Lines, Relations, Drawing and Design

Dr. Branko Kolarevic

University of Miami School of Architecture

This paper introduces a computer-based graphic environment for design conceptualization, or more specifically, for shape delineation and dynamic drawing manipulation, based on construction (regulating) lines and their geometric relations. It also presents ReDRAW—a limited prototype of a relations-based graphic system.

Introduction

The act of drawing, as well as other traditional ways of communicating spatial information in architectural design, is inherently static—it produces drawings, "snapshots" of an evolving design concept. However, shapes depicting an evolving design concept are seldom static—they change constantly. The act of designing is intrinsically dynamic.

This paper stems from the recognition of this disparity. It proposes a highly dynamic, computer-based graphic environment for design conceptualization, or more specifically, for shape delineation and dynamic drawing manipulation, based on construction, regulating lines and their geometric relations. The paper also presents ReDRAW, a limited prototype of a relations-based graphic system, and demonstrates its potential usefulness in conceptual architectural design.

In architectural design, as in other design disciplines, shapes are frequently constructed within some graphic context, which is, at a basic compositional level, set by some abstract organizational devices, such as grids, axes, and construction lines. Durand, Sullivan and Le Corbusier relied heavily on the concept of regulating (construction) lines. Le Corbusier’s architecture and painting from his purist period, for example, were guided by the application of regulating lines (‘les tracés régulateurs’), as shown in Figure 1.

Figure 1: Le Corbusier’s “les tracés régulateurs.”

These construction (or regulating) lines would be much more useful and interesting if they were used not just to create a drawing skeleton, but also to maintain and to regulate the behavior of a drawing’s structure as its elements are manipulated. Through geometric relations and dependencies, construction lines can control positions and orientations of other lines. With them, designers could structure the behavior of the object being designed under future transformations—drawings would become semantically charged and could be manipulated in a semantically sophisticated fashion.

forming an agenda for stabilization and reconciliation of forces on a collision course. They believe that architects may play a pivotal role as mediators of natural and human-made environments. Living up to their expectations will surely reveal new perceptions of the applicability of computers to architecture.
In this scenario, construction, regulating lines define a compositional framework for establishing positions and relations of shapes. Shapes are constructed as combinations of line segments delimited by intersecting construction lines. Each line segment has an underlying construction baseline intersected by two construction lines. This is very similar to traditional manual drafting practice, wherein "pencil" (construction) lines are laid out first, followed by "inking" of the selected portions between intersections [Tan 1991].

Figure 2: A floor plan of Mario Botta's Casa Rotonda composed of intersected 'pencil' construction lines and "inked" line segments.

Architectural composition then becomes a process of forming geometric relations between construction lines. Design begins by laying out interrelated construction lines, its organizing framework. Shapes are constructed by delineating underlying and intersecting construction lines. The designer adds new construction lines, relations, and shapes, and changes those that already exist. In the process, many different options may be explored.

Figure 3: An interpretation and incremental assembly of Mario Botta's Casa Rotonda based on the concept of construction lines and their geometric relations. Geometric shapes and relations are abstracted and translated into a relational drawing. New designs are created by applying the transformations of translation and rotation (Figures 4, 5 and 6).

One's choice of compositional relationships may result in a dramatically different designs even though only a small set of relations and transformations is available. How the composition is assembled, structured or re-structured determines its developmental potential. Using geometric relations, a designer can enforce desired configurations of building components and spaces. The established relations constrain the design possibilities—they structure possible...
manipulations. The relations, however, do not prescribe a particular form. Instead, they bound a space of alternatives without specifying a solution to the design task. As William Mitchell observed, "the choice of modeling conventions and organizational devices that will structure the internal symbolic model ... will determine how the model can be manipulated, and what can be done with it" [Mitchell 1990].

Figure 4: A possible transformation of Mario Botta's Casa Rotunda, based on an interpretation illustrated in Figure 3.

Dependencies determine the behavior of the composition—a designer must understand them to operate successfully upon them. This understanding is required on a basic, pragmatic level: If an object is moved, what other objects will move too? However, if the composition is too complex, its transformation might be difficult to control and envision. In other words, if hundreds or thousands of relationships and interdependencies determine the behavior of a geometric composition, it might be impossible, at least for the human mind, to predict the behavior of a composition subject to transformation. A designer might anticipate the particular result of applying a certain transformation to a composition, yet the system might produce something very different. Such an outcome, however, is not always undesirable.

