A Computable Language of Architecture

Description of Descriptor Language in Supporting Compound Definitions

Sora Key
Carnegie Mellon University, Computational Design Laboratory, USA
http://www.code.arc.cmu.edu
sora@cmu.edu

Abstract: *Language is a particular kind of formal structure. It allows systematic approach to the subject matter for the speaker while providing an exploratory space to reason with. In this paper, we present a simple language that describes spatial qualities of architecture based on the geometry of built elements. We also provide a detailed description of the components and the structure of our language and show how one might construct compound definitions using the language. We then discuss the implication of constructing a language and how we can use it as a tool and research model in the study of architecture.*

Keywords: *Computational representation; qualitative analysis; design tool; spatial description; architecture design.*

Introduction

Language is a particular kind of formal structure. It allows systematic approach to the subject matter for the speaker while providing an exploratory space to reason with. In this paper, we present a simple language that describes spatial qualities of architecture based on the geometry of built elements. We provide a detailed description of the components and the structure of the language and show how we can use the language in constructing compound definitions.

We use the term 'language' in a very limited sense. Our language is a set of computational primitives that can be combined to describe architectural qualities as if architects describe those qualities over floor plan diagrams. We broke down the terms to components to describe building user's experience. Using the basic components, the language can be structured and reconstructed through logical and algebraic operations.

The way we identify terms in constructing our language is rather heuristic and inclusive, ranging from building code (NRC, 2005) to environmental design research literature (Billig and Churchman, 2003; Joseph and Zimring, 2007; Stone, 1998). Some definitions are more frequently appear in the literature – enclosure, visible, view, continuity, accessibility; others are inferred and derived – 'reachable' from 'visible' and 'accessible' from 'reachable.' Two criteria in selecting terms were whether the term can be 1) analyzed by geometric attribute and 2) associated to experiential qualities of architecture space that may have certain impact on behavioral patterns.

Selected terms are treated as the encapsulated object, forming descriptions of the geometric configuration called definition. Definitions are composed of geometric attributes such as distance, angle, length, or area of the abstracted building elements. Those attributes are combined to represent perceivable relationship of the user of the space to the elements – visible, near, far, adjacent, and

narrow. All these consist of the set called primitives. Primitives can be combined by the user of the language to build more terms, compound definitions. The current study provides built-in definitions, however the user of the language can override current definitions as they compose.

We consider it as a platform, a toolbox of functions that take spatial elements and field and return Boolean or numerical values. Language, in this limited sense, serves a role of referential entity, offering rationale for the analyses. Our language is an attempt to illustrate how designers might reason about and extrapolate feelings and behaviors from the measurable factors of architectural space to communicate. What we build is a subset of existing domain language in a more communicable form. The body of our language is by no means complete. However, we believe that, this approach can be a productive first step for future study, as a means to an end.

Problems

We communicate over spatial qualities in many different contexts: in a design studio, when critics talk to students; in a client meeting of architects; in daily life, when people talk about an apartment. We talk about spatial qualities such as enclosed, open, or continuous. Those spatial qualities are perceived and experienced thus one person's notion of spatial qualities may differ from another's. For example, a space can feel spacious enough to a group of people from one culture but not from others. However, although there are many parameters and variables designers can put in, we can still talk about a room being 'spacious.' How much space is considered to be spacious to a particular culture or purpose? Designers should be able to answer the question with a consistent language that others can understand.

Spatial qualities are relative to intended use, and use is culturally relative. Some parts of making design decisions on spatial qualities are objective. Those qualities are important aspects of spatial composition in architecture design; they are commonly referred, discussed and debated. Yet designers haven't had an explicit way to describe and compare qualities of architectural spaces in sufficient detail. Language helps for different people to articulate their notion of spatial qualities. It formalizes and structures ill-defined concepts. We don't have a language specific enough to talk about spatial qualities for architecture, so we build one.

Related work

Computational spatial description of spatial qualities is an attempt to put the representation of the spatial environment on a quantitative base. Existing studies on spatial analysis upon the built environment heavily rely on visibility. Isovist, a visibility model, proposed by Benedikt (1976) provides foundational work of the formal description of visual perceptual field in architectural study. Kincaid's spatial definitions (1997) provide another example to read architectural space in quantifiable terms that can be directly modeled on a computer. He proposes a way to read the architectural field mathematically, which enables formal reading of territory based on the attribute of physical elements.

