A Frame for Experimentation with Shape-grammars in Architectural Education

Michael Hellgardt
STUTS Architects
The Netherlands
Sourav Kundu
Kanazawa University
Japan

ATN's (Augmented Transition Networks) are an appropriate tool for the representation of architectural shape grammars. The application of this frame in stages of shape grammar codification is resumed. Problems of rule definition and of the connection to existing CAD software are raised. The paper is evaluated in the light of problems raised in the introduction and of what is understood in this respect by an Added Value Of Computer Aided Architectural Design.

A Frame for Experimentation with Shape-grammars in Architecural Education

1 Introduction

There is an enormous gap between the machinery available for the representation of design and computational tools available for the generation of design in its literal sense. The gap is threatening because it is part of a world governed by images which have become independant and selfsufficient - a phenomenon discussed in cultural criticism. But the gap is a challenge, too. Shape grammar can help to bridge that gap, and consequently is itself part of that challenge. The challenge is a double one: in terms of computer modelation, and in terms of an old unsettled account: the functioning of grammar in shape and space perception, in the built environment, and in architectural practice, thought and education.

In the literature on shape grammar we find grammars of particular architectural styles, or building types. There are also more fundamental investigations which, however, are relatively abstract and unrelated to such singular grammars. Both ways contain no answer to the gap between representation technology and computational tools that allow a more fundamental approach to architectural design. As a result, which we are not the first to raise (Coyne and Snodgrass, 1993), the impact of fundamental research in the fields of AID (Artificial Intelligence in Design) and shape-grammar on architectural practice and theory has been modest, if at all existing.

Though much of the work is done by architects, the new field of shape-grammar is dominated by standards of cognitive science and related problems and requirements of formal and computational modelation. One might argue that people defining models and writing programs should have more substantial knowledge of architecture. This is certainly not wrong, but not a structurally convincing solution. A structural solution would be to provide specific computational tools required for a computer aided investigation of shape grammar in architecture and spatial configuration. This investigation is primarily a matter of architects. At the other side we cannot expect architects to do the work of computer programmation. A task of scientists is to provide tools that enables the use of a computer without doing that. To make this more explicit in the light of the gap raised: obviously a tool is required which is not restricted to the manipulation of geometric entities, to Computer Aided *Drawing* in other words, but which allows the manipulation of grammatical categories and mental representations in a way that corresponds to assumed intellectual and artistic performances in the brain, associated with architectural design and perception. Only such a tool can make the promise of Computer Aided *Design* real, and only such a tool opens a professionally unbiased computer aided approach to the account of *grammar in architecture*.

The best model for the work to be done is presumably Natural Language Processing. In earlier work we have shown the applicability of a tool widely used in this field, an ATN (Augmented Transition Network, cf Winston, 1984, chapter 9), for the representation of architecture and architectural grammars (Kundu and Hellgardt, 1996). In the present paper this frame is presented and discussed in a shape of generalisation which opens an approach to the problem raised.

2 A shape grammar ATN

2.1 The empty ATN frame

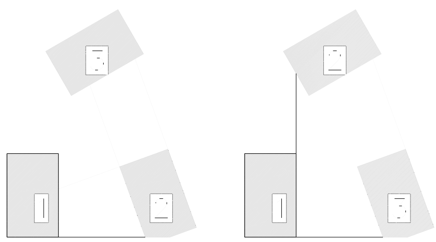
The nodes of nondeterministc networks, such as ATNs, are surrounded by *clouds* of possible choises. The notation of a node reflects that (figure 1). A node has one or more outgoing arcs which envoke functions to be evaluated. When a node is called in a network its first arc is evaluated and its remaining arcs are stored. These stored options are activated as long as no result from other arcs is given, or in case that more than one solution is desired, for instance for exhaustive search. In our application all arcs are of the form f(g(x)), where x is the register containing all information available at a given stage of network evaluation, g is the current and f the next function to be evaluated with the result of g(x). In terminal arcs the f, the next node is empty, or of the class of the network where a network is embedded in (discussed in section 2.3). To this arc notation a tree notation of the network corresponds. The branches, or arcs of this tree are bidirectional, which means that we climb up and down in it from category to instance, and vice versa.

```
(defnode node            (defnode cat
   (arc(1))                 (contin cat-next(1) <cat-rules>)
   ...                      ...
   (arc(n)))                                  (contin  cat-next(n)
<cat-rules>))
```

