# Sortal grammars: A framework for exploring grammar formalisms

**Rudi Stouffs** 

Faculty of Architecture, Delft University of Technology, Delft, The Netherlands

#### Ramesh Krishnamurti

Department of Architecture, Carnegie Mellon University, Pittsburgh, USA

Grammar formalisms come in a large variety and are commonly difficult to implement. To alleviate these obstacles to a more widespread adoption, a framework for developing grammar systems is needed that supports an exploration of alternate and varying grammar formalisms in a rapid prototyping way. We present such a framework based on a formalism for representational flexibility, named *sorts*. Sorts provides a component-based approach for building grammar systems, utilizing a uniform characterization of grammars.

### **1 INTRODUCTION**

Grammar formalisms have been around for over 40 years and are found in a wide variety of disciplines and domains, to name a few, natural language, architectural design, mechanical design, and syntactic pattern recognition. Their implementations, however, have been mostly narrow-focused and sparse. In design, in particular, the expectation of grammar formalisms or similar rule-based systems to pervade design software has so far remained only an illusion. There are three main reasons for this. The first relates to the difficulty stemming from technical considerations of implementing grammars, which we addressed in an earlier paper (1). The second difficulty pertains to ways of enabling designers to employ grammatical rules in a manner that does not impede their act of designing. In this paper, we consider a third difficulty that affects the rapid development, adaptation, and maintenance of grammar-based systems.

Grammar formalisms come in a large variety, requiring different representations of the objects being generated, and different interpretative mechanisms for this generation. Altering the representation may necessitate a rewrite of the interpretative mechanism, resulting in a redevelopment of the entire system. At the same time, all grammars share certain definitions and characteristics. Grammars are defined over an algebra of objects that is closed under the operations of addition and subtraction and a set of transformations. In addition, a match relation on the algebra governs when an object occurs in another object under some transformation.

Building on these commonalities, we propose a framework for exploring different grammar formalisms, based on a variety of algebras and match relations (or interpretative mechanisms).

We base this framework on *sorts*, a concept for representational flexibility (2)(3). Sorts constitute a model for representations that defines formal operations on sorts and recognizes formal relationships between sorts. Each sort defines an algebra over its elements; formal compositions of sorts derive their algebraic properties from their component sorts. This algebraic framework makes sorts particularly suited for defining grammar formalisms. Furthermore, since sorts can easily be adapted and compared, these provide the basis for exploring alternate and varying grammar formalisms.

# **2 SORTS**

Conceptually, a sort specifies a set of similar models; sorts combine to form new sorts under algebraic operations defined over sorts (2). In practice, elementary data types define *primitive* sorts, which combine to *composite* sorts under formal compositional operations defined over sorts (3). For instance, an attribute operator provides for (recursively) subordinate compositions of sorts using an object-attribute relationship, in both a one-to-one and a one-to-many instantiation. For example, the sort of labeled points is specified as a sort of points, with one or more labels assigned as attribute to each point in the data form. The operation of sum allows for disjunctively coordinate compositions of multiple sorts, under many-to-one and many-to-many instantiations, where each sort may–but does not have to–be represented in the data form. As an example, a rule has both a *lhs* (left-hand-side) and *rhs* (right-hand-side) component, either of which can be omitted. Other compositional operations can also be considered, such as an array- or grid-like composition of sorts.

The definition of a sort also includes a specification of the operational behavior of collections, denoted as *forms*, of its members, denoted as *individuals*, for common arithmetic operations. This behavioral specification enables a uniform handling of forms of different sorts, on the sole condition that the universe of all forms of a sort is closed under the respective operations. Additionally, if a match relation exists that is a partial order relation on the sort's forms, a grammar can be defined over this sort. The simplest behavior that fulfills these requirements is a discrete behavior, corresponding to a mathematical set, where the part relation reduces to the subset relation and the operations of addition and subtraction correspond to set union and difference, respectively. A contrasting behavior offers the maximal element representation (4)(5), where any element or individual contains infinitely many, not necessarily disjoint, individuals. This behavior applies readily to intervals over continuous domains, e.g., line or plane segments, or volumes. Primitive sorts have their behaviors assigned in order to achieve a desired effect, e.g., discrete behaviors for points and labels, an interval behavior for line segments, and an ordinal behavior for weights such as thicknesses or tones. On the other hand, a composite sort receives its behavior from its component sorts, based on its compositional relationship (3).

# **3 SORTAL GRAMMARS**

Grammars are formal devices for specifying languages. A grammar defines a language as the set of all objects generated by the grammar, where each generation starts with an initial object and uses rules to achieve an object that contains only elements from a terminal vocabulary. A rewriting rule has the form  $lhs \rightarrow rhs$ , and applies to a particular object if the *lhs* of the rule 'matches' a part of the object under some allowable transformation. Rule application consists of replacing the matching part by the *rhs* of the rule under the same transformation.

