České vysoké učení technické v Praze
Fakulta architektury

Czech Technical University in Prague
Faculty of Architecture

Dr.ir. Henri Hubertus Achten

Teorie, metoda a počítač – tři cesty porozumění moderního architektonického navrhování pomocí počítace

Theory, method, and the computer – three pathways to understand modern architectural design with the computer

1. Summary

Architects extensively use graphic representations to communicate their design ideas personally, between professionals, and others. The design drawing therefore, is an important medium if we want to establish design support by means of computers. In order to make drawings accessible for computers, it is necessary to understand how design drawings are carriers of knowledge.
In this text we outline our research on this question. We investigate how graphic representations are structured, what these structures are, what their knowledge content is, and how they can be applied in Computer Aided Architectural Design as intelligent drawing tools.

2. Souhrn

Architekti hojně využívají grafická zobrazení pro komunikaci svých myšlenek mezi sebou – mezi profesionály, ale i laiky – zákazníky. Výkres projektu je tedy důležité médium pokud chceme vybudovat podpůrný systém pomocí počítače. Aby se kresba stala „srozumitelná“ pro počítač, je nutné porozumět tomu, co je v kresbách hlavním nositelem informace. V této přednášce nastiňujeme náš výzkum této otázky. Zabýváme se jak jsou strukturována grafická zobrazení, co jsou tyto struktury, jakou informaci nesou, a jak mohou být použity v počítačově podporovaném návrhu jako inteligentní nástroje.

Klíčová slova:	výkres, metoda projektování, stavební typ, počítačem podporované architektonické navrhování
Keywords:	drawing, design method, building type computer aided architectural design

---

České vysoké učení technické v Praze

Název:	Teorie, metoda a počítač – tři cesty porozumění moderního architektonického navrhování pomocí počítace
Autor:	Dr.ir. Henri Hubertus Achten
Počet stran:	20
Náklad:	výtisků

ISBN

Contents

3. Introduction

What is the architect doing when he or she is designing? A design project usually starts out with a desire of a client to have a new building or to change an existing building – for example “I want a new house,” “we need a more modern office building,” or “there should be a representative museum.” At the start, information such as required floor surface area, which spaces have to be next to each other, what necessary lighting levels should be, and so forth is missing. To create a design, the architect has to do two things: on the one hand to generate the necessary knowledge, and on the other hand to think up of a solution. These activities are connected: knowledge is gained while creating a solution, and a solution is created using available knowledge. Architectural design therefore, is both speculative and explorative, as well as analytical and investigative (Lawson 1990; Schön 1983).

From the above characterisation, it follows that knowledge is a critical aspect in architectural design (or in any design domain, for that matter). The general theme in this research work is the application of knowledge in the design process. Architects use all kinds of knowledge when designing: knowledge of existing buildings, earlier solutions, norms, rules, heuristics, strategies, styles, and so forth. The kind of knowledge we investigate in this research is knowledge of *building types* and knowledge of *design processes*. This knowledge is used to create a design. The design itself is documented with external representations: sketches, drawings, scale models, computer models, etc. In their final form, they document the building design so that later someone can realise the building. This means that knowledge is stored in the external representations. The question is, how this is done, and how knowledge of building types and design processes is connected to external representations. Because of the central role of the drawing, we look only at *knowledge in design drawings* in this research.

The current overview is built up as follows. First, the drawing as carrier of knowledge is discussed. We identify graphic units as building blocks of design drawings. This sets up the central framework for the research. Second, the design process as methodical series of steps is analysed. We demonstrate how successive sequences of graphic units create steps of a design method. The method therefore, provides *procedural knowledge* (process knowledge) about the design. Third, a brief theoretical overview of building types is presented and how we view the building type in this research. The building type provides *declarative knowledge* (factual knowledge) for the design process. Fourth, with the use of the computer, it is possible to describe the relationship between building type knowledge, design process knowledge, and the design drawing. Finally, we show how this research work helps to support and understand architectural design. For details, the reader is referred to Achten (2004).

4. The drawing as carrier of knowledge

In order to understand drawings as carriers of knowledge, consider the following two design drawings (Figure 1).

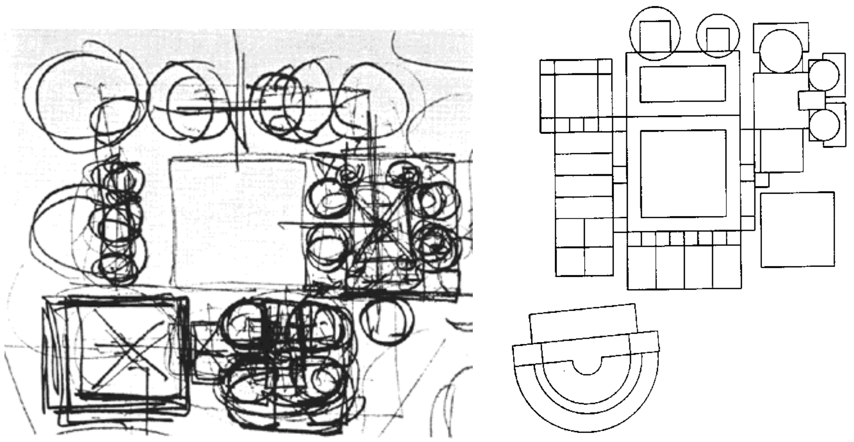

Figure 1. A: Sketch design by Louis Kahn. B: Line drawing after design by Louis Kahn. Images taken from Ronner, Jhaveri and Vasella (1977) and Mitchell and McCullough (1991).