Figure 1, Basic ATN structure (empty ATN frame)

Nodes and arcs are defined as macro functions. Obviously all next functions in the arcs must be encoded as nodes. For the current functions that is less obvious. Corresponding to grammatical structures to be discussed below, rules are addressed as rule categories by the next-nodes (f), and as instances of these categories by the current functions (g). These rule instances are clustered to rule-sets (marked by <...>; brackets, 'defnode' and 'continue' are omitted in figure 2 and 6). They can be encoded as ordinary functions.

2.2 An empty space-between grammar ATN



In the context of machine-design a grammar can perhaps work as a machine designing machines. With respect to cultural, verbal or non-verbal, expressions this is less obvious. An ATN applied in this context is a machine, but not a grammar, a machine that can produce shapes - in our case - with the aid of grammatical knowledge. A machine that can be programmed with grammars. In this section we will outline how an empty ATN frame can be filled, or programmed, with *one* of presumably various possible basic shape-grammar structures. In another contribution to this conference book (Hellgardt, Kundu and Klinkenbijl, 1997) the same frame is applied to quite different structures.

Figure 2

Figure 2 can be read as an ambigous expression. To the left from any added rectangle a new space-between (blank) results. To the right another order of sequence of rectangle allocation shows that more than one rectangle added can be required in order to result into an evaluable space-between. We resume that as follows: shape configurations can be read as a sequence of the allocation of solids in a way that voids, spaces-between between solids, result. New solids can be added creating new spaces-betweens, or making only virtually existing ones acceptable, and so on. Some experimentation with the work of the architect Scharoun has demonstrated that such a principle is relevant in professional architecture (Hellgardt, 1993 and 1994). A survey of non professional architecture (eg Caniggia, 1976) demonstrates that such structures are wide-spread. We can add the Amsterdam Canal House, the M'Zab building culture, array settlements in Japan and many other examples.

As raised in our articles mentioned the *space-between* appears in literature on architecture, in etymology and above all in philosophy (there rather in the shape of the distinction between spatium and extensio). Here we only discuss in general lines how a principle as raised with figure 1 can be translated into some basic principles of grammar and how an ATN frame can be programmed with that.

Resumed in terms of the general frame of figure 1, figure 2 underlies an extended network of the form:

```
space-between_generator       space-between
  extension space-between       cat-next(1) <space-between?>
  space-between extension       ...
                                cat-next(n) <space-between?>

cat-next
  cat-next(1) <cat-instance>  final-test
  ...                           () <final-test>
  cat-next(n) <cat-instance>
  final-test <cat-instance>
```

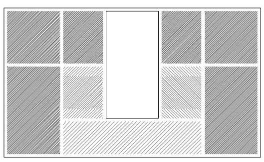
Figure 3, ATN applied for space-between grammar (empty space-between ATN frame)

When a space-between design starts there are two options (two arcs, figure 3, top left). The first option can be labelled *environmental design*: an environmental context is tested if some evaluable space-between is given, or can be elaborated in a way that a spatium design can proceed by possibly filling it and/or testing remaining 'rest'-patches. The result is passed to the next node, an extension. The second option is addressed if none of the available space-between elaboration rules returns a result. In this case the next option of the network is addressed which means that the design starts with an extension (arc 2 of figure 3, top left) creating a new space-between, to be addressed by the next node. If this new space-between is unacceptable the mechanism is repeated and another extensio is added, until an evaluable spatium is given which can be filled and/or tested. This option can be called *tabula rasa* design, design in an empty context. The space-between this kind of design generates is no environmental context any more but a design context.

With that not yet a shape grammar, but an ATN implementation of a basic grammatical assumption for shape configurations, the space-between assumtion, is outlined. Such a basic grammar frame can be filled with various grammars, the grammar underlying work of Scharoun for instance. Here we proceed with a global description of how it can be applied to model a phenomenon which is simpler in terms of computer modelation: Palladio Villa's. Grammar has to be elaborated to some minimum extent exhaustively at this stage.

