generic representation. Each case is discussed in terms of the design decisions it addresses. This can then aid the designer in the work at hand.

4. Discussion and Future Work

The system and techniques outlined in this paper show how approaches from CBD can be utilized to provide a designer with relevant design-content information during the design process, based on a graphic indexing system that also takes into account some of the dynamics involved in design.

The procedure used here will identify all generic representations that have at least one graphic unit or successive graphic unit in common with the graphic representation. Typically, the first hits in the rank-order will have most relevance with the design situation at hand. The whole list of retrieved generic representations is showed to enable the designer to browse through cases that are not directly relevant, but that bear at least some resemblance to the project at hand. In this way, the CBDAS can become a context sensitive browsing tool.

The results of this work can be generalized to other designing disciplines that have well-established graphic conventions in the design process. The fields of mechanical engineering and industrial design seem to share these features with architectural design. The main task in expanding the work to other disciplines first lies in identifying graphic units and generic representations in these fields.

The current system-outline supports assisted graphic unit recognition, it assembles a rank-ordered list of relevant generic representations, and it can

address and retrieve cases of graphic representations that are related to the design problem at hand. A fully completed CBDAS has the following additional functionality: automated graphic unit recognition, a structured information model encoding the cases, some support of case manipulation and adaptation, and storage of new cases. Furthermore, such a system should be able to expand on its theoretical foundation: add new graphic units and generic representations. In the near future, a number of these aspects will be addressed in the ongoing research.

The cases stored in the system need not be completely adaptable for use in the current design. Since they derive from completely different designs, it is not likely to expect that they will have exactly the right elements and objects to make them easily adaptable to other solutions. The value of the cases lies in clarifying the design decisions at hand and showing how others addressed them. Some manipulation must be possible however, on the level of graphic units: for example, selecting the schematic axial system in a drawing, copying and pasting it in the current design, and then manipulating it for its new purpose. It is necessary therefore, to describe the cases in a formalism that allows such manipulation of parts. Research in the VR-DIS programme aims at establishing Feature-Based modeling as an approach for information modeling (van Leeuwen 1999). Work is being done to describe design states and design moves in architecture in terms of Features (Achten and van Leeuwen 1998, Achten and van Leeuwen 1999). A Feature-Based description of generic representations is the next step in the research. This also aids in linking inherently 2D sketching elements to 3D design information of a building design.

Acknowledgements

The presented work is done in the Design Systems group of the Department of Architecture, Building and Planning of Eindhoven University of Technology in the VR-DIS research programme. It builds on a Ph.D. work performed in the Design Methods Group. The work has been under discussion in the group, in particular with Bauke de Vries and Joran Jessurun, and during a guest lecturing week with Prof. Mary Lou Maher from the University of Sydney. Acknowledgements are also due to Jos van Leeuwen for reviewing and critiquing the paper.

References

- Achten, H. H.: 1997a, *Generic Representations – An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design*, Ph.D. thesis, Technische Universiteit Eindhoven/Faculteit Bouwkunde, Bouwstenen 46, Eindhoven.
- Achten, H.H.: 1997b, Generic representations – Typical design without the use of types, in R. Junge (ed), *Proceedings of the 7th International Conference on Computer Aided*