The left drawing is a sketch design, and the right drawing is a line drawing. The difference between the two drawings lies in their appearance, but not in what they actually depict. Both drawings show a composition of spaces. The spaces are indicated with squares, circles, rectangles, and other closed shapes. Knowledge in both drawings deals with the spaces that are in the drawing, composition of the whole, the location of spaces, and the dimensions of the spaces. The sketch design is more ambiguous in the sense that location, shape, and size of the spaces is less precisely defined than in the line drawing.

Because of the lower degree of ambiguity, it is easier to analyse line drawings than sketch drawings. Therefore, they are more suited for the research work. Later development of the work has to be extended however, to include sketch drawings as well.

4.1. Graphic units in the drawing

In order to understand a design drawing, one should not look at the drawing as a whole, nor at the most primitive elements (lines and marks) in the drawing, but at composite primitive elements that have meaning such as space, grid, axis, etc.

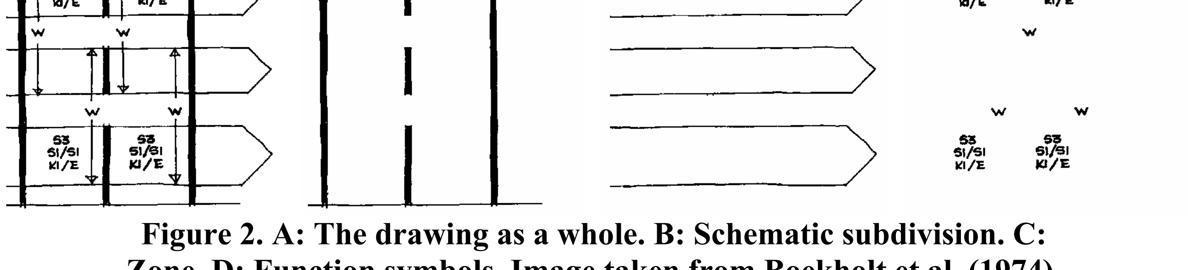

Figure 2. A: The drawing as a whole. B: Schematic subdivision. C: Zone. D: Function symbols. Image taken from Boekholt et al. (1974).

In Figure 2, composite primitive elements that have meaning are the schematically subdivided plan (B), the zones in the drawing (C), and the function symbols indicating different possible uses of the zones (D). These elements are called *graphic units*. As can be seen from the examples above, a graphic unit has two parts: the graphic description (what it looks like), and the meaning to the architect (what it represents). A drawing that consists of only graphic units is called a *generic representation*. In our research, we have identified 24 graphic units and 50 generic representations (Achten 1997). There are 12 graphic units that structure the design and 12 graphic units that represent objects.

The graphic units that structure the design are: *measurement device, zone, schematic subdivision, modular field, grid, refinement grid, tartan grid, structural tartan grid, schematic axial system, axial system, proportion system,* and *circulation system*.

The graphic units that describe a design are: *simple contour, contour, specified form, elaborated structural contour, complementary contours, function symbols, element vocabulary, structural element vocabulary, combinatorial element vocabulary, functional space, partitioning system,* and *circulation*.

The structuring and descriptive graphic units form the building blocks of design drawings. They are in fact the graphical instruments that the architect uses to design.

4.2. Design drawings and knowledge

With the definition of graphic units and generic representations, the question how knowledge is embedded in design drawings can be formulated more precisely. The question now is: what knowledge is associated with a particular generic representation? The restriction to generic representations means that elements which are not classified as graphic units are not considered in the research.

5. Method: the design process as carrier of procedural knowledge

Consider the following five graphic units: *modular field, grid, refinement grid, tartan grid*, and *structural tartan grid* (Figure 3).

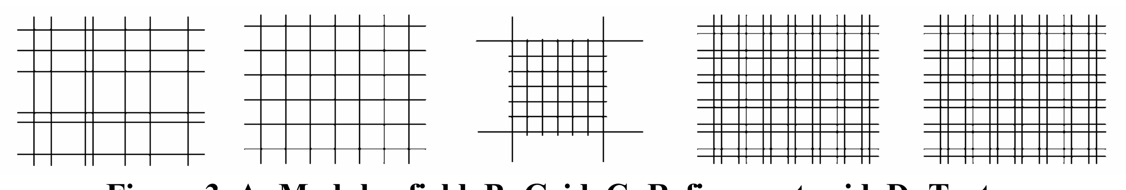

Figure 3. A: Modular field. B: Grid. C: Refinement grid. D: Tartan grid. E: Structural tartan grid.

A modular field consists of a set of regulating lines that coordinate the location of objects and spaces. A grid is a modular field, however, with a constant distance (module) applied to the regulating lines. A refinement grid is an additional subdivision of a grid with a smaller module. A tartan grid is a refinement grid with all but one band removed. A structural tartan grid is a tartan grid with one band reserved for structural elements such as walls and columns.

What happens in this sequence from modular field to structural tartan grid, is that knowledge is added as the graphic units become more and more specific. The modular field has the greatest amount of freedom (and the least degree of structure). The structural tartan grid is the most strictly defined (and offers the most structure). The series of graphic units therefore, can be seen as a small implied design process, in which increasingly more knowledge is added. Observation of the identified graphic units leads to 12 sequences of graphic units (of which the one discussed above is one example). Each sequence is a small cluster of related design decisions (such as grids and modules in the example) – see Table 1.