2.3 An application

Main references are: the impact of the local, in particular Venetian building type (Ackerman, 1977) and Palladio's harmonic theory (Wittkower, 1973). A main feature of the Venetian palace is a central hall between mirrored wings. In terms of our approach the wings, unambigously recognizable in Palladio floor plans, are extensions with a space-between between them. The wings are *chords* consisting of harmonically shaped rooms. Such wing-chords can be relatively easy generated by means of a multiple Cartesian product. This is a top-down approach which do not enter, resulting in some hundreds of wing instances reflecting Palladio's way of harmonic dimensioning which, as the literature demonstrates, was a conscious one.



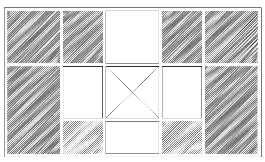


Figure 4

Figure 4 shows paraphrases of the Villa Badoero (left) and the Malcontenta, generated by a space-between ATN. The network starts with an array of solids (dark), called an *extension*, the mirrowing of which creates a void, a *space-between*. Such space-betweens can be articulated in various ways. In Palladio villas we find tunnel- or crossvault (Malcontenta) articulations. Calling rules, the ATN tentatively fills such space-betweens, in the formalism represented as pixel-maps, with patches of color values representing room categories such as loggia (light grey), staircase (darker), niche, passage and fill-block. The patches are defined, roughly, by possibly empty xy-parameters and their position in an interpretation of the space-between in terms of zones and regions.

Important is the sequence of rule application. The Badoero paraphrase underlies a loggia-staircase sequence which would lead to an unacceptable appearence in reversed form. In other wing-contexts only such reversals are successful. The search for such different sequences in not possible in deterministic networks. However, the case of an interpretation of architectural design given here underlies the assumption of a grammar combining deterministic and non-deterministic search. Non-deterministic is only the *filling* of the space between. The mechanism *creating* a space-between is, at least partially, deterministic. The result of such search procedures can be represented as a tree which is not the network evaluation tree mentioned in section 2.1, though it reports a decision paths. These trees are essentially the same as *phrase-markers* (also *Structural Descriptions*, SDs; Chomsky, 1965). As an result of grammatical performances, or their simulation through computation, they are equivalent to a phonetic or, in our case, graphic representation such as figure 4.

