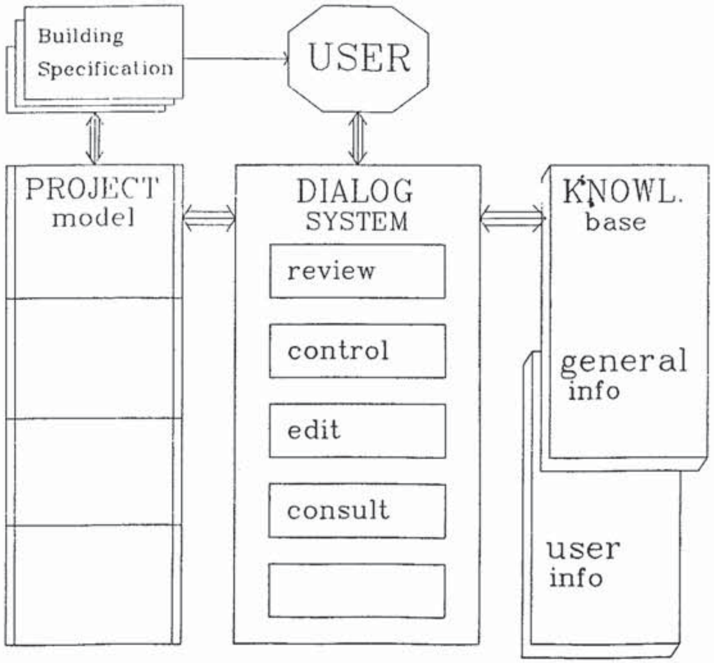


fig. 1: DARC-model

	general know base			user know. base		
	irr.	proj	user	irr.	proj	user
1st level						
2nd level						
3rd level						

fig.2: General and User dependent
Knowledge base structure

Automated Architectural Detailing: A Knowledge-Based Approach

Antony D. Radford

Architectural Computing Unit
Department of Architectural Science
The University of Sydney
NSW 2006 Australia

John R. Mitchell

Mitchell Walker Wright
7 Ridge Street
North Sydney NSW 2060 Australia

KEYWORDS

Working Details, Expert Systems, Knowledge Engineering.

ABSTRACT

Working details are the means by which an architect describes how the parts of a building are to be fashioned and assembled, and they are therefore central to the design and construction process. Their purpose is to ensure both functional sufficiency and aesthetic standards. The automated or semi-automated production of details within or linked to computer-aided drafting systems can lead to improvements in quality and consistency as well as productivity, and systems are available to facilitate detailing for steelwork, reinforced concrete, pipework, system building and other aspects of construction. Where the context is well defined and there are few appropriate solutions, automated detailing can be reduced to the selection of a correct procedure or pattern from a library of possibilities. Where there are many feasible solutions and the scope of the detail is large, automated detailing systems must make use of knowledge about the way parts of a detail can be assembled, what is good practice, and how a particular form of solution fits within a desired style or grammar of design.

This paper describes a knowledge-based approach to automated detailing using production rules as design generators. It is argued that a design model which makes explicit the assumptions and limitations of the knowledge it contains is necessary in the development of practically useful detailing systems in architecture. An extensive example and its prototypical implementation is described.

Production Automatique de Plans de Details: Une Approche qui s'Appuie sur l'Utilisation de Bases de Connaissances

Antony D. Radford

Architectural Computing Unit
Department of Architectural Science
The University of Sydney
NSW 2006 Australia

John R. Mitchell

Mitchell Walker Wright
7 Ridge Street
North Sydney NSW 2060 Australia

MOTS-CLES

Plans de Details, Bases de Connaissances, Systemes Experts.

SOMMAIRE

La production de plans de details est le moyen par lequel l'architecte decrit precisement comment les differentes parties du batiment doivent etre fabriquees et assemblees. Ces plans revetent donc une importance primordiale dans le processus de conception et de construction, leur fonction est d'assurer a la fois des criteres fonctionnels et esthetiques. La production automatique ou semi automatique de plans de details, integrees ou associees a un systeme de DAO (dessin assiste par ordinateur), peut apporter des ameliorations considerables tant en qualite et en coherence qu'en productivite.