1	Contour → specified form → elaborated structural contour.
2	Simple contour → specified form.
3	Contour → complementary contours.
4	Function symbols → zone → functional space → element vocabulary.
5	Element vocabulary → combinatorial element vocabulary.
6	Function symbols → combinatorial element vocabulary.
7	Modular field → grid → refinement grid → tartan grid → structural tartan grid.
8	Element vocabulary → structural element vocabulary.
9	Measurement device → proportion system.
10	Schematic subdivision → partitioning system.
11	Schematic axial system → axial system.
12	Circulation scheme → circulation.

Table 1: Sequences of graphic units.

Putting the groups of sequences of graphic units into a larger series of subsequent graphic units leads to a design method: a prescribed series of steps that leads to a design. The order of the groups is not predetermined, as it depends on the goals and strategy of the architect. Table 2 shows one particular sequence of generic representations.

1	Simple contour	Establish the general shape of the design, by creating a building envelope.
2	Combination of contours	Tentatively define major parts of the building envelope.
3	Specified form	Tentatively define the dimensions of the building envelope.
4	Complementary contours	Locate the building shape in the site, taking care that the surrounding area gets its own shape.

5	Zone	Establish organizational zoning of the building plan.
6	Schematic subdivision in zone	Establish general layout within the zone.
7	Schematic subdivision in zone in contour with function symbols	Mark functions in the schematically subdivided zone and apply it to the building envelope.
8	Zone in specified form	Determine the zoning in the tentative dimensions of the building.
9	Schematic subdivision	Come up with a subdivision principle to distinguish main parts of the building.
10	Schematic subdivision in contour	Apply the subdivision principle to the contour of the building.
11	Grid	Establish a grid module.
12	Schematic subdivision in grid	Coordinate the main parts of the building to the grid.
13	Schematic subdivision in specified form	Apply the subdivision principle to the tentative building layout.
14	Schematic axial system	Establish a set of axes for the building organisation.
15	Axial system in specified form	Apply the axes to the tentative building layout.
16	Contour in grid	Apply the building envelope to the grid.
17	Zone in contour in grid	Apply the zoning to the building envelope and coordinate it with the grid.
18	Partitioning system in contour	Establish the partitioning within each of the main parts of the building.
19	Circulation scheme	Establish a circulation principle.
20	Circulation in contour	Apply the circulation principle to the building envelope.
21	Element vocabulary	Establish an element vocabulary to indicate functional use in the building.
22	Element vocabulary in contour	Apply the element vocabulary to the building.
23	Element vocabulary and function symbols and grid in specified form	Establish the functional layout of the building design.

Table 2: A sequence of generic representations.

5.1. Design method and knowledge

With the definition of sequences of generic representations, it is possible to define more precisely what procedural knowledge is. Procedural knowledge is embedded in the sequence of generic representations. It is the knowledge that is required to move from one generic representation to the next in the series.

6. Theory: building type as carrier of declarative knowledge

A building type is a group of buildings that share strong similarities in terms of organisation (open office, cell office, combi-office, etc.), composition (nine-square plan, central composition, wing type, etc.), or function (office, hospital, stadium, etc.) When an architect is familiar with a building type, he or she has a lot of general knowledge that can be used in the design process.

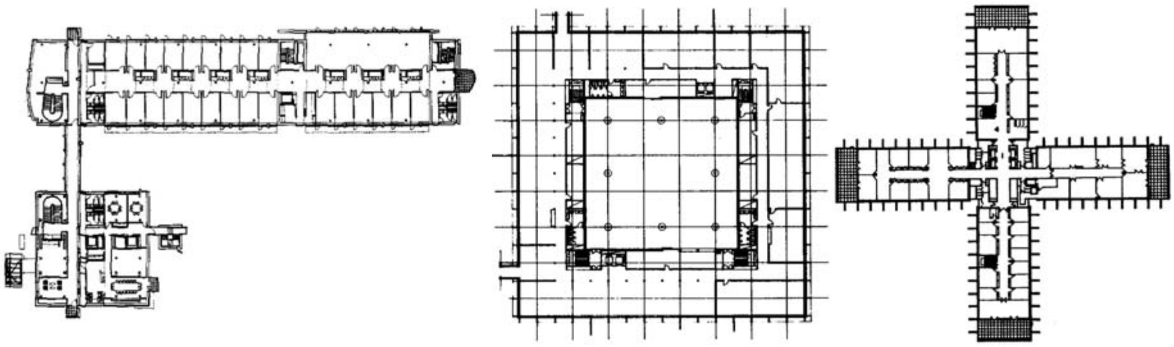

Figure 4. Three examples of office buildings with different organisation and composition. Images taken from Peters (1973).

What knowledge is exactly embedded in a building type, is a point of discussion in architectural theory. Roughly four views can be identified: ambiguity versus explicitness, and idealistic versus procedural. The *ambiguity* view holds that type cannot be defined in an explicit manner but only through its instances (de Quincy 1825, Argan 1963, Colquhoun 1981, Rossi 1982:40-41, Habraken 1985:27-28). Related-yet-different instances of the building type can be created by appeal to the indefiniteness of type. This approach offers in fact no mechanism or principle that can be studied more carefully. The opposite view, *explicitness*, holds that the building type can be defined explicitly, and that architects can be instructed to create instances of the type (generally associated with Durand: see Perez-Gomez 1983:4, Vidler 1977a, 1977b, and Westfall and van Pelt 1991:146-148). The explicitness approach clarifies the creation of instances of the type by identifying procedures and principles of instantiation.

