John S. Gero, Scott Chase and Mike Rosenman (eds), *CAADRIA2001*,
Key Centre of Design Computing and Cognition, University of Sydney, 2001, pp. 369-378.

A "SPELLING" CHECKER FOR ARCHITECTURAL DRAWINGS

Grammatical and syntactic analysis in structured representations

ALEXANDER KOUTAMANIS [1, 2] AND VICKY MITOSSI [1]
[1] *Ministry of Housing*
[2] *Delft University of Technology*
The Netherlands

Abstract. CAAD representations for the early design stages have traditionally focused on aspects apparently relating to design creativity. These, however, may be unconnected to the control and analysis of design constraints that affect the further development of the design. The stability and reliability of control and analysis rely on what (despite the dangers of the linguistic analogy) we might call the *grammatical* and *syntactic* well-formedness of the representation. The paper reports on the control of grammar and syntax in a representation of spatial and building elements with respect to both the syntagmatic and the paradigmatic dimension.

1. Design representations and CAAD

The distinction between drawing and design in CAAD ideology would have remained one of the minor misguided academic discussions had it not being resurfacing in the ongoing computerization of architectural design practice. Despite valiant efforts, academic research and teaching cannot claim to be the main reason for the recent acceleration in architectural computerization. The sudden change in practice has been due to wider sociotechnological developments, such as the democratization of computer technologies and the digitization of information. The promise of increased efficiency, compactness and flexibility, as well as the sheer modernity of computerization, has resulted into a massive transfer of drawing activities to the computer. Unfortunately it appears that the primary purpose of the application of CAD systems is to produce analogue drawings. The lack of interest in comprehensive and effective design information has led to stagnation in the development of structure and utility in CAD drawings. This stagnation conflicts with two other developments in architectural practice. The first is the explosive increase of design constraints, mostly externally determined in e.g. the brief and building norms. The second development is the emphasis on early decision taking, frequently on the basis of earlier experiences, cases or precedents.

CAAD research is abundant in structured, well-defined representations, mostly for early and conceptual design. These representations have traditionally focused on aspects apparently relating to design creativity, such as flexible, effortless and rich geometric modeling (Schmitt, 1996) and automated design generation. However, modeling capabilities are generally unconnected to the control and analysis of design constraints that affect the further development of the design. These usually refer to functional and spatial aspects that are only implicit in a representation of design 'solids'. Representations from generative systems fare better in this respect, especially when they capture the duality of 'solids' and 'voids', i.e. spaces and the building elements that bound spaces (Steadman, 1983; Yessios, 1987). Such representations can form a basic nucleus for unified information processing in early design, from geometric modeling to floor area analysis and from visualization of activity allocation to feedback from light simulations (Koutamanis and Mitossi, 2000), as well as for the indexing and retrieval (Koutamanis, 2000a; Koutamanis, 2000b).

The transfer of representation methods and techniques from CAAD research to CAD practice means that the superficial drawing activities currently taking place in CAD systems must either be:

- *(bottom-up approach)* transformed into the implementation level of appropriately structured representations (Marr, 1982), or
- *(top-down approach)* completely replaced by such structured representations in new digital design environments.

In the framework of educational activities we have opted for the bottom-up approach, so as to build on the popularity of computerization and promote digital design thinking by reference to the possibilities and limitations of the practical side (Koutamanis, 1996; Koutamanis, 1999). This approach underlies a large proportion of CAAD education and is characterized by strong connections to analogue techniques and practices, either in terms of continuity or of explication (Mitchell and McCullough, 1995; Neuckermans and Geebelen, 1999; Schmitt, 1993).

Continuity has also been our main principle for the development of a representation of spatial and building elements for the Dutch Government Buildings Agency, which is responsible for the design and management of public buildings in The Netherlands (Leusen and Mitossi, 1998). In order to minimize input requirements and keep touch with the drawing practices in the Agency, the representation was implemented in a standard CAD system (AutoCAD, starting with version 13). The representation addresses specific requirements of the Agency and its activities, including measurability of areas allocated to different activities, analysis of functional aspects such as pedestrian circulation and evaluation of fulfillment of programmatic requirements (Mitossi and Koutamanis, 1998).