Analytical methods, related to design tools, can support decision making. Do and Gross (1997) introduces a collection of various analytical methods of architectural qualities based on isovist. They propose to develop CAD tools to understand designers' decision making process. Among various approaches, visibility graph model from space syntax has gone to integration of visibility model to an actual tool that helps to predict human movement based on social relationships (Turner et al, 2001). In a prototype system called TAC, Koile (2001) shows how abstract terms are mapped onto physical characteristics of a building to enable design decision support. Experiential qualities of architecture space are inference rules in classifying physical characteristics of building elements.

Language, by nature, shapes mental representation of conceptual construct. Detailed classification

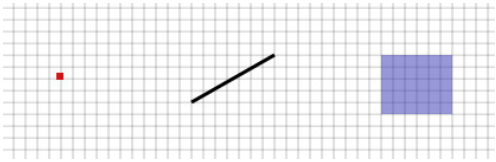


Figure 1
Graphical notation of elements, point, line, space within field

of types of components and relations can be found in studies of psychology of linguistics. In structuring spatial relations of verbal language, Gordon and Sadler (1996) distinguish object relations (e.g. location x, y of a thing) from referential relations (e.g. X is above Y). Bowerman (1996) addresses that building semantic primitives, meaningful chunk of relationships, from small units serves mental classification to help apprehension of spatial relationships.

Descriptor language

Our language addresses certain concepts that describe perceivable spatial experiences. We call them spatial qualities. Our language comprises the relationships between the built form and the user experience of the space. To analyze the geometric configuration of the form of built elements, we first build up the relationships into mathematical functions. We identify entities, relationships, parameters and variables. We currently provide 8 built-in qualities as the basic terms. The language consists of several components: *elements, primitives,* and *definitions*.

Elements

Elements are point, line, and space. Point indicates location; line implies an edge; space is a zone where activities can happen. Line represents the abstract geometry of built elements, the objective properties of built form. A point also represents a building user. Each element has its own geometric attribute called *property*. A point has (x, y) coordinates. A line has length, angle and the properties of point. A space has area and ratio on top of all properties of line. Graphic notation of Elements is shown below (Figure 1).

Primitives

Primitives are relationships of the configuration of multiple elements while *properties* are objective (geometric) attributes of a single element. *Length, height,* and *size* are properties. *Distance, angle, parallel,* or *overlaps* are primitives that require multiple elements. There are two types of primitives: *geometric primitives* and *perceptual primitives*. Geometric primitives are relationships between built elements. Perceptual primitives are relationships between a viewer and built elements. Below is the list of Geometric Primitives. The right side of the arrow shows the type of returned value.

- distance (element, element) →number
- overlap (space, space) → Boolean (yes/no)
- intersect (line, line) → Boolean (yes/no)
- angle (line, line) or angle (point, line) → number (degree)
- contains (space, point) or contains (space, line) → Boolean (yes/no)
- on (line, point) → Boolean yes/no
- parallel (line1, line2) → Boolean (yes/no)

List of Perceptual Primitives is as below. Square brackets indicate an array. Curly brackets indicate a set.

- visible_elements (viewer, [elements]) → [list of elements]
- visible (viewpoint, [elements], field) → {set of visible points}
- adjacent (viewpoint, [elements]) → [list of the surrounding elements on e.g. 4 side]
- nearest (viewpoint, [elements]) → closest element to viewpoint

Some perceptual relationships require and additional parameters as the appropriate range. Range can be determined by the functional activity of the space. For example it is possible to determine the range of narrowness by the activity (e.g. for walking, seating for a group of 40 people). Below terms assumes that they measure the space of the interest that contains the given point provided as an argument.

- narrow (element, range) → Boolean (yes/no)