Conception and Perception

"Information is purely formal and has no meaning. It is impersonal rather than interpersonal. The more it can be freed of the human component, that is, of such things as emotions and values, expectations and perceptions, the more reliable and valid it becomes... information presupposes communication. To be received, let alone used... requires prior agreement, that is, some communication."
The consequences of propagating changes to the composition can be very surprising. Indeed, resulting configurations can be genuinely new, and, in some instances, might trigger innovation and creativity. If the results of the operations were absolutely predictable, there would be little room for creative discovery. "Imagination needs something to play with," as Mitchell [1999b] asserts. A drawing can serve a vehicle on the path from the known to the unknown, from predictable to unpredictable.

Though it is clear that geometric relations play a pivotal role in this scenario, one must consider which geometric relations will constitute the compositional repertoire. One might start by looking at some of the geometric relations that are known to be useful in architectural design, such as alignment, connection, parallelism, perpendicularity, etc. Many types of relations are now known to be implementable and useful, and would provide an immediate degree of utility. Using this approach would provide potential insights into the more theoretical issues gained through actual use of the system that would implement the initial set of relations. New relations could be devised until a deeper understanding of computational and design requirements is gained.

The following lexicon of geometric relations is proposed as a starting point for investigation:

- CONNECTED AT a point
- INTERSECTED AT a point
- ALIGNED ALONG a curve
- PARALLEL TO a curve
- PERPENDICULAR TO a curve
- ANGLED TO a curve
- SYMMETRICAL (bilaterally) TO a curve

These geometric relations are present or recognizable in any architectural composition. At this point, it might be appropriate to present examples or case studies to support this contention. However, these relations are accepted as axioms, propositions that are self-evident though undemonstrated. This is not to suggest that these are the only geometric relations that comprise the lexicon. The hypothesis is rather that the number of geometric relations is large and cannot be determined in advance, but that a fairly small set of

Howard's End

INFORMATION + MEANING = COMMUNICATION

ACADIA 1994
carefully selected relations could provide an appropriate compositional repertoire. Another hypothesis defines new relations as combinations of already defined relations—after all, the lexicon provides only the morphemes in the language.

Even if we recognize essential geometric relations, we face the difficulty of representing them. In contrast to the domain of objects in geometric representation, there is no single established model or set of models that is widely used to represent relationships. As a consequence, asserts Robert Woodbury [1990], "the absence of such unifying concepts leaves the idea of modeling relationships outside of the formal realm of geometric modeling and consigns it to an ad-hoc role in any system." It is indeed quite difficult to define some unifying concepts in the realm of geometric relations. Different design domains rely on different sets of geometric relations. For example, "tangency" is one of the most essential relationships in mechanical engineering, while it is rarely exploited in architectural design.

Furthermore, the model we choose to represent the geometry can significantly restrict (or expand) the possibility of defining an effective and computationally efficient set of relations. The task is then twofold: not only will we look for an appropriate lexicon of relations for architectural design conceptualization, but also for an appropriate geometric context to represent relations in the lexicon. The process is bi-directional: the relations we admit in the lexicon will influence the selection of the geometric context, and the selection of the context will determine which relations may be included in the lexicon.

ReDRAW (RElational DRAWing)

This section introduces ReDRAW, a relations-based graphic system, as a means of exploring computational and application issues associated with the geometric relations. ReDRAW (RElational DRAWing) is a working but very limited prototype of a relations-based drawing system. Presented is its repertoire of geometric and relational primitives, interface, data structure, important algorithms, explicit and implicit representation of relations and bi-directional propagation of changes.

Description

ReDRAW is a relations-based drawing application modeled, in part, on traditional manual drafting practice. The user lays out infinite "pencil" construction lines while simultaneously specifying positional relations (none, parallel, perpendicular or angled) and dependency (none, uni-directional or bi-directional) between them. ReDRAW currently supports straight (linear) construction lines only. Its positional repertoire is limited to binary relations—parallel, perpendicular and angled lines. Ternary relations, such as symmetry and intersection, are not currently supported, since they might introduce cycles into ReDRAW's database representation.

To construct shapes the user "inks" selected portions of construction lines that are bound by intersections with other construction lines. Connectivity and alignment among shape segments are implicit and supported through the database structure, described later in this section.