The *idealistic* view of type maintains that there is some abstract entity as a perfect ideal ‘building type object’ of which the actual buildings are imperfect examples (de Quincy 1825; Mitchell 1990:86-94). Work in this direction aims to create comprehensive very general structures that describe the building type – such as a structured set of variables and their relationships (Gero 1990, Coyne et al. 1990, Oxman 1990, and Rosenman and Gero 1993). Instances are created by assigning values to the variables, while maintaining the relationships. The focus on such abstract structures tends to downplay the importance of the design process.

Based on the work described above on drawings, knowledge, and methods, we propose a *procedural* approach. In the procedural approach, we pose that a building type not only contains knowledge about its parts, organisation, function, etc. (so-called *declarative knowledge*), but also knowledge about the steps that the design process should follow in order to produce a building design that belongs to the type (so-called *procedural knowledge*). Different designs are created by variation in the process (order of steps) and variation in each step (related design decisions per step).

6.1. Knowledge of the office building type and generic representations

In Table 2, a sequence of generic representations is presented. In order to demonstrate the knowledge captured in a generic representation, we consider the office building type, and look in more detail at the first generic representation of the sequence: *simple contour*.

In the generic representation *simple contour*, the building envelope is established. Since the building envelope is closely related to the floor area, the following parameters are defined: `number_of_floors`, `surface_area`, and `story_height`. In order to make decisions about surface_area, it is necessary to know about aspects that influence floor area in office buildings: minimum feasible surface area, gross area/net area ratios, and installations ratios. The `surface_area` is related to the `number_of_floors`, which is influenced by definitions of low-, medium, and high-rise office buildings, `story_height`, and structural principles.

Suppose there is the following very simple brief for an office building: a low- to medium-rise standard office building, for anonymous future tenants. The site is a rectangular area measuring 75x75 m^2, oriented north-south. Nearby buildings pose no special circumstances with respect to obstruction, shading, distance from site boundaries, etc. The rentable floor area has to be 5500 m^2. In order to draw an office building outline according to *simple contour*, it is necessary to make decisions about floor surface area and the number of floors.

In order to establish the surface area of a floor, it is necessary to know what are feasible office floor areas, the gross/net area ratio, the boiler and chiller required space and the air handling required space. This knowledge is related to the office building type. This can mean for example in this case that the minimally feasible rentable floor area is 600 m^2, gross/net area ration is 1.35, the boiler and chiller take up 2% of the building floor area, and the air handling takes up 4% of the building floor area.

Where does this knowledge come from? It is the common knowledge related the group of low- to medium-rise standard office building, or in other words, the office building type. This knowledge applies when the brief, client, or architect are not considering other options that will overrule the standard knowledge. Notice that this knowledge is not indefinitely valid: changes in building techniques, norms, social demands, or office work patterns change, then the knowledge will also change.

In order to establish the number of floors, it is necessary to know the height of low- to medium-rise office buildings (top floor below 22 meters above the site), floor heights of office buildings (3.00 m, 3.10 m, 3.40 m, 3.70 m, or 4.20 m), safety regulations related to building height (from 18.30 m height additional fire-fighting stairs are required), and structural systems related to building height and number of floors (different structural systems for 1-4 floors, 5-7 floors, and 8-10 floors).

Within the context of the simple brief, drawing a *simple contour* for an office building, implies that decisions have to be taken about number of floors, surface area, and storey height. In order to take these decisions, knowledge is required – in this case knowledge of the office building type. This example demonstrates very clearly that drawing implies decision-making which is very closely connected to knowledge of the building type.

7. Synthesis: generic representations, design methods, and knowledge

With the work described so far on generic representations, design methods, and building types, we derive the following synthesis. Knowledge of a building type is contained in a series of generic representations. The series of generic representations defines procedural knowledge. Each generic representation by itself defines declarative knowledge. Whereas other computational approaches stress the importance of an abstract data structure, such as a frame or prototype, the emphasis in this work lies on the well-structured drawing that has a particular meaning to the architect: the generic representation.

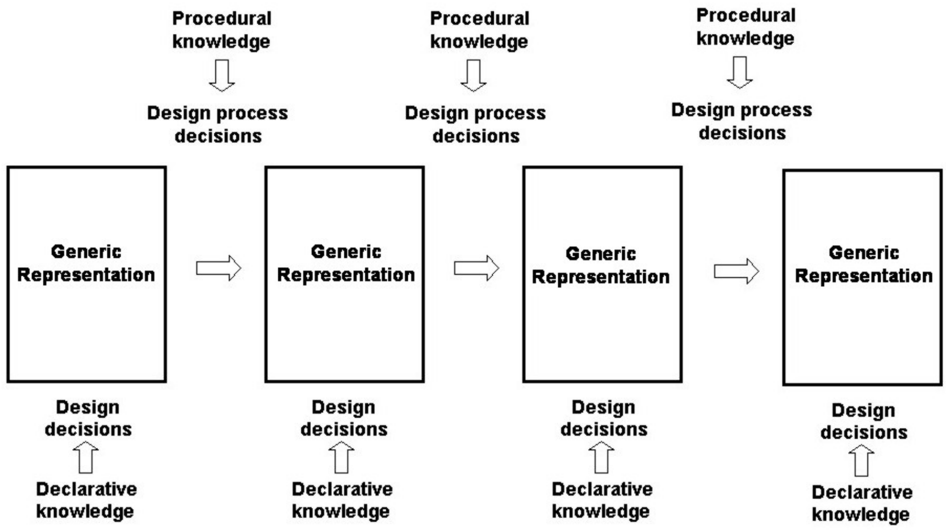

Figure 5. The relationship between generic representations and declarative and procedural knowledge.

