A Grid Model for the Design, Coordination and Dimensional Optimization in Architecture.

Daniel Léonard, Olivier Malcurat

A grid model for the design, coordination and dimensional optimization in architecture

D.Léonard[1] and O. Malcurat[2]

C.R.A.I. (Centre de Recherche en Architecture et Ingénierie)
School of Architecture
54000 Nancy, France

INTRODUCTION

Our article treats layout grids in architecture and their use by the architects for the purposes not only of design but also of dimensional coordination and optimization. It initially proposes to define an architectural grid model as well as a set of operations to construct them. Then, it discusses this model and its capacity to assist the designers in their everyday work of (re)dimensioning[3].

The architectural grid, as an instrument of design, is omnipresent in the work of architects whatever the times or the societies we may take into consideration. Today, it is generally the starting point of a number of projects (residences, offices, hospitals, etc.). As a geometrical artifact, its power of abstraction enables it to convey architectural elements such as spaces (an urban network for example), works (structural walls) or products (tiles)(Zeitoun 1977), (Pelliteri 1997).

This universality of the architectural grid is explained by two essential functions. Its quality of graphic reference allows the dimensioning and the positioning of architectural objects. These are its metrological and topological functions. As a principle of composition, the architectural grid allows the whole and the element to be reconciled and to integrate a finite set of dimensions into a single and hierarchical system. That is its coordination function. The model of architectural grid that we propose preserves these two principal functions.

CLASSIFICATION OF ARCHITECTURAL GRIDS

The multiplicity of the examples of architectural grids and the complexity of their modeling because of this multiplicity could lead us to a first classification (Figure 1). For instance, the set of the plane grids could be divided in four types: rectangular architectural grids (most current), triangular architectural grids, polar architectural grids and other architectural grids, those which do not belong to the three preceding categories. This way of specializing the architectural grids however requires the definition of a model for

[1] Professor at the Architecture School of Nancy
[2] Post graduate student
[3] We thank A. Mirgaux from the Henri Poincaré University for his help for the mathematical part of this work.

each type of architectural grid and only the architectural grids most usually used are seen equipped with a model and data-processing instrumentation.

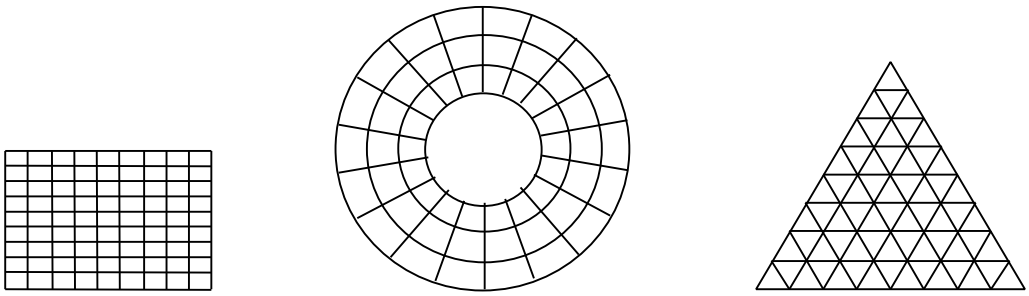

Figure 1: Classification of the set of plane grids by the shape.

Our approach is different and more topological (O. Malcurat 1997). The criterion of classification that we retain is not the form but the number of dimensions of architectural grid: we consider that any architectural grid consists of one or several basic architectural grids, that we qualify monodimensional grids (Figure 2). For example, the rectangular architectural grid is made of two orthogonal monodimensional grids. However, the monodimensional character does not imply the rectilinear character: the polar architectural grid for example is made of two monodimensional grids, one of which is concentric and the other radiant one.

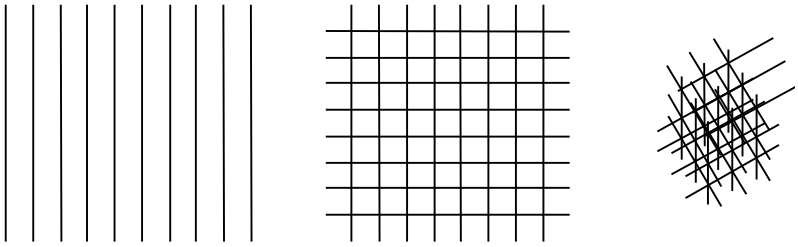

Figure 2: Classification of the set of grids according to the number of their monodimensional grid.

Rules and monodimensional grids

We introduce the concept of *rule* as being the dimensional structure of the monodimensional architectural grid. This concept enables us to separate what " is located " and what is not, which varies and what remains invariant at the time of an operation of displacement (translation) or change in orientation of the architectural grid. Thus, an architectural grid is composed of a rule and a situation (position and orientation).