Lorsque le contexte est bien defini et qu'il y a peu de solutions possibles, la production automatique de plans de details peut etre reduite au choix de procedure ou de forme appropriee dans une bibliotheque de methodes possibles. Lorsque l'ensemble des solutions possibles est large et que les plans de details comportent beaucoup l'elements, les systemes de production automatique doivent faire usage de connaissances ayant trait a la façon dont les parties des elements peuvent etre assemblees, au "savoir faire" du specialiste et a l'adequation d'une forme de solution particuliere a une grammaire ou a un style de conception.

Ce papier decrit une approche qui s'appuie sur l'utilisation de bases de connaissances pour la production automatique de plans de details et qui utilise des regles de production pour generer des solutions. Nous mettons en avant le fait qu'un modele de conception qui degage explicitement les bases et les limites des connaissances qu'il contient est necessaire pour que ces systemes soient utiles et performants en architecture. Nous developpons un exemple et decrivons sont implementation prototypique.

THE WORKING DETAIL

The working detail in architecture is the means by which an architect describes to a builder how the parts of a building are to be fashioned and assembled. Their purpose is to ensure both functional sufficiency and aesthetic consistency. As descriptive drawings, working details depict facts: the sizes of members; the topological and geometrical relationships between components; and (via notes) the materials and finishes to be used. The knowledge which is employed in establishing these facts comes from the designer, and is rarely explicated in the drawing. Many architectural practices maintain sets of standard details to cover recurring situations, and it is possible to buy books of "good practice" standard details, published by some authoritative organisation or individual. Because they deal only with facts and with final designs, standard details contain no intrinsic basis for their modification, so that a decision on what may or may not be varied rests entirely with the user of the detail. The knowledge is embodied in the drawing and is not retrievable. It is, indeed, implicitly assumed that an architect or draftsperson altering the detail has access to the same body of information that was available to the original designer. This may not be true; a reason for using standard details is not only to increase productivity but also to overcome a lack of such knowledge. An ideal detailing system should be founded not on standard solutions but on providing the knowledge for synthesising particular solutions. The knowledge should be contained as one description of the solution and should be retrievable.

KNOWLEDGE IN CAD DETAILING SYSTEMS

Computer-aided drafting systems originated as a tool to increase productivity by replacing a traditional pen and paper drafting medium. In the early systems all knowledge about the artifact being drawn remained with the draftsperson and the computer system dealt only with lines. Geometrical modeling systems recognised the need to represent the artifact as an entity with particular characteristics. Going further, a few drawing systems will automate procedures like the laying out of timber studs and noggins in a stud partition wall according to pre-set rules about stud spacing and the handling of junctions and openings. There are also systems available for the semi-automatic layout of concrete reinforcement and similar tasks.

Most of these schemata employ fixed data structures which make use of some implicit knowledge about the geometrical form of objects within the domain of application of the system. The knowledge is embodied in the computer code which implements procedures, either internally within the system (and therefore fixed by the system designer and programmer) or externally in a macro language associated with the system. Macro languages allow users to incorporate some of their own specialised knowledge about the way in which objects are created and represented, but only as ad-hoc extensions to a fixed data structure and computer code. For example, a stair generation module[1] will generate macro language commands for a computer drafting/modeling system to produce both graphic and nongraphic information for stair cores. Given answers to questions such as floor to floor height, door locations, quality and cost parameters, materials and the Federal, State or Local building codes which are to be applied, the system will generate small and large scale plans, typical sections, specifications and details for critical joints drawn from libraries of standard details and templates. The use of macro languages, though, is based on developing procedures for automatically drawing details rather than representing knowledge about how those details should be.