We propose that the question of drawing, decision-making, and knowledge is not restricted to building types, but applies to all design problems when an architect designs by means of well-structured drawings as is the case with generic representations. This has the implication that drawings can become the key for computer support, *if only* the underlying graphic units can be utilised in some way.

The utilisation of underlying graphic units in drawings can take two different forms. One way is to recognise graphic units in drawings. This amounts to putting graphic units “in the paper,” so to speak or by making paper more intelligent. We call this strategy therefore *Paper plus*. The other way is to develop drawing tools that are based on graphic units. This is like putting graphic units "in the pen," or by making the pen more intelligent. This strategy consequently is called *Pen plus*.

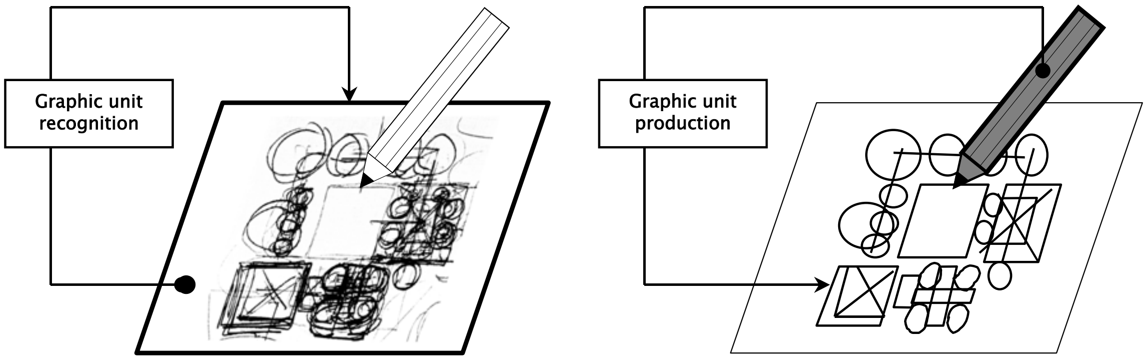

Figure 6. A: Paper plus approach. B: Pen plus approach.

Even with the availability of increasingly powerful and easy to use three-dimensional modelling tools, drawing and sketching will remain for a long while an important design technique. Because there is still no reliably precise direct three-dimensional interaction possible with three-dimensional models (other than by two-dimensional manipulation of mouse and pen), the user interface between architect and program will remain two-dimensional for the time being. *Paper plus* recognition of graphic units therefore, allows the architect to focus on the drawing and sketching, while the computer is left with the task to interpret the work. Apart from this, another motivation for graphic unit recognition in drawings is that automated recognition is a proof that graphic units can be described in such a consistent manner that a computer is able to identify them.

Considering the *Pen plus* approach, graphic units offer many angles to implement intelligent tools that help the architect. Such tools have the substantial advantage that from the very beginning the drawing is built up by well-defined objects. Therefore the internal representation of the design is well-structured, consistent, and accessible for machine reasoning. Disadvantages of this approach are that the architect is required to learn new tools, and that any drawing style that falls outside the scope of the tools cannot be supported, or in all cases, not interpreted by the system. User interface issues and handling become important issues so that the architect is not distracted from the work.

8. Paper plus: recognition of graphic units in drawings

Automated recognition of drawings relies on an understanding how drawings are constructed in general, and on the domain in which the drawings are made. Most recognition systems assume that a particular convention of depiction is used (such as plan, isometric projection, perspective, etc.) and that the drawings deal with a particular domain (such as interior architecture, architecture, mechanical engineering, and so forth). Under these assumptions, and usually within a well-defined area of application, such systems operate reasonably well. Within the framework of graphic units and generic representations, it is possible to say that these systems usually recognise up to five or six different graphic units. Since there are 24 graphic units, it is not difficult to imagine that the scope of such systems can be expanded considerably.

Any drawing recognition system will have basically the same setup (Figure 7): (a) drawing area; (b) drawing pen; (c) module for tracking and segmenting strokes made by the pen; (d) multi-agent module for determining which graphic units are present in the drawing; (e) visual feedback of results from d-module (Achten 2005).

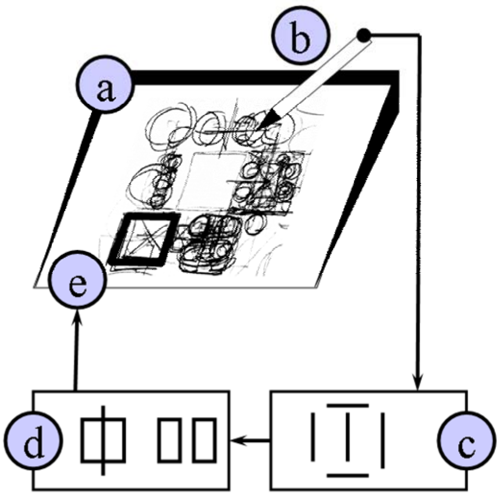

Figure 7. General structure of a drawing recognition system.