- Architectural Design Futures held in Munich, Germany, 4-6 August 1997*, Kluwer, Dordrecht, pp. 117-133.
- Achten, H.H. and Oxman, R.M. and Bax, M.F.Th.: 1998, Typological knowledge acquisition through a schema of generic representations, in J. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '98*, Kluwer, Dordrecht.
- Achten, H.H. and Mallory-Hill, S.M. and Vries, B. de: 1998, *Workshop Case Based Reasoning and Design*, Design System Report 1998/2, Eindhoven.
- Achten, H.H. and Leeuwen, J.P. van: 1998, A feature-based description technique for design processes: A case study, in H. Timmermans (ed), *Proceedings of the Fourth International Conference on Design and Decision Support Systems in Architecture and Urban Planning, July 26-29*, CD-ROM, Maastricht.
- Achten, H.H. and Leeuwen, J.P. van: 1999, Feature-based high level design tools: a classification, in G. Augenbroe and C. Eastman (eds), *Computers in Building: Proceedings of the CAADfutures '99 Conference*, Kluwer, Dordrecht, pp. 275-290.
- Chiu, M-L. and Shih, S-G.: 1997, Analogical reasoning and case adaptation in architectural design: computers vs. human designers, in R. Junge (ed), *Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures held in Munich, Germany, 4-6 August 1997*, Kluwer, Dordrecht, pp. 787-800.
- Coulon, C-H.: 1995, Automatic indexing, retrieval, and reuse of topologies in architectural layouts, in M. Tan and R. Teh (eds), *The Global Design Studio – Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures*, Singapore, pp. 577-586.
- Durand, J.-N.-L.: 1804, *Précis des Leçons d'Architecture*. Paris.
- Grassi, M. de and Giretti, A. and Pinese, P.: 1999, Knowledge structures as episodic memory in architectural design: An example of protocol analysis, in Brown, A. and Knight, M. and Berridge, P. (eds), *Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe*, The University of Liverpool, Liverpool, pp. 576-583.
- Groot, E. de and Mallory-Hill, S.M. and Zutphen, R.H.M. and Vries, B. de: 1999, A decision support system for preliminary design, in M.A. Lacasse, and D. Vanier (eds), *Proceedings of the 8th International Conference on Durability of Building Materials and Components*, Vancouver, pp. 970-979.
- Gross, M.D. and Ervin, S.M. and Anderson, J.A. and Fleisher, A.: 1988, Constraints: knowledge representation in design, *Design Studies*, 9(3), p. 133-143.
- Gross, M.D.: 1990, Relational modeling: A basis for computer-assisted design, in M. McCullough and W.J. Mitchell and P. Purcell (eds), *The Electronic Design Studio - Architectural Knowledge and Media in the Computer Era*, The MIT Press, Cambridge, Massachusetts.
- Gross, M.D.: 1996, The Electronic Cocktail Napkin - computer support for working with diagrams, *Design Studies*, 17(1), 53-70.
- Heylighen, A. and Segers, R. and Neuckermans, H.: 1998, Prototype of an interactive case library for architectural design, in H. Timmermans (ed), *Proceedings of the Fourth International Conference on Design and Decision Support Systems in Architecture and Urban Planning, July 26-29*, CD-ROM, Maastricht.
- Heylighen, A. and Neuckermans, H.: 1999, Learning from experience: Promises, problems and side-effects of CBD in architecture, in Brown, A. and Knight, M. and Berridge, P. (eds), *Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe*, The University of Liverpool, Liverpool, pp. 567-575.
- Kolodner, J.: 1993, *Case-Based Reasoning*, Morgan Kaufman, San Mateo.

- Lee, H. and Lee, J.: 1999, Design adaptation for handling design failures, in M. Tan and R. Teh (eds), *The Global Design Studio – Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures*, Singapore, pp. 567-576.
- Leeuwen, J.P. van: 1999, *Modelling Architectural Design Information by Features*, Ph.D. Thesis, Eindhoven University of Technology/Faculteit Bouwkunde, Bouwstenen 54, Eindhoven.
- Maher, M.L. and Pu, P.: 1997, *Issues and Applications of Case-Based Reasoning in Design*, Erlbaum, London.
- Mallory-Hill, S.M.: 1998, Building a case-based design assistant for workplace environment design, in H. Timmermans (ed), *Proceedings of the Fourth International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, July 26-29, CD-ROM, Maastricht.
- McFadzean, J. 1999: Computational Sketch Analyser (CSA): Extending the boundaries of knowledge in CAAD, in Brown, A. and Knight, M. and Berridge, P. (eds), *Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe*, The University of Liverpool, Liverpool, pp. 503-510.
- Oxman, R.: 1995, Design case bases: graphic knowledge bases for the design workspace, in M. Tan and R. Teh (eds), *The Global Design Studio – Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures*, Singapore, pp. 555-565.
- Qian, D. and Gross, M.D.: 1999, Collaborative design with Netdraw, in Augenbroe, G. and Eastman, C. (eds), *Computers in Building – Proceedings of the CAADfutures '99 Conference*, Atlanta, pp. 213-226.
- Richens, P.: 1999, The Piranesi system for interactive rendering, in Augenbroe, G. and Eastman, C. (eds), *Computers in Building – Proceedings of the CAADfutures '99 Conference*, Atlanta, pp. 381-398.
- Schaaf, J. and Voss, A.: 1995, Retrieval of similar layouts in FABEL using AspecT, in M. Tan and R. Teh (eds), *The Global Design Studio – Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures*, Singapore, pp. 617-626.
- Stellingwerff, M.: 1999, SketchBoX, in Brown, A. and Knight, M. and Berridge, P. (eds), *Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe*, The University of Liverpool, Liverpool, pp. 491-497.
- Tan, M. and Gan, J. and Indorf, P. and Man, D. and Teh, R. and Datta, S.: 1995, Multivalent architectural case information for creative reasoning, in M. Tan and R. Teh (eds), *The Global Design Studio – Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures*, Singapore, pp. 543-554.
- Trinder, M.: 1999, The computer's role in sketch design: A transparent sketching medium, in Augenbroe, G. and Eastman, C. (eds), *Computers in Building – Proceedings of the CAADfutures '99 Conference*, Atlanta, pp. 227-244.
- Tzonis, A. and Lefaivre, L.: 1986, *Classical Architecture. The Poetics of Order*, MIT Press, Cambridge, Massachusetts.
- Verstijnen, I.M.: 1997, *Sketches of Creative Discovery – A Psychological Inquiry Into the Role of Imagery and Sketching in Creative Discovery*, Ph.D. Thesis, Delft University of Technology, Delft.
- Wittkower, R.: 1973, *Architectural Principles in the Age of Humanism*. Academy Editions, London.