A KNOWLEDGE-BASED APPROACH TO DETAILING

The approach we shall describe is based on production rules in a generative expert system. The aim of such a system as a synthesis tool is to allow designers to work at a high level of design descriptions, subsuming lower level synthesis in the generation of such designs. The higher level could be the complete or partial building or detail. Since designers currently do much of their work at high level states there is nothing fundamentally new in this notion. An architect designing a building typically produces first a sketch design, then an outline design, then a detailed design. The first two assume that the details which are implicit in the outline specification of the final design are capable of being generated later, meaning that either the architect or others have the knowledge to design those details. The significant notion here is that this assumed knowledge will lie within the computer design system, and the process of generation therefore automatic or semi-automatic. The specification of a given design state (i.e. that the whole building will exist with certain characteristics) will imply reference to the lower level design states and the knowledge necessary to reach those states.

The model assumes that the problem of deciding on the final state for the design can be decomposed into a series of linked subproblems, each of which concerns the design of an output state given an input state. The output state from one stage becomes the input state for the next stage. The rules for the transformation from input to output state are triggered if defined conditions exist in the input state, or in a wider "state of the world" within which the design is taking place, figure 1. Execution of the model depends on proceeding through the stages and finding a set of conditions in the current design state and the "state of the world" which match the conditions necessary to trigger a rule. The complete set of rules is called a shape grammar[2] or a design grammar[3]. Gips and Stiny[4] have discussed the uniform characterisation of production rule systems as grammars. If there is only one rule which can be fired at each stage, and hence only one possible final design, or if there is a preset mechanism for choosing between alternative applicable rules, then execution of the model is automatic. Control of execution is discussed further below, but a working example will be presented first.

EXAMPLE: AUTOMATED EAVES DETAILING

EAVES is a prototypical generative expert system which implements a grammar developed[5] for eaves details. In fact, the domain is more restricted than eaves details in general; the rules within the expert system will only generate eaves details to match Australian domestic construction around the years 1910-12. By developing a set of design rules (a grammar) within a coherent and established style of architecture, the aesthetic consistency of the language of designs within the grammar is ensured by the rules themselves. Any resulting detail will be aesthetically acceptable.

The generative knowledge is represented as a sequence of "condition and consequence" production rules, figure 2. Thus rule 1 of the grammar proposes that:

If there is a single leaf brick external wall and a pitched roof
then a wall plate shall terminate the wall
and
the wall plate shall be fixed to the wall by straps.

As rules are applied, the design passes through a series of intermediate states (first wall, then wall with wall plate, then wall with wall plate and rafter, and so on) until the complete design is generated. In a traditional detail only the shape is described (by a drawing), with appropriate annotations. Computers allow the maintenance of many different descriptions of the same design. A detail can be described symbolically (for the internal convenience of the computer), by two and/or three dimensional shape, by written specification, by cost, by a record of the course of its creation, by the knowledge implicit in the decisions taken during its creation, and any other appropriate form. Moreover, with computers it becomes very easy to switch between these different descriptions. Thus the state of the eaves detail after applying rule 1 might be specified by:

(1) the shape;
(2) the specification, i.e. "Fix 4 x 2 in (100 x 50 mm) sw wall plate to single leaf brick external wall using 1 x 1/16 in (25 x 1.5 mm) galvanised ms straps at 10 ft (3000 mm) centres tucked under minimum 6 courses of brickwork";
(3) the quantities, i.e. "X m of 4 x 3 in (100 x 75 mm) sw, Y No. 20 x 1 x 1/6 in (500 x 25 x 1.5 mm) galvanised ms straps", where X and Y are calculated from the length of external wall over which the detail applies;
(4) the cost, i.e. "Timber $A, straps $B, fixing $C, total $D", where A, B, C and D are calculated from facts about the costs of materials and labour in assembling materials.

and finally:

(5) the knowledge, i.e. "Wall plate needed for nail fixing of rafters. 4 x 2 in (100 x 75 mm) is traditional

size. 4 x 2 in would work but would not look right. Straps needed to hold down roof against wind uplift".

COMPONENTS OF THE EAVES EXPERT SYSTEM

The EAVES system consists of the following components, figure 3:

(1) An inference engine, implemented in the PROLOG language.
(2) A knowledge base, essentially the set of acceptable methods to construct (arrange) the elements of the roof and wall as they meet at the eaves. The rules are derived from good practice within the chosen style applicable to residential construction and expressed as PROLOG predicates.
(3) A graphics editor, providing all the graphical creation, display and manipulation facilities on a TEKTRONIX graphics workstation, written in FORTRAN77.
(4) A graphical database, storing graphical descriptions of the shape elements.
(5) A user interface, to interpret and control the users input by keyboard and/or tablet for the graphics editor, written in FORTRAN77.
(6) A graphics display, used to define and display graphical descriptions of elements, and to show the progressive and final applications of shape rules, written in FORTRAN77.