We propose to use multi-agent technology and online recognition in the system. The multi-agent approach acknowledges explicitly the limited capabilities per classifier; and that classifiers (in the guise of agents) should communicate with other classifiers to settle ambiguities. The parallelism inherent in multi-agent systems, in particular when multiple interpretations are possible, supports a weighed and balanced exchange of viewpoints. Within the context of graphic units, we define each agent in the system as specialized for recognition of one particular graphic unit.

Online recognition means that computer interpretation takes place while the architect is drawing. This approach is used often in handwriting recognition because of the high efficiency of the stroke direction feature (Liu et al., 2003). Similarly, we suppose that the stroke direction feature is also an important feature for sketch drawing recognition.

Graphic unit recognizing agents therefore, reason on the basis of strokes (modules c and d in Figure 7). A stroke is described in a number of absolute and relative features (Achten 2006). Absolute features are characteristics of the stroke itself, such as length, direction, and curvature. Relative features are defined with respect to earlier strokes, such as parallelism, alignment, and drawn consecutively. An agent that searches for *simple contours* for example, tries to find absolute features FA_1 (straight line) and FA_2 (circle). Additionally, it is looking whether the following relative features apply: FR_1 (closed), FR_2 (equal length), FR_{13} (medium to whole), FR_{14} (large to whole), FR_{17} (end-to-end connection). Each complete graphic unit is defined as a unique set of absolute and relative features.

There are only three absolute features:

- FA_1 (straight line): A segment that is straight within a margin area.
- FA_2 (circle): A segment that forms a closed circle within a margin area.
- FA_3 (curve): A segment that is curved in any way other than circle or straight line.

The following are the relative features. The notation {F} can be read as "the elements in set F":

- FR_1 (closed): {F} form a closed shape.
- FR_2 (equal length): {F} are of roughly the same length.
- FR_3 (one to many): {F} have one dissimilar element.
- FR_4 (equal distance): {F} have roughly similar distance between elements.
- FR_5 (line spaced): {F} have elements located on a line.
- FR_6 (few to many): {F} are in two unequal sized groups of roughly similar elements.
- FR_7 (zigzag): {F} have end-to-end connected lines that form a zig-zag pattern.
- FR_8 (closed sub): {F} contain multiple small closed elements.
- FR_9 (inside closed): {F} form a zigzag pattern which fills an outer contour.
- FR_{10} (multiple closed): {F} contain multiple medium closed elements.
- FR_{11} (letterset): {F} are specific to letter shapes. This is a shorthand way to isolate specialised letter recognition.
- FR_{12} (numberset): {F} are specific to number shapes. This is a shorthand way to isolate specialsed number recognition.
- FR_{13} (medium to whole): {F} are roughly half the size of the bounding area.
- FR_{14} (large to whole): {F} are roughly equal the size of the bounding area.
- FR_{15} (small to whole): {F} are roughly 1/10th the size of the bounding area.
- FR_{16} (isolation): {F} are distant from other elements. Distant means more than one times the length of an element in a particular direction to the next element in that direction.
- FR_{17} (end-to-end connection): {F} connect to each other by their ends.
- FR_{18} (arrowhead): {F} have arrowhead shaped lines.
- FR_{19} (perpendicular): {F} are oriented in straight angles to each other.
- FR_{20} (open): {F} do not form a closed shape.
- FR_{21} (parallel): {F} are roughly in parallel with each other.
- FR_{22} (similar direction): {F} are created in the same direction (begin- end point).
- FR_{23} (dashed line): {F} are constructed of dashed lines.
- FR_{24} (many to many): {F} are in two roughly equal sized groups.
- FR_{25} (displacement): {F} are in two roughly equal groups, with a similar displacement x,y vector.

In the work up to now, we have defined 3 absolute features and 25 relative features. This corroborates a general intuition that the basic strokes in drawings are fairly straightforward (line, circle, or curve), and that recognition mostly depends on relationships between strokes.

9. Pen plus: graphic units as drawing tools

Graphic units can also form the basis to create drawings tools. “Structural Sketcher” is tool based on graphic units and generic representations. It is a PhD-project by Slava Pranovich (Pranovich 2004). The system has tools for making graphic units and also provides a way to define and maintain changes in the relations between graphic units.

Consider the graphic unit *grid*. Drawing systems offer grids to position elements more easily, but only as a tool and not as a meaningful element on its own. There is however a strong link between a grid and the objects that coordinated on this grid (Figure 8). Changing the objects or the grid should have consequences for either the grid or the objects that relate to the grid. Structural Sketcher works with graphic units and also maintains the relations between graphic units, such as between contours and grids.