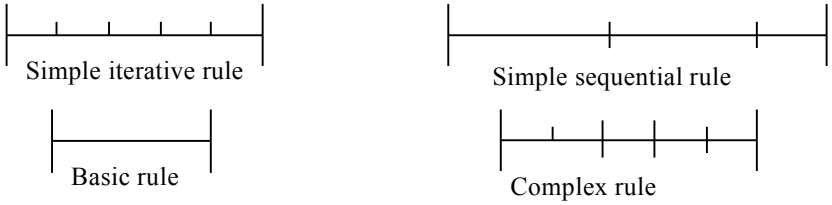

Figure 3: Examples of rules

Rule is a sequential succession of *spans* (Here « span » means space separating two axes from grid whatever its cartographic scale). Each span can itself be the support of a rule. Thus, a rule consist of several hierarchical levels of rules.

A DATA MODEL FOR THE RULES

Construction of the rules

We have identified four basic syntactic patterns to build a rule with other rules. Two of them are structural operations:

- the *sequence operation* which builds a sequential rule with at least two other rules,
- the *repetition operation* which builds an iterative rule with one rule and a factor of repetition.

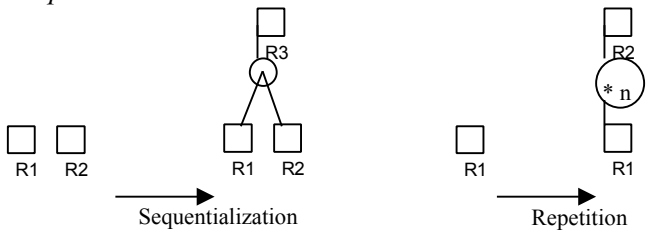

Figure 4: Schema of sequence and repetition operations.

The two other are information operations:

- the *symmetry operation* which builds a symmetric rule with an existing rule; by default the arbitrary reading direction is left to right. The symmetry operation reverses the reading direction of the rule.
- the *superposition* operation builds a superposed rule with at least two other rules which have the same measurement. The global measurement of each superposed rule is adapted to be the same as the others when a recalculating dimension operation occurs (see below).

Other Rule definition operations

The representation of a particular rule in the proposed model consists of a hierarchical composition of rules which finishes by elementary rules. Nothing however prevents from modifying a rule by modifying its structure or its elements (or by modifying the structure or the elements of one element which built it, …).

We have also identified some operations of composition, of decomposition, and recombination on the rule, such as duplication, particularization, inversion, regular filling, subdivision, fusion, hierarchical reorganization, etc. which are as many operations of assistance to the design.

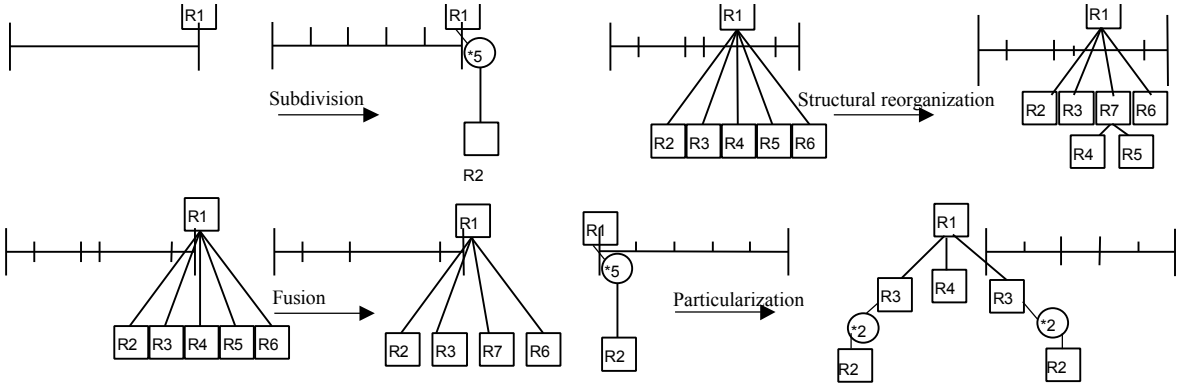

Figure 5: Schema of some operations on the rules

Rule value and rule measurement

To complete the model, we introduce two data, which are attached to the rules:

- The value of the rule which indicates the current length of the rule to one moment in the project and, for a repetitive rule, the factor of repetition.

- The measurement of the rule which is optional and indicates constraints on the value of the rule or on the type of the value (minimum value, maximum value, to be an integer for the factor of repetition).