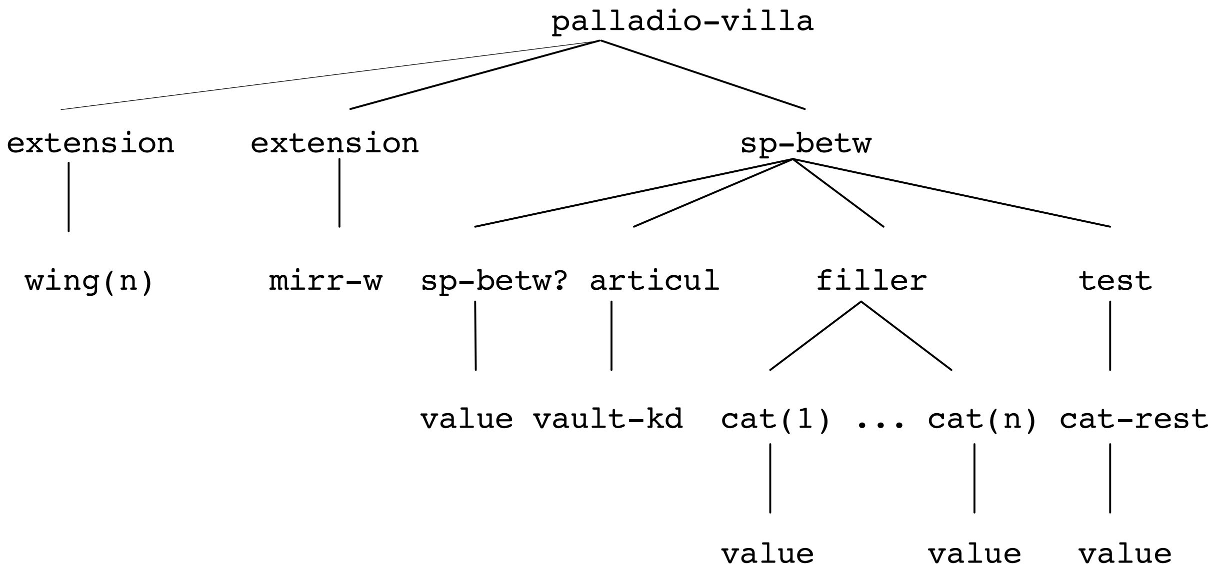


Figure 5

An important feature of generative expressions, in contradiction to stereotype ones, is that they are nested expressions, not in terms of objects, but in terms of recurring grammatical operations of a same type. In any room-category of the tree above another Palladio Villa can be searched for and inserted. More realistic is a staircase, for example, consisting of landing-extensions and a resulting space-between possibly usable for flights, as shown with different color/grey values in figure 4.

With some more properties, ignored here, this simple network comes up with all possible variants of non-bisymmetric villas of one 'pitch' (we do not enter details about that) given with Palladio's legacy. And it comes up with a series of similar cases. The state space of legal rule combinations, selected independantly from evaluation, is large. Filled with rule-sets with only a few rules it soon amounts 1000's of paths for a single of some 100s of wing-instances. We experiment with GA(Genetic Algorithm) directed search in such state spaces of architectural design.

The tree above shows the result of a path in the following application of the frame of figure 2:

```
palladio-villa                       space-between
  extension  <eval-context>            sp-betw_artl <space-between?>
  space-between <make-wing>

sp-betw_artl                         fill-cat
  fill-cat(1) <vault-articl>           fill-cat(1) <cat-rule>
  ...                                  ...
  fill-cat(n) <vault-articl>           fill-cat(n) <cat-rule>
                                       test_sp-betw <cat-rule>

test_sp-betw
  () <test-rule>
```

Figure 6, space-between ATN applied for Palladio villa's

This network demonstrates, though in a rudimentary form, that a grammar is defined not just by rules but the interweaving of different types of rules in terms of what we can call grammatical functions of shape configuration.

3 Rule definition tools and connection to existing software

The distinction between categories addressed by nodes and rule-sets associated allows that rules defined independantly form an ATN frame can be called within a given ATN frame. This enables the implementation of shape rules in a grammatical context as extensions of commercial CAD systems (cf Fawcett, 1990). All the user then has to do in order to 'program' an ATN frame with a grammar, is to enter names of rule-categories and rule-sets.

The added value of a shape grammar ATN, however, is still restricted if no associated rule definition system is available. To an interpretation of architecture shape grammars rule-types correspond, not only in terms of their grammatical function, but also in terms of some basic design principle generally associated with particular drawing acts. Rules that combine labelled shapes for instance (Stiny, 1980). In our approach another rule-type of this rule-species is addressed: a kind of patches-distribution or *brushstroke* technique which works with pixel-maps and reflects an assumed way of spatial orientation. If the functioning of such rule-types can be generalized, rule definition macros can be defined which write and encode rules on the basis of user-dialogs allowing graphic intervention on the monitor, if desired. A rule is called in an actual network evaluation of the kind discussed here with two arguments, the result of the previous operation and the evolving configuration, here a pixel-map. When a user defines a rule only the number of such arguments is specified, together with a series of arguments required for the automated encoding of a rule, such as the dimensioning and zone/region parameters emntioned above.

4 Conclusion: aspects of architectural education

Rule definition procedures of the pre-structured kind described can make use of SDs, which enables the user to combine architectural design and grammar design and which can be compared to what is called *language learning* (cf Chomsky). Language learning is a good model for design learning. Education, seen in the light of this paper, is a combination of the application of rule-types provided by some basic instruction, enabling rule learning supported by a grammar frame of the kind presented. Concluding, we will discuss problems of the kind of a *symbiosis* of human and machine learning involved with that.

In the introduction we pointed to the danger of a selfsufficient image-technology, and of an unadequate impact of requirements of computer modelation on research. Stiny's and Mitchells (1978) simulation of Palladio Villa's, the importance of which we do not ignore, can be seen in the light of the latter. The underlying method of concatenating room configurations in 'tartan' grids is only loosely related to Palladio's work and related literature - as has been partially remarked by the authors themselves. We try to find a more deliberate anthropomorphic combination of technical knowledge required for computer modelation and of knowledge in fields of application of such modelation, such as architecture and architectural history and theory. The intention of our demonstration was to give an example of how an at the outset empty technical frame, the ATN, can be programmed in different stages with such knowledge.