The user manipulates created compositions by applying editing operations (erase, move, rotate) to selected construction lines. ReDRAW automatically propagates changes while maintaining previously established relations. If some of the relations cannot be maintained after transformation, ReDRAW can automatically establish new relations (in the Smart Mode) or delete them. The user can also change previously established relations, by altering either the type of relationship or its dependency.

Interface

ReDRAW's interface consists of three elements: a drawing window, an icon menu representing various tools and modifiers, and pull-down menus with file, edit and view commands. ReDRAW's icon menu consists of six groups of icons or tools:

(1) drawing tools: "magic wand," "pencil," and "pen;"
(2) positional modifiers (i.e., relations): none, parallel, perpendicular and angled;
(3) link modifiers (i.e., dependencies): none, one-way (uni-directional) and two-way (bi-directional);
(4) query tools: "show master(s)" and "show dependent(s);"
(5) editing tools to erase, move or rotate; and
(6) viewing tools to pan, zoom in and zoom out.

Figure 7: ReDRAW's drawing window with the icon menu.

Usage

The generation and transformations of drawings take place within the drawing window. As previously mentioned, ReDRAW is modeled, in part, on traditional manual drawing practice. The user first lays out infinite "pencil" construction lines while simultaneously specifying their relations and dependencies. To construct shapes, the user "inks" selected portions of the construction lines that are bound by intersections with other construction lines. Figure 8 illustrates the step by step creation of two overlapping "L" shapes characteristic of the work of architect Peter Eisenman. Tool icons ("pencil," positional and link modifiers) and a reference construction line (if a relationship is specified) are shown for each step.

Figure 8: Creating Eisenman's "L" shapes.

Data Structure

ReDRAW's data structure encodes a developing design in terms of construction (pencil) and inked lines. Construction line (ConLine) is a primary database construct. Construction line is an unbounded straight line specified as a structure in C programming language syntax:

```c
typedef struct
{
    Boolean deleted;
    int direction;
    float slope;
    float intercept;
    Relation parentRel;
    int numofChildren;
    Relation childrenRel[MAX_NUM_CHILDREN];
} ConLine;
```

Howard's End

ACADIA 1994
Description of the construction line based on the slope and intercept is derived from the algebraic definition of a straight line:

\[ y = \text{slope} \times x + \text{intercept} \]

where

\[ \text{slope} = \frac{\Delta y}{\Delta x} \]

Figure 9 illustrates this definition.

The revolution in communication technologies is changing the scope, pace, and organization of architectural practice and challenging many of our traditions and conventions. In the electronic age, several of the functions that used to engage our minds will be handled by computers and other communication media. Such things as detailed information, statistics, formulas, checklists or research data...
Finally, construction and inked lines records are stored in arrays—gConLines[] and gInkLines[], respectively.

Transformations

Linear geometric transformations of translation, rotation, reflection and scaling represent a standard repertoire for manipulating the compositions. Currently, ReDRAW supports only Euclidean transformations of translation and rotation, which may be applied to construction ("pencil") lines only.

Figure 13 illustrates, step by step, a possible way of transforming the two overlapping "L" shapes whose creation process was depicted in Figure 5. Icons of chosen tools and a selected construction line to be transformed are shown for each step.

Applying a transformation, such as translation or rotation, to an object might require that the same transformation be applied to related objects. In ReDRAW, translation or rotation of a single construction line results in translation and rotation of the line’s "children" (i.e., the construction lines linked to it in a uni- or bi-directional relationship) in order to satisfy already established relationships. Transformation is applied recursively to the children's children until there are no more dependents to trace. In addition, if a construction line to which a transformation is originally applied has a bi-directional relationship with its "parent," transformation is also propagated upward through the hierarchy.

Propagating translation throughout the hierarchy does not violate existing relationships. However, parameters associated with certain relationships might change as a result of the translation. For example, if the construction line to which the translation is applied in upward propagation has a uni-directional parallel parent relationship, the distance between the two will change.

Rotation, on the other hand, can invalidate some of the uni-directional relationships in upward propagation. If line B, for example, is in a uni-directional parallel relationship to line A, and if it is rotated, its relationship will be voided. Since the dependency is uni-directional, line A will not be affected by the rotation. If the relationship is voided, there will be a conflict to resolve. This exemplary conflict can be resolved in three ways. One solution is to cancel the transformation. The second solution is to eliminate the relationship since it is violated. The third solution might require establishing a new relationship, since the original intention was to have the two lines related. Even if the solutions to a given conflict are known, a new question emerges: Who should select a proper solution, the user or the system in some "automatic" fashion? If the user must select an appropriate solution, the system could pose numerous "low-level" questions to the user as a result of a single transformation. On the other hand, if the system has to make a choice, the user might have little or no control over the conflict resolution process.