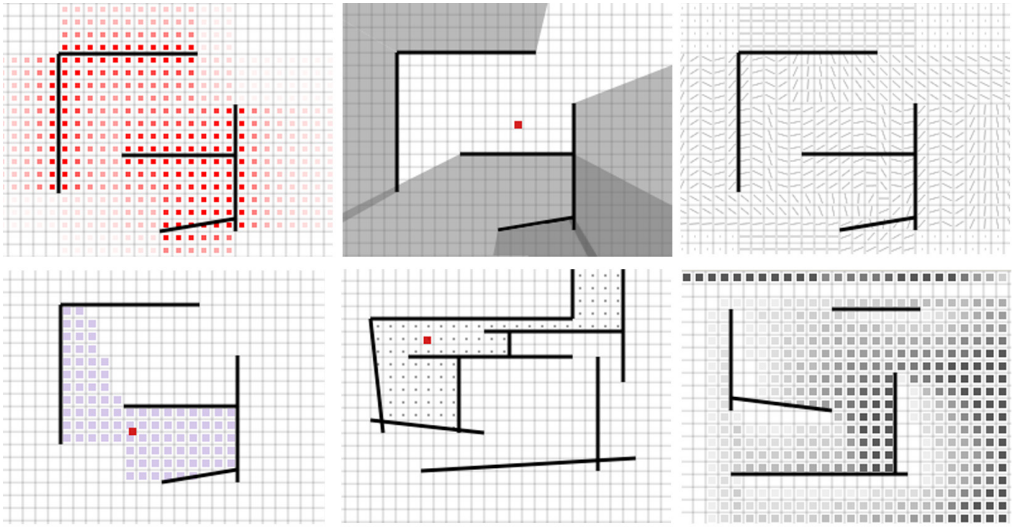


Figure 2
Graphic representation of built-in definitions: screenshots of Enclosure, Viewfield, Directionality, Continuity, Reachability, Spaciousness. (From left to right, top to bottom)

(antonym is wide)
- far (element, range) → Boolean (yes/no) (antonym is near)
- nearby ([elements], range) → [elements within range]

Built-in definitions

Definitions are the basic terms we provide as qualities that encapsulate multiple mathematical functions (primitives). Our language currently provides eight built-in definitions: *enclosure, viewfield, continuity, directionality, reachability, accessibility, spaciousness, and transitionality* (transitionality and accessibility are currently under implementation). Below shows the graphical representation of the built-in definitions followed by the list of mathematical notation of the built-in definitions (Figure 2).

- Enclosure (field, [walls]): sum of inverse distance of all visible walls
- Viewfield (field, [walls], viewpoint): a set of visible points from viewpoint
- Continuity (field, [walls], viewpoint, range): a set of point that satisfies two conditions 1) visible to and from viewpoint, 2) within range of any visible wall from viewpoint
- Directionality (field, [walls]): average angle of nearby walls
- Reachability (field, [walls], viewpoint): a set of points that have at least one existing path from viewpoint
- Spaciousness (field, [walls], viewpoint, angle): size of the set of visible points within 120 degree towards angle divided by size of the set of all visible points from viewpoint
- Transitionality (field, space1, space2) = overlapped area (space1, space2)
- Accessibility (field, [walls], viewpoint) = inverse (number of turns multiplied by distance)

We structure relationships between components of the built-in definitions in a hierarchical manner (Figure 3). Components at the bottom are measurable. Terms at the top are higher-level description, names of built-in qualities.

Creating compound definitions

Operators

The language supports compound definitions. One can combine primitives and definitions using the logical *operators*: *conjunction (AND), disjunction (OR),*