At this stage the rules in the knowledge base only contain information about arrangements and shape. Designs are generated by the serial firing of a list of rules, each conditional upon the state of the design as determined by the firing of previous rules in the sequence. A design is reached once all the rules have been tested. Part of the network of rules and the implicit links between them via conditions is shown in figure 4; the heavy lines show the dependencies between rules in the design of a particular eaves detail shown in figure 5. Only the knowledge base and the graphical database are peculiar to eaves detailing; the rest of the system can be applied to other details.

DISCUSSION

The implementation issues, as with any generative expert system based on production rules, concern the encoding of the rules within a computer system, the representation and recognition of design states and mechanisms for control and for tracing back from a complete design to the lower level states which it subsumes[6]. The simplest control mechanism is to ask the system user to accept or reject the consequences of each rule in turn, trimming the number of possibilities by a process of interactive selection. This is what EAVES now does; the acceptance or rejection can either be done during each use of the detailer or set up in advance so that the process is automatic. What we usually want to do, though, is to specify some salient points about the desired result and let the expert system sort out which valid rules will generate a design with those characteristics. For a detailing expert system we need to work backwards from characteristics of final design to the rules and conditions which allow those characteristics to exist (backward chaining), the next stage of development for EAVES.

Generative expert systems allow designers to concentrate their work on the new and different aspects of a design, rather than diverting attention to the re-solving of detailed design problems where there are already acceptable and well established solutions available. It provides a base level on which the designer can work. Their domains will for some time be small in scope and lie within well defined areas, either through limiting the application (to doors, or windows, or eaves, etc) and/or the variety (to a system building method, to a particular style, to a type of door, window, etc). To realise their potential as an aid to both productivity and quality in the design of building details there is a need for a solid theoretical basis for their development which will emphasise the accessibility of the knowledge being used, its availability to modification, and the acceptability of the result with respect to an architect's traditional concern with aesthetic as well as functional quality.

ACKNOWLEDGEMENTS

Part of the work reported in this paper is supported by the Australian Research Grants Scheme.

REFERENCES

1. Jung/Brannen, "Artificial Intelligence Based Expert Systems", in Pioneers of CAD in Architecture, edited by A.M. Kemper (Hurland/Swenson, Pacifica, 1985).

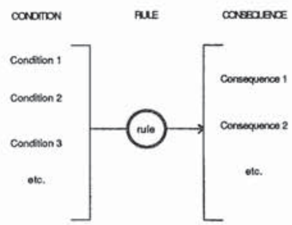
2. G. Stiny, "Introduction to Shape and Shape Grammars", Environment and Planning B, 7, pp.343-351 (1980).

3. J.S. Gero and R.D. Coyne, "Logic Programming as a Means of Representing Semantics in Design Languages", Environment and Planning B, 12, pp.351-369 (1985).