Rules Grammar and semantic remarks

With the BNF (Backus-Naur Form) grammar we describe now a possible rules language:

Rule ::= *Basic-Rule* | *Sequential-Rule* | *Iterative-Rule* | *Superposed-Rule* | *Symmetric-Rule*
Basic-Rule ::= 'basic' *Value*
Value ::= *Real*
Sequential-Rule ::= 'sequential' *Value* '(' *Rule {Rule}*$^{+}$')'
Iterative-Rule ::= 'iterative' *Value Iterative-factor Rule*
Iterative-factor ::= *Positive-Integer*
Superposed-Rule ::= 'superposed' '(' *Rule* {*Rule*}$^{+}$')'
Symmetric-Rule ::= 'symmetric' Rule

As we can see, symmetric rules and superposed rules have no explicit value. The implicit value of a symmetric rule is the value of the rule on which is built the symmetric operation. In the same way, the implicit value of a superposed rule is the value of each rule which are taking part in the superposed rule. It is the same value because of the superposed rule.

OPERATIONS OF CALCULATING DIMENSIONS

However, the architectural grids (and rules which represent them) also permit other operations, which we call *calculating dimensions operations*, and which corresponds to the adjustment or perfecting of the architectural project.
The operations of (re)calculating dimensions which we propose consist of recalculating the whole of the rules (or a part of this one) when a local modification of dimension is made by the user (on one of the rules which composes it).

system of inequalities and equalities

The totality of the rules can be seen as one inequality and equality system. The inequality formulae on the rule values come from the measurement constraint if it exists. In the formulae below, $r_i.v$ is the effective value of the r_i rule, $r_i.min$ is the minimal value if it exists and $r_i.min$ is the maximal value if it exists.

$$\begin{cases} r_i.v \geq r_i.min \\ r_i.v \leq r_i.max \end{cases}$$

The equality formulae on the rule values come from the hierarchical structure of the set of rules. For each sequential rule exists one equality formula between the value of the sequential rule and the sum of the rules values which compose the sequential rule. For each iterative rule exists one equality formula between the value of the iterative rule and the product between the iterative factor and the value of the rule on which is built the repetitive rule. For each superposed rule exist equality formulae between the value of the iterative rule and the product between the iterative factor and the value of the rule on which is built the repetitive rule.

Example

Let the system of rules be (Figure 6):

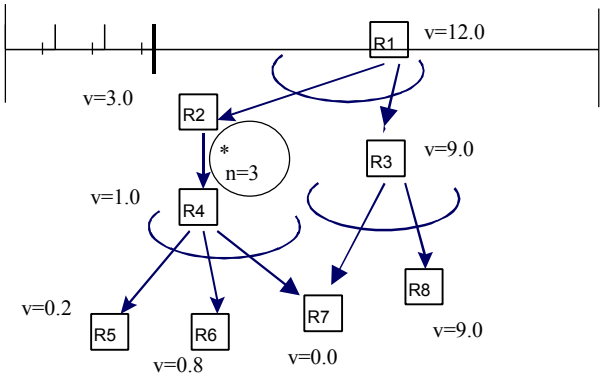

Figure 6: Example of rule system

This rule system can be translated into two systems; one is an inequation system, the other is an equation system (

Figure 7):

Rules	Type	Min	Max	value
R_1	Seq	12	12	12,00
R_2	Seq	0	1000	3,0
	factor	3	3	3
R_3	Seq	0	9,0	9,0
R_4	Seq	0	1,0	1,0
R_5	Basic	0,18	0,22	0,2
R_6	Basic	0,7	0,9	0,8
R_7	Basic	0	0	0,0
R_8	Basic	8.0	10.0	9.0

$$\begin{cases} 12{,}0 \le R_1.v \le 12.0 \\ 0{,}0 \le R_2.v \le 1000{,}0 \\ 3 \le R_2.\text{if} \le 3 \\ 0{,}0 \le R_3.v \le 9{,}0 \\ 0{,}0 \le R_4.v \le 1{,}0 \\ 0.18 \le R_5.v \le 0.22 \\ 0.7 \le R_6.v \le 0.9 \\ 0{,}0 \le R_7.v \le 0{,}0 \\ 8.0 \le R_8.v \le 10.0 \end{cases} \qquad \begin{cases} R_1.v = R_2.v + R_3.v \\ R_2.v = R_2.n * R_4.v \\ R_4.v = R_5.v + R_6.v + R_7.v \\ R_3.v = R_7.v + R_8.v \end{cases}$$

Figure 7: Inequation and equation system of a rule system

Objective function

These systems, to be solved, must be supplemented with a particular function called objective function. We identified two kinds of objective functions.