Figure 3
Structural hierarchy of components

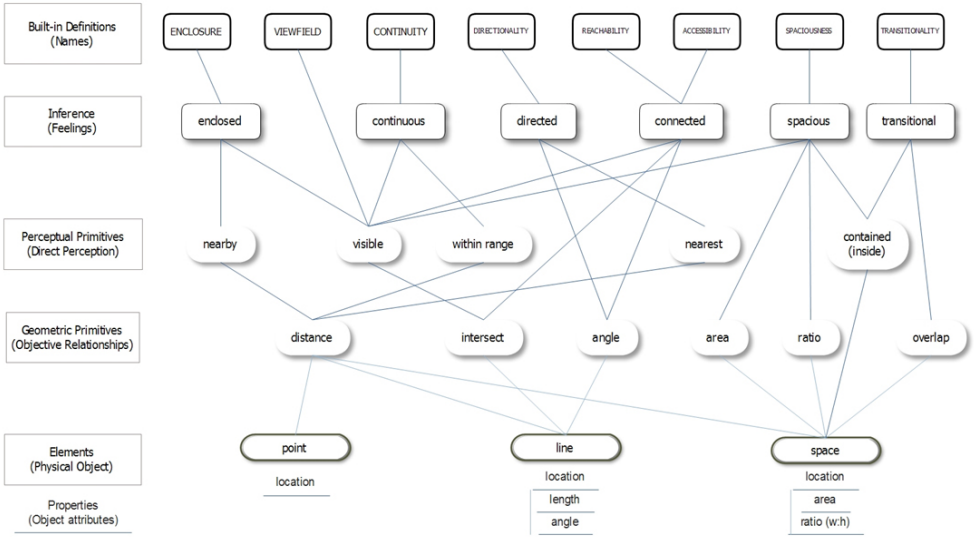
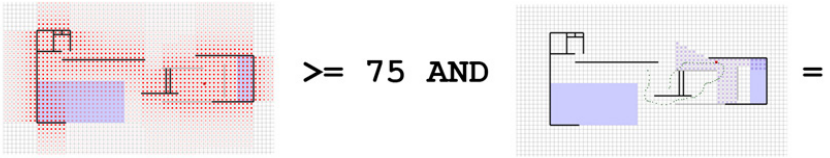
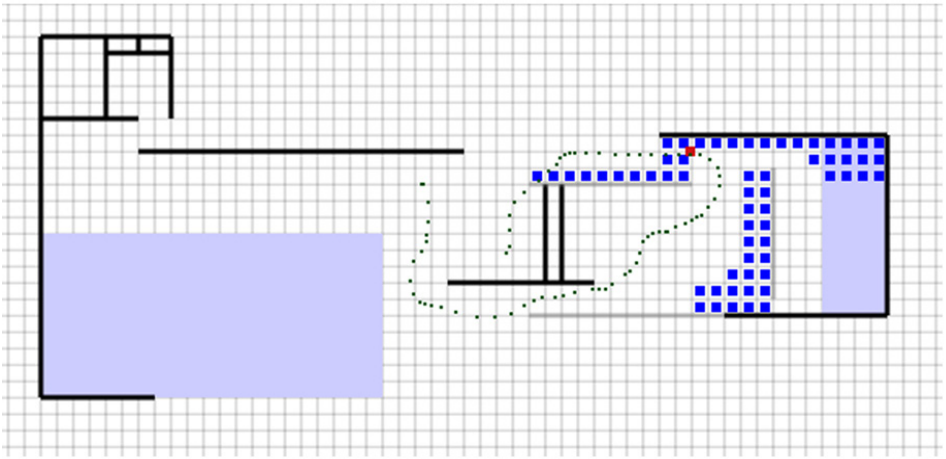


Figure 4
Graphic representation of preferred seating area following equation 1. Blue color indicates preferred seating area. Red dot indicates a viewer's location. Green dots indicate a path of the viewer





and *negation (NOT)*.

An example

Spatial qualities can play the role of selection criteria. One can construct a definition by selecting primitives and built-in definitions to create compound user-defined qualities. We can think of higher level description of qualities for a type of activity. An architect defines *preferred seating area* as highly enclosed yet still continuous so one wouldn't feel isolated by the lack of visual connection while sitting. He defines preferred seating area as 'highly enclosed and continuous.' Using our language, his definition of 'preferred' for seating activity can be represented as in equation 1.

preferred (field, viewpoint, range) =(enclosure (field) >= range) AND continuity (field, viewpoint) (1)

Parameter *field* is the designated architectural space with width and height. Viewpoint is the viewer's location. Below figure shows the result of the analysis from each definition (enclosure, continuity) and the result of the operation as the architect sets the level of enclosure (for the preferred sitting area) at 75% or greater (Figure 4).

Rules for building compounds

1. Using logical operators, it is possible to create more definitions with subtle difference. Two compound definitions that use same built-in qualities, visible and reachable, can be constructed as below shown in python code (Figure 5).
2. It is possible to use basic definitions with geometric primitives (Figure 6).
3. It is also possible to create a new definition using geometric primitives only (Figure 7).
4. More or less extensive description (tight or loose) is possible by adding more conditions to the term (Figure 8).

```
def forbidden (space, viewpoint):
  visible (viewpoint) AND (NOT reachable (viewpoint))

def hidden (space, viewpoint):
  (NOT visible (viewpoint)) AND reachable (viewpoint)

def welcoming (space, range, viewpoint):
  accessible >= range AND (NOT forbidden (viewpoint))

def private (viewpoint):
  (NOT visible (viewpoint)) AND enclosed

def prospect (space, viewpoint) = private (viewpoint) AND spacious
```

Figure 5
Compound definitions

```
def recessed (space, viewpoint):
  hidden (viewpoint) AND adjacent (viewpoint)
```

Figure 6
Compound definition built of a built-in definition and a geometric primitive

```
def bright (space, range, viewpoint):
  sum (height (surrounding_walls (viewpoint))) / size (space)  > range
```

Figure 7
Compound definition built of geometric primitives only

```
def segregated1 (s1, s2, s3):
  NOT (overlaps(s1, s2) OR overlaps(s2, s3) OR overlaps(s3, s1))

def segregated2 (min_distance, s1, s2, s3):
  (segregated1(s1, s2, s3)
AND distance (s1, s2) >= min_distance
AND distance (s2, s3) >= min_distance
AND distance (s3, s1) >= min_distance)

def segregated3 (range, s1, s2, s3):
  (segregated1(s1, s2, s3) AND
  (far(s1, s2, range) AND far(s2, s3, range) AND far(s3, s1, range))
```