Particularly when it can be used by architects and students without specific computer programming knowledge, a tool such as an ATN, that allows the mechanic application of grammatical knowledge, can enrich design practice. From an intellectual and artistic point of view, however, such a tool not only enables and supports but also challenges design experimentation and above all design knowledge at a more fundamental level: grammar. Certainly, commercial CAD systems widely in use today are also a challenge in that sense and can be of impact on design learning and trigger new approaches to design. But these systems are drawing, not really design systems. They provide no means to make the kind of learning and experimentation involved with, or even required for their use explicit in terms of computer modelation to the benefit of their *updated* use in future

As any kind of technological innovation, any kind of computer modelation is an ambigous challenge to human learning. The powerful machinery of image production - *image-appeal*, in fact -available today can have a destructive impact on design learning. This effect is based on the confusion of the appearence, or image of something with the something itself. Intellectual curiosity is addressed on a surface level in this confusion. Both in practice as in education, computer supported shape-grammar investigation would address it at a deeper layered and more constructive level: the one of understanding.

Grammar in architecture, as raised in the introduction too, is relatively unexplored, certainly in the light of grammar knowledge in language theory, lingustics, semiology and natural language processing. This situation is a matter of architecture and has nothing to do with computers, but today it can adequately and seriously be approached *only* with computers and computerised tools. The ATN frame discussed here is aimed in that direction. What makes such a tool interesting for any kind of, such as architectural education is that the user is not addressed as an idiot by a black box of software the intellectual working of with is unknown and inaccessible to him. In contrast the use of the grammar design tool outlined here is not possible without some knowledge of grammatical structures the tool is based on - knowledge that could be the initial information in a first years course on architectural design.

This is a kind of *dead* knowledge. The use of the tool proposed, however, is not possible without some *living* ideas, intentions, obeservations in terms of grammatical principles underlying architectural design, or a case of it. To some extent the tool must be neutral. At the other side it must be taylored to certain intentions etc. Section 2 demonstrates, though only globally, the necessity and functioning of that kind of combination. We think it is exactly such a combination that would challenge both the teacher and the student to explore structures of grammar with the aid of computers, not only in order to make richer designs, but also in order to make his understanding and intention in terms of architectural design, understanding and perception - without or *with* the use of a computer - explicit.

References

Ackerman, J.S. (1977). Palladio, Penguin Books, Harmandsworth, Great Britain.

Caniggia, G. (1976). Strutture dello spazio antropico, uniedit editions, Firenze.

Chomsky, N. (1965). Aspects of the theory of syntax, the MIT Press.

Coyne, R. and Snodgrass, A. (1993). Rescuing CAD from rationalism, Design Studies 14.

Fawcett, W.H. (1990). Shape rules for architecture, paper presented at EuropIA conference, Liège, Belgium.

Kundu, S. and Hellgardt, M. (1996). A networks approach for representation and evolution of shape grammars, in J.S. Gero and F. Sudweeks (eds), Artificial Intelligence in Design '96, Cluwer Academic Publishers.

Hellgardt, M. (1993). Syntaktische Aspekte der Arbeit von Hans Scharoun, Begründungen und Erläuterungen zu den im Begleitprogramm der Ausstellung 'Werkschau Hans Scharoun' vorgeführten Fragmenten einer Scharoun-Syntax, Akademie der Künste, Abteilung Baukunst, Berlin.

Hellgardt, M. (1994). Dentro l'architettura di Scharoun, Housing 6, Etaslibri, Milano.

Hellgardt, M. and Kundu, S. (1997). Objective casualty - an experiment with recognition and emergence in design practice, AVOCAAD conference book.

Stiny, G. and Mitchell, W.J. (1978). The Palladian grammar, and: Counting Palladian Plans, Environment and Planning B 5, 5-15 and 189-98.

Stiny, G. (1980). Kindergarten grammars: designing with Froebel's building gifts, Environment and Planning B 7, 409- 62.

Winston, P. H. (1984). Artificial intelligence, Addison-Wesly Publishing Company, Reading, Massachusetts.

Wittkower, R. (1973). Architectural principles in the age of humanism, Academy Editions , London.