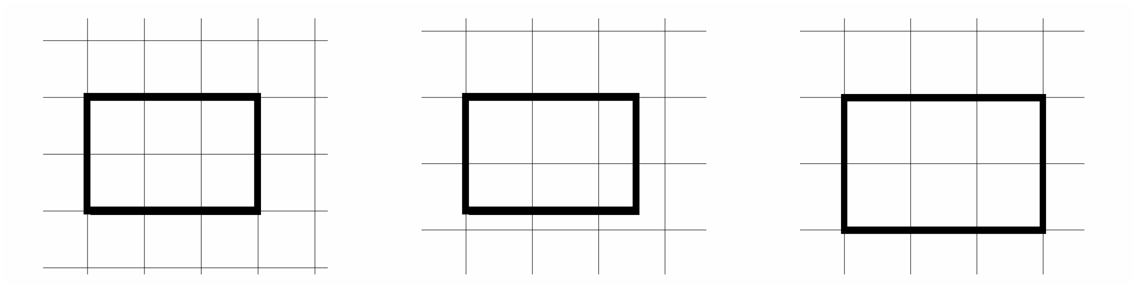

Figure 8. A: An object in a grid. B: The module size of the grid is increased. C: The object follows the new grid module.

In the final version of Structural Sketcher, 14 graphic units are implemented. For the creation of shapes, these are: *simple contour*, *contour*, *specified form*, *elaborated structural contour*, and *complementary contours*. For grids these are: *grid*, *tartan grid*, *structural tartan grid*, and *refinement grid*. For specialised elements these are: *element vocabulary*, *structural element vocabulary*, and *combinatorial element vocabulary*. Furthermore, the graphic units *zone* and *axial system* are implemented.

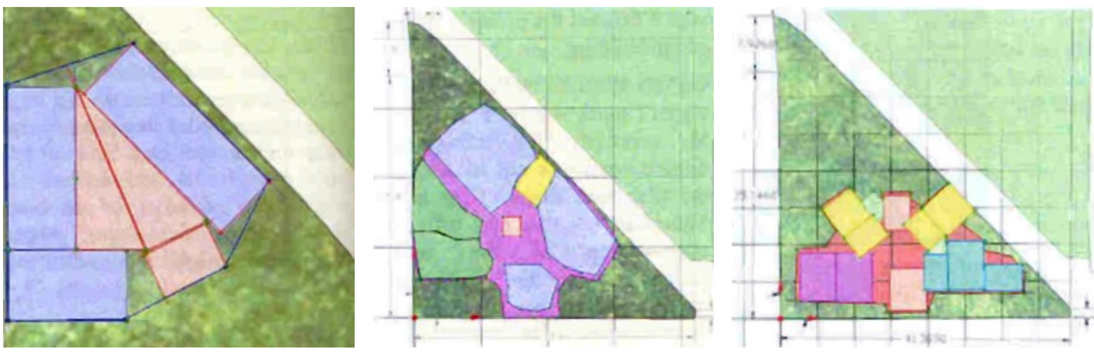

Figure 9. Three designs created with Structural Sketcher.

With Structural Sketcher contours, grids, elements, zones, and axial systems can be created, as well as the relationships between them. Relations can be either uni-directional (one graphic unit supervises another) or bi-directional (two graphic units influence each other). A relation between two graphic units is presented graphically as an arrow between two points, which belong to related graphic units. Multiple connection points can be defined per graphic unit. The user can set attributes for each graphic unit and relation to define which transformations in the graph have to be performed or not, such that a large variety of geometric relations is available to the user.

The first instance of the Structural Sketcher system offered total freedom in the relations between graphic units that are present. From the usage of the system both with researchers and architects, it appeared that so much freedom was not productive since the user had to maintain all meaningful relations. This took away a lot of time just for managing the system, which was very distracting. Based on this finding, we looked more carefully at the relationships between graphic units that hold for all generic representations, and implemented these in the last version of Structural Sketcher.

The user interface is critical to an unobtrusive use of the system. It is not that difficult to offer many options and much flexibility, but a major challenge is to define interaction techniques and visual metaphors such that the user can define what he wants in an intuitive way. In the last version of Structural Sketcher, many options that apparently were not used much were removed, and concentrated on the instantiation of graphic units and their manipulation, rather than control of relationships.

What Structural Sketcher clearly demonstrates, is that there is a large amount of internal relationships in drawings. To a trained observer, such as an architect, these relations are obvious when he or she is looking at the drawing itself. Nevertheless, it is far from obvious how to actually encode it. It is still a largely unsolved question how such relations can be automatically captured by a computer tool.

10. Conclusion

The design drawing is a carrier of knowledge. It is an important graphic tool by which architects develop designs. In this research work, we have shown that design drawings are structured with graphic units, that these graphic units are closely related to knowledge, for example taken from building types. We pose that the “graphic reasoning” of the architect is possible because making a drawing means is closely connected to making design decisions and using knowledge about buildings and the design problem.

Based on the centrality of the design drawing, we have shown two main kinds of application of this knowledge: a framework for recognition of graphic units in drawings, and a sketch tool called Structural Sketcher for drawing with graphic units. Future work will focus mainly on further development of automated graphic unit recognition in drawings. This gives computer tools the opportunity to understand design drawings while the architect is drawing – they may then assist in reasoning about the design, providing relevant knowledge, and automate analysis tasks. In short, through understanding drawings, the computer will better understand the design intentions of the architect.

11. References

Achten, H.H. (1997). *Generic Representations - An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design*. Ph.D. Thesis, Eindhoven: Eindhoven University of Technology.

Achten, H.H. (2004). *On the Computation of Well-Structured Graphic Representations in Architectural Design*. Bouwstenen 88. Eindhoven: TU/e.

Achten, H.H. (2005). Resolving Some Ambiguities in Real-Time Design Drawing Recognition by Means of a Decision Tree for Agents. In Martens and Brown (eds.) *Computer Aided Architectural Design Futures 2005*. Dordrecht: Springer.