The central problem in implementing grammars is the *matching problem*, that of determining the transformation under which the match relation holds for an *lhs*. Clearly, this problem depends on the representation of the elements of the algebra. Sorts offer a representational flexibility where each sort additionally specifies its own match relation as a part of its behavior. As composite sorts derive their behavior from their component sorts, the technical difficulties of implementing the matching problem only apply once for each primitive sort. New primitive sorts can be developed, distributed, and adopted by other users without any need for reconfiguring the system. At the same time, the appropriateness of a given grammar formalism for a given problem can easily be tested, the formalism correspondingly adapted, and existing grammar formalisms can be modified to cater for changing requirements or preferences.

# **4 EXAMPLES**

A uniform characterization for a variety of grammar systems is given in (6). Krishnamurti and Stouffs (1) survey a variety of spatial grammar formalisms from an implementation standpoint. Here, we consider the specification of some of these examples using sorts.

### 4.1 Structure grammars

Structure grammar is an example of a set grammar (7). A *structure* is represented as a set of pairs, each consisting of a symbol, e.g., a spatial icon, and a transformation. The resulting algebra corresponds to the Cartesian product of the respective algebras for the set of symbols and the group of transformations. Both symbols and transformations define sorts with discrete behavior, i.e., respective sets match under the subset relationship. These combine into a composite sort under the attribute relationship; each symbol in a set may have one or more transformations assigned as attribute.

# 4.2 Tartan Worlds

The *Tartan Worlds* (8) is a spatial grammar formalism that bestrides string and set grammars. We consider a simplified string grammar version of the *Tartan Worlds*: each symbol in a string corresponds to a geometrical entity represented as a graphical icon and located on a grid. A rule in this *Simple Tartan Worlds* (1) consists of one symbol on the *lhs* and symbols on the *rhs* given in their spatial relation. An equivalent sortal grammar may be defined over a sort composed over a grid of a sort of graphical icons. On a fixed-sized grid, the behavior of the composite sort breaks down into the behavior of the sort of graphical icons, e.g., ordinal or discrete, over each grid cell. The matching relation is defined in the same way.

# **4.3** Augmented shape grammars

A *shape* (9) is defined as a finite arrangement of spatial elements from among points, lines, planes, or volumes, of limited but non-zero measure. A shape is a part of another shape if it is embedded in the other shape as a smaller or equal element; shapes adhere to the maximal element representation (4)(5). Shapes of the same dimensionality belong to the same algebra, these define a sort. A shape consisting of more than one type of spatial element types. The algebra given by the Cartesian product of the algebras of its spatial element types. The respective sorts combine under the operation of sum, as a disjunctive composition.

A shape can be augmented by distinguishing spatial elements, e.g., by labeling, weighting, or coloring these elements. Augmented shapes also specify an algebra as a Cartesian product of the respective shape algebra and the algebra of the distinguishing attributes. However, the

resulting behavior can better be expressed with a sort that is a subordinate composition of the respective sorts, i.e., combined under the attribute operator. A sort of labels may adhere to a discrete behavior, a sort of weights to an ordinal behavior; a weight matches another weight if it has a smaller or equal value.

# **5 CONCLUSION**

Technical considerations make grammar systems generally difficult to implement. Part of this difficulty also relates to the appropriateness of a given grammar formalism for a given design problem, and to the representational demands that grammar systems impose on users and developers. Together, these difficulties inhibit the rapid development, adaptation, and maintenance of grammar-based systems. Adopting an existing grammar system may present the user with a system that is not exactly suited for the purpose. On the other hand, it is unreasonable to expect every user to develop a grammar system from scratch or invest the time to analyze and adapt an existing system. Instead, a development environment for grammar systems based on sorts will provide the user with the ability to define a grammar formalism and explore its appropriateness for the problem at hand, then, to integrate it into a larger application.

### REFERENCES

(1) R. Krishnamurti and R. Stouffs, Spatial grammars: motivation, comparison and new results, *CAAD Futures '93* (eds. U. Flemming and S. Van Wyk), pp. 57-74, North-Holland, Amsterdam, 1993.

(2) R. Stouffs and R. Krishnamurti, An algebraic approach to comparing representations, *Mathematics & Design 98* (ed. J. Barallo), pp. 105-114, The University of the Basque Country, San Sebastian, Spain, 1998.

(3) R. Stouffs and R. Krishnamurti, Sorts: A concept for representational flexibility, *CAAD Futures 1997* (ed. R. Junge), pp. 553-564, Kluwer Academic, Dordrecht, The Netherlands, 1997.

(4) R. Krishnamurti, The maximal representation of a shape, *Environment and Planning B: Planning and Design* 19 (1992), 585-603.

(5) R. Stouffs, 1994. The algebra of shapes, Ph.D. dissertation, Department of Architecture, Carnegie Mellon University, 1994.

(6) J. Gips and G. Stiny, Production systems and grammars: a uniform characterization, *Environment and Planning B: Planning and Design* 7 (1980), 399-408.

(7) C. Carlson, R. McKelvey and R. Woodbury, Introduction to structure and structure grammars, *Environment and Planning B: Planning and Design* 16 (1989), 215-242.

(8) R. Woodbury, A. Radford, P. Taplin and S. Coppins, Tartan worlds: a generative symbol grammar system, *ACADIA '92* (eds. D. Noble and K. Kensek), 1992.

(9) G. Stiny, Introduction to shape and shape grammars, *Environment and Planning B: Planning and Design* 7 (1980), 343-351.