ReDRAW's conflict resolution mechanism, in its current implementation, bridges these two extremes by operating in either an inactive or active ("smart") mode. In its inactive mode, ReDRAW simply eliminates invalidated relations. In its "smart", active mode, it establishes new uni-
directional relationships by computing the angle between the two construction lines. This is not a perfect solution, since all of the conflicts that might result from propagating a single transformation through the hierarchy are handled in the same manner. In other words, invalidated relations are either eliminated or new ones established. However, this solution does eliminate extensive user intervention in solving the potentially numerous low-level conflicts that might be too distracting and unimportant in design conceptualization. After all, if the results of propagation are unacceptable, the user can always use the “undo” command to reverse them.

In addition to Euclidean transformations of translation and rotation, ReDRAW also supports deletion of “inked” and construction “pencil” lines. Deleting a construction line can also invalidate the previously established relations. For example, if line A is in a parallel relationship to line B, and if line B is in a parallel relationship to line C, deleting line B will invalidate both relations. Again, there are three possible solutions. First, deletion could be canceled; second, both relations could be eliminated; third, a new relationship, line A parallel to line C, could be established based on invalidated relations.

ReDRAW also handles deletion in two modes: inactive and active. With inactive mode, invalidated relationships are eliminated. In active mode, ReDRAW establishes new relationships based on invalidated relations. Initially, ReDRAW’s “inference” mechanism was envisioned as being based on logical properties of relations (symmetry and transitivity) and valid deductive inference. If we analyze the previous example, wherein line A is parallel to line B, and line B is parallel to line C, we may deduce, based on the transitivity of the parallel relationship, that line A is also parallel to line C. Implementing such an inference mechanism is straightforward—each relation supported by the system will have its symmetry and transitivity lookup table.

However, since ReDRAW’s repertoire of relations comprises only parallel, perpendicular, and angled relationships, a simpler and more efficient mechanism is implemented. Parallel and perpendicular relations could be considered as special cases of the angled relation, respectively measuring 0° and 90° between construction lines. If we have two relations, for example, line A in relation R1 to line B, and line B in relation R2 to line C, then we can determine a third relation R3 between A and C simply by computing the angle between A and C.

Since relationships in ReDRAW are uni- or bidirectional, dependency direction must also be taken into account. ReDRAW preserves the direction of the relationship that is “higher” in the hierarchy. If we consider the previous example, the new relationship, R3, will have the same dependency direction as relationship R2.

Again, the user has a choice of setting ReDRAW into the inactive or active (“smart”) mode. In active mode, inference computation is hidden from the user. The final result is what one might call a “transparent” and “fluid” restructuring of the composition.

ReDRAW also provides for the substitution of once established relationships. Both the type of relationship and its dependency could be changed through the “magic wand” tool. After a new relationship and dependency are selected, the user first selects a construction line whose parent relationship will be changed, and then selects a new reference, a new parent construction line. A relation change that requires reference substitution will result in the reparenting of the tree structure—the process through which a tree node and all of its children are moved to a new parent.

Changing or substituting an existing relationship can introduce cyclical dependencies. For example, if A -> B (line A in relationship R1 to line B), B -> C, and C -> A, then changing C’s relationship to C -> A will result in a cyclical dependency. ReDRAW can discover a dependency cycle by tracing the hierarchy both upward and downward. Upon recognizing a potential cyclical dependence, ReDRAW cancels the substitution and informs the user of its action.

Since hundreds, or even thousands of geometric relations might be established in an architectural parti, a designer will need some ability to anticipate the consequences of propagating changes after transformation through the

will be readily retrievable. The critical human mental tasks will be those which will provide a context or framework within which to convert this information into effective communication. We will focus on such things as creative thinking, problem solving or creative formulation of problems or identifying needs, resources and goals. This shift in the focus of human thinking has particular
composition. The complexity, or a number of relations alone, will make the “mental” tracking of dependencies almost impossible. A computer-based graphic context, such as RedRAW, should therefore aid a designer in visualizing the dependencies within a drawing. In other words, a designer should have the ability to query the database about dependencies and relationships.