The goal of one of them is to optimize the project or a part of it. For example the architects want to maximize (or minimize) a surface, a space or a length. Thus the objective function is defined with one value rule combination (product, sum, etc.). This kind of economic function is used at the beginning of the project.

The goal of the other kind of objective function is to minimize the value modifications when a rule value change arises. Thus this objective function is defined as a the minimum of the sum of the distance between old and new rule values. This kind of objective function is used when the project is well in progress.

Choice of a method for Resolving the system

This kind of problem (having to solve system of equalities and inequalities with an objective function) is well-known since 1947 with the work of G. Dantzig on linear systems. Many mathematical programming techniques and methods (Simplex, Monte Carlo, Conjugate Gradient, etc.) were proposed ever since this date and can be applied to find the optimal solution of a such system (Lasdon, L. S. & all 1978). Depending on the problem to be solved, certain methods can be more or less efficient or absolutely not adapted (Legras J. 1980).

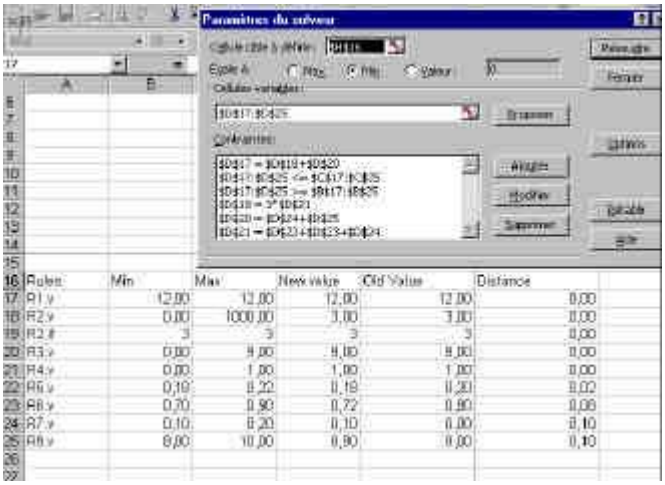

Figure 8: Resolution of the rules system with the Microsoft Excel Solver.

Because of the objective function which is a nonlinear function, of the constraints that are linear and of certain decision variables that are restricted to integers, we called our problem a "quadratic mixed-integer programming problem".

Theoretically, this problem can be solved efficiently with techniques that use the Simplex method to determine the feasible region and special methods based on the properties of quadratics to find the optimal solution for the quadratic part of the programming problem and a Branch and Bound method for the mixed-integer part of the problem.

Practically, we have used the Microsoft Excel Solver (Hui S. 1999) to verify that it is possible to find rapidly the newest values of the rules when a value modification constraint arises (Figure 8).

CONCLUSION AND FUTURE WORKS

What we viewed in this article is the possibility of organizing the architectural grids into hierarchies which are the support of values and measurement constraints. The alterations of constraint can be taken in account by values modification with specific algorithms.

We have now to verify the usability and the pertinence, for the architect-designers, of the proposed data model of grids. We work at the realization of a 2D CAD prototype which includes grids like those. A difficulty is to describe how the architectural objects are related to the grids. We chose to link the control points of architectural objects with the intersection points of the grids.

An other direction of work is on the data model itself. In the model, the grids are spaces series, however the architects conceive not only with spaces but also with axis. We have to integrate this concept of axis into the data model of grids.

BIBLIOGRAPHY

Bignon J.C. (1997), *La trame, un assistant à la conception technique*, les cahiers de la recherche Architecturale. éditions Parenthèses.

Gross M. D., *Why can't CAD be more like Lego ? CKB, a program for building construction kits*, Automation in Construction, n°5. 1996.

Hui S. (1999), *Excel Solver Add-In Tutorial*, http://commerce.ubc.ca/MBAcore/tutorials/MSExcelSolver/solver1.html.

Lasdon, L. S. & all (1978), *Design and testing of a generalized reduced gradient code for nonlinear programming*, ACM Trans. Math. Software 4, pp. 34--50.

Legras J. (1980), *Algorithmes et Programmes d'Optimisation non linéaire avec contraintes*, Ed. Masson.

Malcurat O., (1997), *La trame comme outil d'aide à la conception architecturale*, mémoire de Diplôme d'Architecte, Ecole d'Architectuire de Nancy, 1997.

Mitchel W. J. L, *The logic of architecture, Design, computation, and cognition*, The MIT Press, Cambridge, Massachusetts, 1990.

Pellireri G., (1997), *A tool for a first analysis of architectural façades*, Automation in Construction, n°5.

Zeitoun J. (1977), *Trames planes* ; Ed. Dunod, Paris, 1977.