2. Well-Formedness as Grammar and Syntax

2.1. SEMANTICS AND STRUCTURE

The stability and the reliability of the representation and the performance of analyses and communication based on it rely on the accurate, precise and effective description of architectural entities in the representation. The well-formedness of the symbols denoting these entities and of the whole representation are the cornerstones of the bottom-up approach.

The semantics of the representation is straightforward. For practical reasons we have chosen to restrict our semantic basis to the basic integral entities of architectural thinking, spaces and the building elements that bound these spaces. These entities arguably form the basis for the description and analysis of practically all aspects. Moreover, this representation comes close to conventional drawings used in the early design stages. This similarity is essential for a quick and smooth transition from such drawings to structured design representations.

Also for practical reasons we have chosen to keep relationships between primitives implicit. The only additional information to the geometry and location of a symbol is its type (space, wall, door, etc.), which is indicated by modularization in layers, and eventual annotations of activities contained in a space, cost data or other alphanumeric properties, which are indicated by annotations and links to external databases. The implementation and maintenance of all relevant relationships is deemed too cumbersome and tenacious to be accepted as a rationalization of conventional representation. Instead, relationships are recognized automatically on the basis of local intelligence mechanisms (Koutamanis, 2000c). These make use of fundamental properties of the spatial or building primitives, such as their shape, location and functional type to extract properties, perform measurements, identify relationships and trigger actions. Local intelligence makes possible a high degree of abstraction and flexibility in the representation and avoids deterministic, normative characterization and classification of properties. Local intelligence also underlies the control of both main aspects of well-formedness in the representation:

1. The *grammatical* aspect concerns the well-formedness of symbols for spaces and building elements. These should be acceptable as symbols of real-world entities and remain consistent with the constraints of the implementation environment. Grammatical correctness ensures that data extracted from and measurements performed in the representation are precise and accurate.
2. The *syntactic* aspect refers to relationships between symbols in the representations. These too should reflect the real-world relationships

between spaces and building elements, and be unambiguous and consistent. Accurate and efficient recognition of local relationships, e.g. conditions of access between two spaces by means of a door or a similar opening, relies on syntactic correctness.

The inherent dangers of the linguistic analogy (Collins, 1965) and the multiple uses of linguistic terms in CAAD were counterbalanced by the connotations of structure and the precise identifications of control targets, means and goals. In particular, the distinction between grammar and syntax (symbols and relationships) and the analogy with spelling and grammar control in text processing has been beneficial to both developers and users of the representation.

2.2. GRAMMAR

The grammar of the representations refers to each symbol individually. The criteria which determine its acceptability as a representation of a real space or building element relate to the geometric structure of the primitive and in particular to the compatibility and redundancy of its primary features. In a floor plan or a 2½ D model we represent spaces and building elements as closed polygons, circles or ellipses. The sides of the polygons must be three or more if rectilinear or two or more if curvilinear. The well-formedness of a shape relates to the number and position of its vertices and the resulting relationships between sides.

One basic category of unacceptable forms is *self-intersecting shapes*, i.e. shapes with one or more pairs of sides that intersect at points other than their vertices. These shapes are impossible as spaces or building elements. For their detection we make use of two alternative strategies. The first is the detection of intersecting sides. To do so we must extract the vertices of the shape from the description of the polygon in the drawing's database:

$$(v_0, v_1, v_2, \ldots)$$

These vertices are then used to derive the sides which are not explicit in the polygon's internal description:

$$(s_1, s_2, \ldots) \text{ with } s_1 = (v_0 \,.\, v_1) \text{ and } s_2 = (v_1 \,.\, v_2) \text{ etc.}$$

The intersection of each pair of sides is calculated. If the intersection point is not a member of the vertices set, then the shape is self-intersecting.

$$I(s_i, s_j) = p_{ij} \text{ and } p_{ij} = v_k$$

The second approach is to test the transformability of the shape. Self-intersecting forms are normally not acceptable as input for the generation of higher-level graphic objects. For example, in AutoCAD a self-intersecting lightweight polyline cannot be used to generate a region.

Another basic category of unacceptable shapes is *open shapes*. These are by definition incomplete, as a space or a building element is always bounded by other, adjacent elements. An open shape misses closure, i.e. has all its vertices but misses one of its sides, normally its last one:

$$s_n = (v_n \,.\, v_0)$$

Other forms that are considered to be unacceptable on the basis of their general structure are judged by semantic criteria. For example, a rectangular space one hundred meters long and twenty centimeters wide is highly implausible, as is a near intersection, i.e. a similarly implausibly short distance between a vertex and a side. Corresponding acceptability criteria relate to the subject matter and can only be defined on an ad hoc basis for a specific class of designs or design representations.