RedRAW supports four types of queries regarding the dependencies and relationship established in a composition. First, the user can query the database on the parent relationship of a selected construction line, and request the information about its type, dependency direction, and a reference (or parent) construction line. Second, the user can request that direct dependents of a selected construction line be displayed. Since RedRAW supports both uni- and bi-directional dependencies, transformations can affect construction lines throughout the hierarchy, not just those “below” the construction line to which the transformation has been directly applied. As a consequence, the user can query the database to display all of the construction lines that will be affected by the application of a certain transformation. In addition, the user can request the display of all of the construction lines whose transformation will affect the selected construction line.

RedRAW’s Limitations

RedRAW is limited in its features. Its primary purpose was to demonstrate the ideas presented in this study. In its current implementation, it supports only straight lines and linear relations and provides only rudimentary editing and viewing operations. Its primary deficiency is the lack of two very important ternary relationships—bilateral symmetry and intersection. Since ternary relations require two references, potentially cyclical dependencies could be developed in the drawing structure. Implementing ternary relations would require a different database structure and a different database maintenance mechanism.

In its current capacity, RedRAW supports only hierarchical, uni or bi-directional dependencies. Its maintenance mechanism is based on simple, direct propagation. However, as additional features are added, such as the support for ternary relationships, its maintenance mechanisms will become increasingly more complex, and will probably rely on relaxation to resolve potential conflicts.

Currently, RedRAW deals exclusively with geometric relations. In its current implementation, it does not support parametric definition of relations. Though the current database design does support numeric parameters, it does not support formulas.

RedRAW does not provide “search and replace” function of shape grammars. However, its database structure, coupled with ECART’s shape recognition capabilities [Tan 1991], could provide such a feature.

Conclusion

This paper proposes a dynamic relations-based graphic environment for design conceptualization that could provide a qualitatively different way to explore shape, dimension, and geometric organization. On a more specific level, it explores a relational description of shapes based on the concept of regulating (construction) lines and their geometric relations. It demonstrates how interrelated construction lines, as an organizing “device” in design conceptualization, could become much more useful and interesting when they are used not just as a rigid skeleton, but also to regulate the behavior of a drawing and to maintain its essential structure while its parts are manipulated.

The principal conclusion of this study is not that designing is necessarily done as proposed, but that it might and beneficially be. The proposed relations-based approach to design conceptualization would benefit designers by allowing them to generate new information within the design task efficiently and effectively through graphic processes, i.e., by providing a graphic means of generating new but always contingent information through dynamic manipulation of the design object’s relational structure. The proposed approach would expand the designer’s ability to speculate about possibilities. It places value on explicit formulation—its use requires “discipline” and an understanding of the relation-based approach to
design as a method. Once the approach is understood, it may be used effectively to program the “behavior” of a design object.

This paper also presents ReDRAW, a limited prototype of a relations-based graphic system, demonstrating its potential usefulness in conceptual architectural design. Like most prototype developments, ReDRAW evolved from assumptions and expectations that might now require some change in order for ReDRAW to develop into a more fully implemented design tool. Put in perspective, ReDRAW represents a beginning of a search for a new approach to computer-based design conceptualization—a promising one, I hope.

Notes

1 Although independently conceived and developed, ReDRAW and ECART, by Milton Tan [1991], share a similar data structure. Two different, yet complementary concepts—relations-based representation and transformation, and recognition of emergent shapes—are supported by a similar database design.

2 It is important to note that this description applies to binary relations only, such as parallel or perpendicular. Ternary relationships, such as intersection or symmetry, would require an additional index number pointing to a second reference construction line.

3 Note that such a description of inked line segments provides for implicit representation of connectivity and alignment relationships. If two inked lines share the same base construction line they are “automatically” considered to be aligned.

4 Arrays are used instead of linked lists for some purely practical reasons, primarily because the access to array members is faster than access to the members of linked lists. However, arrays are limited in size while linked lists are limitless. Also, elimination of a member is faster in linked lists than in arrays. The future development of ReDRAW will implement linked lists.

5 Tan’s ECART prototype for shape recognition and ReDRAW share a similar database representation. By incorporating the results of Tan’s study, ReDRAW’s value as a conceptualization tool could be considerably expanded.

References


significance for designers in that problem recognition, problem resolution and creative exploration of alternatives have been the traditional core of design education. This makes it incumbent on us now to convert the fundamentals of the craft-oriented architectural education to a body of knowledge. In the future the critical knowledge base for the designer will shift from detail, prototypes, or conventions to one of processes and performance.