Achten, H.H. (2006). Feature Clusters for Online Recognition of Graphic Units in Drawings. Bourdakis and Charitos (eds.) *Communicating Spaces*. Volos: University of Thessaly.

Argan, G.C. (1963). On the Typology of Architecture. Papadakis and Watson (eds.). *New Classicism*. Den Haag: SDU Publishers, 1990.

Boekholt, J.Th., Thijssen, A.P. and Dinjens, P.J.M. et. al. (1974). *Denken in Varianten: Het Methodisch Ontwerpen van Dragers*. Alphen aan de Rijn: Samson.

Colquhoun, A. (1981). *Essays in Architectural Criticism: Modern Architecture and Historical Change*. Cambridge, MA: The MIT Press.

Coyne, R. and Rosenman, M. and Radford, A. and Balachandran, M. and Gero, J. (1990). *Knowledge-Based Design Systems*. Reading: Addison-Wesley Publishing Company.

Gero, J.S. (1990). Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine* **1990**(winter): 26-36.

Habraken, N.J. (1985). *The Appearance of the Form*. Cambridge, MA: Awater Press.

Lawson, B. (1990). *How Designers Think – The Design Process Demystified*. London: Butterworth Architecture.

Liu, C.L., Nakashima, K., Sako, H. and Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition* **36**(10): 2271-2285.

Mitchell, W.J. and McCullough, M. (1991). *Digital Design Media*. New York: Van Nostrand Reinhold.

Oxman, R. (1990). Architectural Knowledge Structures as "Design Shells". McCullough, Mitchell and Purcell (eds.) *The Electronic Design Studio*, Cambridge: The MIT Press.

Perez-Gomez, A. (1983). *Architecture and the Crisis of Modern Science*. Cambridge, MA: The MIT Press.

Peters, P. (1973). *Entwurf und Planung – Verwaltungsbauten.* München: Callwey.

Pranovich, S. (2004). *Structural Sketcher: A Tool for Supporting Architects in Early Design*. Ph.D. Thesis, Eindhoven: Eindhoven University of Technology.

Quincy, Q. de (1825). Type. Translated by Anthony Vidler. *Oppositions* **1977**(8): 147-150.

Ronner, H., Jhaveri, S. and Vasella, A. (1977). *Louis Kahn – Complete work 1935-74*. Basel und Stuttgart: Birkhäuser Verlag.

Rosenman, M.A. and Gero, J.S. (1993). Creativity in Design Using a Design Prototype Approach. Gero, J.S. and Maher, M.L. (eds.). *Modeling Creativity and Knowledge-Based Creative Design*, Hillsdale: Lawrence Erlbaum Associates.

Rossi, A. (1982). *The Architecture of the City*. Cambridge, MA: The MIT Press.

Schön, D. (1983). *The Reflective Practitioner – How Professionals Think in Action*. London: Temple Smith.

Vidler, A. (1977a). The Idea of Type. *Oppositions* **1977**(8): 93-115.

Vidler, A. (1977b). The Production of Types. *Oppositions* **1977**(8): 93-115.

Westfall, C.W. and Pelt, R.J. van (1991). *Architectural Principles in the Age of Historicism*. New Haven/London: Yale University Press.

12. CV Henri Achten

Born 9 January 1967, Venlo, the Netherlands.

Study
BSc. and MSc. degree in Architecture. Eindhoven University of Technology (TU Eindhoven), Faculty of Architecture, Building and Planning (1986-1992).
PhD degree *Generic Representations: An Approach for Modelling Procedural and Declarative Knowledge of Building Types* (1993-1997: TU Eindhoven).

Work
Post-doc researcher at TU Eindhoven, Faculty of Architecture, Building and Planning *Virtual Reality – Design Information System* (1998-2000).
Assistant professor at TU Eindhoven, Faculty of Architecture, Building and Planning *Design theory and CAAD* (2000-2005).
Assistant professor at Czech Technical University in Prague (CTU Prague), Faculty of Architecture: *Computer aided architectural design* (2005-).

Academic activities
Co-promotor and advisor for 2 PhD candidates (TU Eindhoven).
Assessment committee 1 PhD candidate (Aalborg University).
Reviewer for 6 academic journals and 39 conferences.
Scientific committee member of 9 conferences.
Author of 6 books, 10 reviewed academic research journal articles, 11 articles in conference proceedings as books, and 34 conference papers.
Co-researcher in 2 European research projects, and in 1 GAČR research project.
Organiser of 5 conferences.

Teaching activities TU Eindhoven
Supervisor 26 MSc. graduation projects.
Teacher in design studio (1996-2005).
Teacher in 7 courses (1995-2007) in BSc. and MSc. programme.
Teacher in 2 courses (2005-) in ADMS programme.

Teaching activities CTU Prague
Assistant in design studios Prof.Ing.Arch. Fanta CSc., Prof.Ing.Arch. Pospíšil CSc., Ing.Arch. Florián PhD., Ing.Arch. Pata (2002-2007).
Teacher in CAD 2 for international students and CAD II, III, IV (2005-2007).
Tutor in 5 workshops of series *Digitální architekt*.

Academic functions
President of the *Association for Education and Research in Computer Aided Architectural Design in Europe* (2005-2007), Vice President 2001-2005.
Member editorial board of *IJDC*, *IJAC*, and *Computer Graphics & Geometry*.