Beyond impossible and implausible shapes CAD drawings may contain *erroneous objects* with invisible errors that do not alter the appearance of an object but may affect other important properties such as the number of its sides and its area or volume. Erroneous shapes are detected by the existence of redundant vertices. These are produced in abundance in CAD drawings and cannot be detected visually.

Redundant vertices fall under two basis classes: *coincident* and *trivial* vertices. Coincident adjacent vertices result into sides of zero length and otherwise coincident sides, e.g.:

$(v_0, v_1, v_1, \ldots)$ (coincident adjacent vertices)

$(v_0, v_1, v_0, v_1, \ldots)$ (coincident non-adjacent vertices)

Trivial vertices result into redundant, trivial sides. Trivial vertices are generally vertices on one side of a shape:

$$(v_0 \,.\, v_2) = (v_0 \,.\, v_1) + (v_1 \,.\, v_2)$$

Erroneous shapes pose a significant problem for the reliability and utility of computational design representations. Redundant sides mean that a rectangle may be mistaken for a pentagon and consequently generate misleading information, e.g. that a space has five walls instead of four. Also measurements can be severely altered by redundant sides. For example, the following vertices result into a measurement of zero area:

$$(v_0, v_1, v_2, v_3, v_0, v_3, v_2, v_1)$$

The following sequence doubles the apparent area:

$$(v_0, v_1, v_2, v_3, v_0, v_1, v_2, v_3)$$

Sides of zero length are not only redundant but also illogical. A CAD program may refuse to accept an erroneous shape of this type in certain operations, e.g. the creation of a higher-level graphic object, similarly to self-intersecting shapes. This simplifies the detection of some erroneous shape types but does not cover the whole spectrum of possible errors. Consequently, much effort has been put into the detection and correction of coincident vertices, with the following main conclusions:

1. The sequence of detection and correction is critical. It should start with coincident adjacent vertices, continue with non-adjacent coincidence and conclude with trivial vertices. If applied in a different sequence, it may result into deformation of the shape.
2. Most errors due to vertex coincidence can be corrected if the sequence of detection and correction is correct. This can be extended to shapes rejected because they were not transformable to higher-level objects, with the exception of self-intersecting forms.

2.3. SYNTAX

The syntactic dimension refers to the local and global relationships between primitives in a design representation. These relate to the relative positioning of different objects in the design and primarily to adjacency conditions. In a floor plan this means that grammatically correct adjacent objects that are share (part of) one or more sides. The syntactic well-formedness of the floor plan is evaluated on the basis of violations of this condition.

A fundamental category of syntactic errors is *coincident shapes*, i.e. grammatically correct shapes which share the same set of vertices:

$$(v_0, v_1, v_2, \ldots) = (w_0, w_1, w_2, w_3) \text{ if } v_i = w_j \text{ and } (v_{i+1} = w_{j+1} \text{ or } v_{i+1} = w_{j-1})$$

e.g.

$$(v_0, v_1, v_2, v_3) = (v_0, v_1, v_2, v_3) = (v_2, v_3 , v_0, v_1) = (v_3, v_2, v_1, v_0)$$

Coincident shapes are a physical impossibility, as they occupy the same space. Coincidence resulting from complex design intentions such as temporal aspects or activity allocation in the same space are not taken into account because they relate to non-geometric properties and should be implemented separately form the geometric representation of the design, for example as annotations.

Overlapping shapes coincide only in part, i.e. have intersecting perimeters. These too represent a physical impossibility. Overlapping can be detected either on the basis of intersecting sides and surfaces or by that some but not all of the vertices of one shape are inside another.

Inclusion is an intricate condition. On the apparent level we can claim that it is also subject to the physical impossibility of two objects occupying the same space. This, however, fails to account for entities such as holes (Casati and Varzi, 1994). The conventional solution in CAD drawings is to use auxiliary information, such as “obstacle” shapes which replicate included objects and link them to the including one. We employ local intelligence mechanisms which detect included shapes and determine their acceptability on the basis of their types and the type of the containing shape.