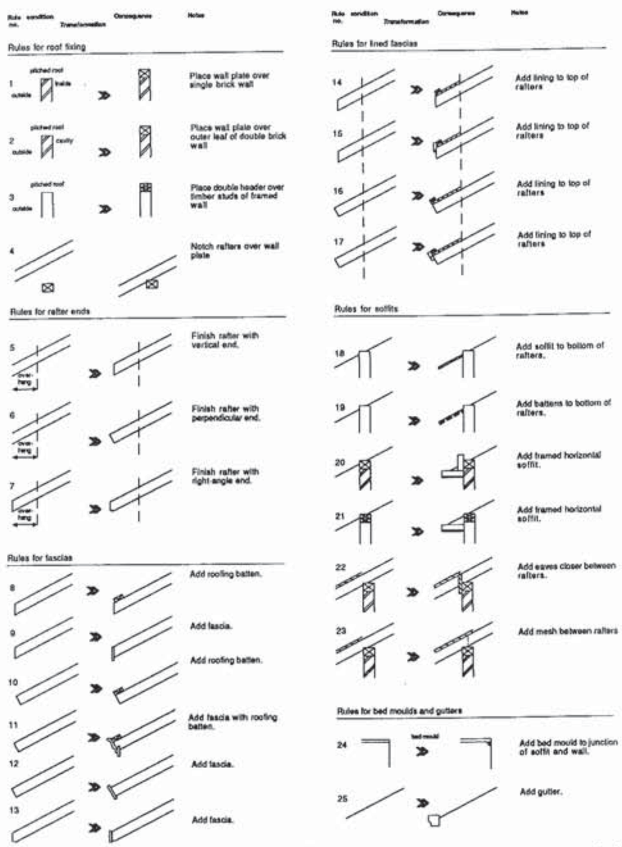
4. J. Gips and G. Stiny, "Production Systems and Grammars: A Uniform Characterization", Environment and Planning B, 7, pp.399-408 (1980).

5. A.D. Radford and J.S. Gero, "Towards Generative Expert Systems for Architectural Detailing", Computer-Aided Design, 17, pp.428-435 (1985).

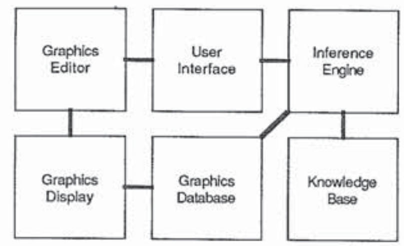
6. R.D. Coyne, "Knowledge-Based Planning Systems and Design: A Review", Architectural Science Review, 28, pp.95-103 (1985).



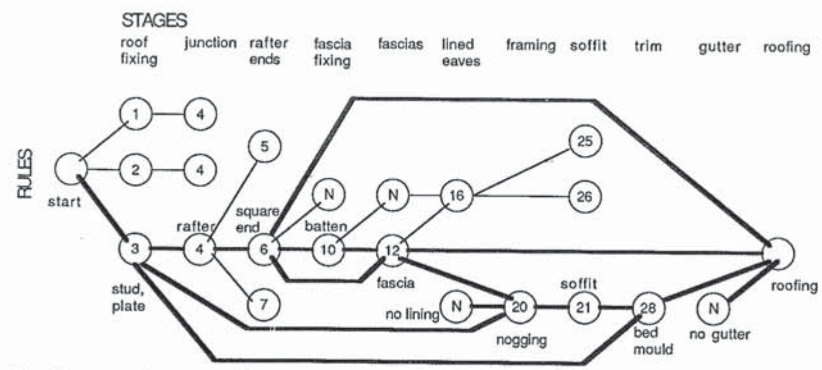
1. The operation of a transformation rule in a design grammar.



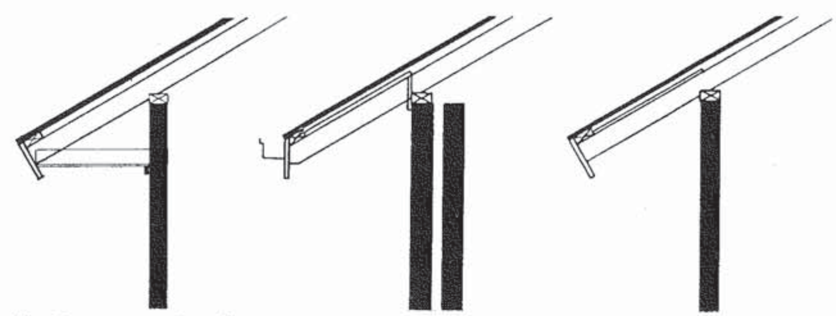
2. Shape transformation rules in a grammar for eaves details.



3. Components of the EAVES detailing system.



4. Some rules at different stages of the EAVES system and the links between them through conditions in the state of partial designs.



5. An eaves detail (left) resulting from the rules implemented in figure 4, and (centre and right) alternative designs implemented through the execution of different rules.