Figure 8
Multiple definitions under same name

Discussion

We think there are at least two good reasons for creating and using a language in the study of architecture. The first is communication. The primary purpose of language is communication and we believe that spatial qualities can be communicated and should be. Designers have spatial concepts they want to clarify and deliver. With a language the concepts can be defined and presented in a communicable way to others for further discussion and feedback. A language serves the intermediate structure that signifies and symbolizes ideas about architectural space and our approach illustrates how we can decompose and recompose those conceptual terms that describe spatial qualities of architecture from measurable attributes, geometry of the built form. The second is the extension of the language as a design tool. We have current technology available and proper use of language is beneficial in enabling us to harness different types of tools we already possess. For example, a language can be used to analyze

existing designs in the study of architecture design. Knowledge-based, geometric modeling and visualization, and simulation tools are the opportunities we can expect to develop further. A language, embedded in various types of tools, can provide a framework to analyze, predict, evaluate and test our design solutions in a scientific manner. It can also broaden the possibilities of using existing generative digital media, offering important criteria in decision making for design exploration. Computation provides one way to build a tool (a language) for our purpose.

Implication of the use of our language

We see the implication of our language as a descriptive framework that can be further utilized as a research model in the study of architecture. We present a computable language so we can implement the language. It provides the users or designers a way to build up conceptual definitions of spatial qualities as a means to analyze the geometry of their designs. Those conceptual terms might be useful for the language users during design processes. When designers project their ideas of geometry onto the meaningful spatial experiences, they also predict the use of the architectural spaces. Spatial qualities, the consequence of the geometric configuration, indicate the complex relationship between the form and the user of the configuration. We believe, although complex, the relationships can be structured if we have a medium that can help us structure them. It should read as the first step towards collaborative effort by serving a tool kit to explore and gather existing domain knowledge that is familiar to designers.

The consequence of building a language is twofold. In doing so, computation of the language offers a foundation to develop a necessary toolkit for testing (simulation) purpose. For example we can run simulations against to the observable patterns of activities in existing building spaces. This type of study relates the use of the language in a scientific manner. The use of language may also help clarification of the vague ideas in defining spatial experience as a referential entity, a term that describes personal experiences in a communicable way, which can be further extended by the user. In this way, language offers an exploratory design space to the speaker.

References

Benedikt, ML: 1979, To Take Hold of Space: Isovists and Isovist Fields, Environment and Planning B., 6, pp. 47-65.

Bowerman, M.: 1996, Learning How to Structure Space for Language: A Crosslinguistic Perspective, in P. Bloom, M. Peterson, L. Nadel and M. Garrett (eds), Language and Space, MIT Press, Cambridge, pp. 385-436.

Billig, M and Churchman, A: 2003, Building Walls Of Brick And Breaching Walls Of Separation, Environment and Behavior, 35(2), pp. 227-249.

Do, E. and Gross. MD: 1997, Tools for Visual and Spatial Analysis of CAD Models, Proceedings of CAAD Futures '97, Munich, Germany, pp. 189-202.

Joseph, A and Zimring C: 2007, Where Active Older Adults Walk: Understanding the Factors Related to Path Choice for Walking Among Active Retirement Community Residents, Environment and Behavior, 39(1), pp. 75-105.

Kincaid, D.: 1997, An Arithmetical Model of Spatial Definition, Department of Architecture, unpublished Master of Architecture thesis, MIT, Cambridge, MA.

Koile, K: 2001, The Architect's Collaborator: Toward Intelligent Tools for Conceptual Design, Ph.D. dissertation, MIT, Cambridge, MA.

Logan, G. and Sadler, D.: 1996, A Computational Analysis of the Apprehension of Spatial Relations, in P. Bloom, M. Peterson, L. Nadel and M. Garrett (eds), Language and Space, MIT Press, Cambridge, pp. 493-529.

National Building Code of Canada: 2005, National Research Council Canada.

Stone, N: 1998, Windows and Environmental Cues on Performance and Mood, Environment and Behavior, 30(3), pp. 306-321.

Turner, A. et al: 2001, From Isovists to Visibility Graphs:

A Methodology for the Analysis of Architectural Space, Environment and Planning B, 28, pp. 103-121.