A final consideration on the syntactic dimension is the global *completeness principle*: the area or volume inside each building mass should be completely occupied by grammatically and syntactically correct shapes representing spaces or building elements.

3. Syntax and Grammar Checking

3.1. DRAWING DIMENSIONS

Grammatical and syntactic control can be performed along one or more several dimensions of drawing. These are common to a number of representations, from writing to speech. The *syntagmatic* dimension concerns the sequential structure of the representation. In a drawing this is the sequence of graphic production. In comparison to other representations, this dimension is relatively weak in drawings. In written text, for example, the sequence of words in a sentence reflects the syntactic and grammatical constraints of the language. In a drawing, on the other hand, the sequence by which elements are added is not necessarily related to structure of the described objects or to the development of the design in the designer's mind.

The *paradigmatic* dimension is complementary the syntagmatic one, as it concerns the range of standard primitives used in the representation. These can be analyzed at their own level of implementation mechanisms, as well as at the level of representation symbols (Marr, 1982).

The *mechanical* dimension relates to the anatomy of drawer in interaction with furniture and materials. Such constraints may seem trivial at first sight, but may determine several representation levels. For example, mechanical displacement of the arm may alter stroke matching. Also changes in starting position may affect direction of rotation around circles (Van Sommers, 1984).

The *cognitive* dimension operates on the two interrelated levels of action and perception. Action planning and control constrains syntagmatic and mechanical aspects through e.g. eye movement in response to action, positive feedback, anticipatory pursuit, and eye, head and hand coordination in general (Kowler, 1990; Miles and Wllman, 1993). Also relevant are issues of mental imagery and working memory.

3.2. PARADIGMATIC CONTROL

Our grammatical and syntactic control focuses primarily on the paradigmatic dimension. This means that control is applied to the whole representation (or well-defined, coherent chunks) first at the grammatical and then at the syntactic level. The grammatical control ensures that all primitives concerned are acceptable as symbols of spaces or building elements. Straightforward corrections such as closure or the removal of trivial vertices are performed automatically. Other problems are referred to the user. The syntactic control examines subsequently the spatial relations between primitives and identifies interpretation and implementation problems that the user has to resolve. Automated correction at the syntactic level is generally undesirable, as it entails establishing priorities of importance and flexibility in the design.

The main advantage of paradigmatic control is that it is performed on an if-needed basis, at the critical moments prior to information exchange or analysis. This makes control unobtrusive and efficient. On the other hand, feedback can be too extensive and confusing, especially if the representation contains many errors. Having to correct a wide range and large number of errors is a tedious and intensive task. This can be alleviated by the frequent use of paradigmatic control, so that feedback remains focused and restricted to the specific areas or aspects being processed. However, experience in both educational and professional applications shows that users tend to defer control and correction to critical moments.

3.3. SYNTAGMATIC CONTROL

An alternative to paradigmatic control at appropriate moments is continuous evaluation of user actions and results: the addition of a graphic object or the modification of an existing object triggers the control system that checks the grammatical and syntactic properties of the object and its neighbourhood. Syntagmatic control resolves the problems encountered in paradigmatic control but introduces an obtrusive agent in the user's activities. In an educational context (including on-the-job training) this may be acceptable but it adds little to the effectiveness and reliability of the representation beyond a certain stage of proficiency. Moreover, in drawing the syntagmatic dimension does not necessarily follow design reasoning. The sequence of drawing actions is frequently a product of extrinsic conditions and may involve auxiliary and temporary constructions. Control may interfere with such operational matters, unless the user is very disciplined and consistent.

The main promise of syntagmatic control lies in its learning potential. A control system capable of training itself and learning from experience involves inevitably more dimensions than the paradigmatic one. In particular the syntagmatic dimension and (through the ergonomics of syntagmatic aspects) the mechanical dimension have to be integrated in the triggering mechanisms of quality control. A truly responsive and at the same time unobtrusive grammatical and syntactic control should be able to anticipate the user's actions, as well as interpret them in variable, possibly multiple ways in order to capture cognitive intricacies and semantic variations.

4. Discussion

The well-formedness of a design representation is essential for the utility and reliability of information it contains. For example, errors in the area measurement of even modest numbers of spaces can have severely distorting effects on analyses of activity allocation. Similarly, the syntactic quality of the whole representation determines the completeness of the same information, as well as the unambiguity of spatial relationships in the representation. Overlapping spaces may also distort an area analysis, while the lack of adjacency between a door and a space means inability to detect access in identifying a circulation route. For such reasons the grammatical and syntactic quality of a representation become prerequisites to any meaningful computerized analysis of design aspects.

Experience with paradigmatic and syntagmatic control of grammar and syntax in educational and professional contexts suggests that control mechanisms also have a positive influence on efficiency, especially beyond an initial stage of acquaintance with the representation and its constraints. At the same time, in complex situations control may return feedback that is tedious and cumbersome to comprehend and process. For the moment, the appropriate balance between paradigmatic and syntagmatic control appears to be a matter of personal preference.

A control system is unlikely to have any long-term impact if it remains unrelated to learning. In the current implementations learning is restricted to user training with the representation. Future development should also address machine learning from syntagmatic and mechanical stimuli. These provide arguably cues for both anticipating following actions and inferring relationships between graphic elements. The explicit inclusion of more complex formal structures, i.e. local and global coordinating devices (Koutamanis, 1997), could provide a flexible reference framework for interpreting the meaning of graphic elements and related user actions.

References

Casati, R., and Varzi, A. C.: 1994, *Holes and Other Superficialities,* MIT Press Cambridge, Massachusetts.

Collins, P.: 1965, *Changing Ideals in Modern Architecture,* Faber and Faber, London.

Koutamanis, A.: 1996, CAAD teaching in the electronic era, *in* A. Ekholm, S. Fridqvist, and J. af Klercker (eds), *Education for Practice.* ECAADE , Lund.

Koutamanis, A.: 1997, Multilevel representation of architectural designs, *in* R. Coyne, M. Ramscar, J. Lee, and K. Zreik (eds), *Design and the Net,* Europia Productions, Paris.

Koutamanis, A.: 1999, Designing with the computer: the influence of design practice and research, *in* H. Neuckermans, and B. Geebelen (eds), *Computers in Design Studio Teaching,* KU Leuven, Leuven.

Koutamanis, A.: 2000a, Recognition of spatial grouping in rectangular arrangements, *Design and Decision Support Systems in Architecture, Proceedings of the 5th International Conference,* Eindhoven University of Technology, Eindhoven.

Koutamanis, A.: 2000b, Representations from generative systems, *in* J. S. Gero (ed.), *Artificial Intelligence in Design '00,* Kluwer, Dordrecht.

Koutamanis, A.: 2000c, Representing design generation in architecture, *in* C. Soddu (ed.), *Generative Art 2000,* Alea, Rome.

Koutamanis, A. and Mitossi, V.: 2000, On representation, *Design Systems Reports*, **2000,** 74-82.

Kowler, E.: 1990, *Eye Movements and Their Role in Visual and Cognitive Processes*, Elsevier,Amsterdam.

Leusen, M. V. and Mitossi, V.: 1998, A practical experiment in representation and analysis of buildings, *4th Design and Decision Support Systems in Architecture and Urban Planning Conference,* Eindhoven.

Marr, D.: 1982, *Computer Vision,* W.H. Freeman, San Francisco.

Miles, F. A. and Willman, J.: 1993, *Visual Motion and its Role in the Stabilization of Gaze,* Elsevier, Amsterdam.

Mitchell, W. J. and McCullough, M.: 1995, *Digital Design Media, 2nd ed,* Van Nostrand Reinhold, New York.

Mitossi, V. and Koutamanis, A.: 1998, Spatial representations as the basis of formal and functional analysis, *4th Design and Decision Support Systems in Architecture and Urban Planning Conference.* Eindhoven.

Neuckermans, H. and Geebelen, B.: 1999, *Computers in Design Studio Teaching*. KU Leuven, Leuven.

Schmitt, G.: 1993, *Architectura et Machina.*, Vieweg, Wiesbaden.

Schmitt, G. N.: 1996, *Architektur mit dem Computer,* Vieweg,Braunschweig/Wiesbaden.

Steadman, J. P.: 1983, *Architectural Morphology,* Pion,London.

Van Sommers, P.: 1984, *Drawing and Cognition,* Cambridge University Press, Cambridge.

Yessios, C. I.: 1987, The computability of void architectural modeling, *in* Y. E. Kaluda (ed.), *Computability of Design,* Wiley, New York.