S. Chien, S. Choo, M. A. Schnabel, W. Nakapan, M. J. Kim, S. Roudavski (eds.), *Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2016,* 115–124. © 2016, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

THE COMPUTATION OF DESCRIPTION GRAMMARS

Two case studies

RUDI STOUFFS
National University of Singapore, Singapore
Delft University of Technology, Delft, The Netherlands
stouffs@nus.edu.sg; r.m.f.a.stouffs@tudelft.nl

Abstract. We revisit two case studies that adopt a shape grammar to relate different architectural styles. Both adopt a description scheme, augmenting the shape grammar, as the main vehicle for relating different styles, however, they both present the description rules only conceptually. Following a description grammar interpreter and its notation for descriptions and description rules, we explore a valid explication of both description schemes. This exploration serves three purposes: firstly, as a demonstration of the notation adopted; secondly, as an evaluation of the applicability of the description grammar interpreter and its notation to these case studies; and, thirdly, as a demonstration of the explication of description grammars from concept to computation.

Keywords. Shape grammars; description grammars; architectural styles; description.

1. Introduction

A shape grammar is a formal device for producing a language of shapes, e.g., corresponding to an architectural style. In this paper, we consider two case studies that adopt a shape grammar to relate, possibly, different architectural styles. Al-Kazzaz (2011; also, Al-Kazzaz et al, 2010) considers a shape grammar for hybrid design reflecting on a heterogeneous corpus of antecedents, specifically, traditional minaret designs. Ahmad (2009; also, Ahmad and Chase, 2006) considers the transformation of a shape grammar encoding an existing style in response to a change in design style requirements, applied to Greek temple façades and mobile phone designs. Both Al-Kazzaz and Ahmad use description grammars, augmenting the shape grammar, as the main vehicle for relating different styles. In both cases, the description schemes are conceptually developed and only partially explicated; quite a few of the details remain uncovered.

In this paper, we revisit both case studies and suggest an explication of the description schemes. Specifically, we explore the embedding of the detailed computational processes of Ahmad's style mapping and of Al-Kazzaz's user guide and evaluation metrics within the notation adopted for descriptions and description rules by a description grammar interpreter (Stouffs, 2015). This exploration serves three purposes: firstly, as a demonstration of the notation for descriptions and description rules; secondly, as an evaluation of the generality of the notation and applicability of the description grammar interpreter to these case studies; and, thirdly, as a demonstration of the explication of description grammars from concept to computation.

2. Ahmad's style mapping

Ahmad (2009) proposes a style description scheme based on the concept of semantic differential to map the style characteristics of shape rules. These characteristics are specified as numeric values quantifying opposing adjectival pairs for each shape rule. The values are collected through rule application and analysed to characterize the style or styles of the design and of the language of designs as generated by the grammar. The mapping of the style range of the grammar may serve as a guide for grammar transformation. Ahmad presents two exemplar grammars, one for Greek temple façades and one for mobile phone designs; we limit our study to the first example.

The Greek temple façade grammar specifies 5 rule sets. Composition rule sets B and C consider style descriptor ranks reflecting on spatial relations between primitive shape elements: *Symmetric—Asymmetric*, *Monolithic—Fragmentary*, and *Stable—Directional*. Specification rule sets D, E and F consider style descriptor ranks reflecting on the primitive shape elements themselves: *Rectilinear—Curvilinear*, *Symmetric—Asymmetric*, and *Simple—Detailed*. Each rule in these sets specifies a shape transformation, a rank for each relevant style descriptor, and a weight. The style descriptor rank is specified both as a numeric, either 1 or −1, and an alphanumeric value. For example, in the case of style descriptor *Symmetric—Asymmetric*, the rank is either 1, "Symmetric", or −1, "Asymmetric". Derived designs collect style descriptor ranks for each rule applied, also considering the weight of the rule. Based on this collection of style descriptor ranks, the final design is ranked according to the style descriptors *Unity—Diversity*, *Balanced—Unbalanced*, *Simple—Complex*, and *Dominance*.

Ahmad presents a number of style analysis examples of student designs. Design 15 (Figure 1) applies composition rules B4, C3, and specification D15, E7, F18 and F1. The rules have different weights, either 1, 2 (D15) or 3 (E7). The rules define style descriptor ranks that can easily be collected as positive and negative values, separately. Ahmad (2009, pp. 174–175 (Tables 49–53)) separately presents the conditions for the definitions of the style descriptors for derived designs, i.e., *Unity—Diversity*, *Balanced—Unbalanced*, *Simple—Complex*, and *Dominance*, as well as a *similarity* concept. Each style descriptor takes one of three values, for example, the style descriptor *Unity—Diversity* takes the values "Unity" (if all the descriptors are "Similar" according to the *similarity* concept), "Diversity" (if all the descriptors are "Dissimilar" according to the *similarity* concept), and "Partly unified and partly diversified" (otherwise). The condition for design elements (or their spatial relations) to be "Similar" is that "three-fourths or more design elements in a derivation have either positive or negative values for a style descriptor" (Ahmad, 2009, p. 174 (Table 49)). Otherwise they are considered "Dissimilar". We refer to Ahmad (2009, p. 175 (Tables 51–53)) for the conditions for the other style descriptors.

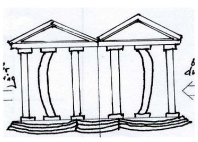
	Rule	Frequency	Weight	Total Weight	Descriptor	Value	Descriptor	Value	Descriptor	Value
Design 15 (Student Q)	**Column spacing rules**									
	A1.1	1								
	A1.2	2								
	A1.3	2								
	Composition rules									
	B4	1	1	1	Symmetric	-1	Fragmentary	1	Stable	-1
	C3	1	1	1	Symmetric	-1	Fragmentary	1	Stable	-1
	Total weight of positives					0		2		0
	Total weight of negatives					2		0		2
	Similarity				Similar		Similar		Similar	
	Specification rules									
	D15	3	2	6	Curvilinear	1	Symmetric	-1	Detailed	1
	E7	2	3	6	Rectilinear	-1	Symmetric	-1	Detailed	1
	F18	2	1	2	Curvilinear	1	Asymmetric	1	Detailed	1
	F1	4	1	4	Rectilinear	-1	Symmetric	-1	Simple	-1
	Total weight of positives					8		2		14
	Total weight of negatives					10		16		4
	Similarity				Dissimilar		Similar		Similar	
	Design style description									
	Partly rectilinear and partly curvilinear									
	High degree of balance									
	High degree of complexity									

Figure 1. Design 15 (taken with permission from Ahmad, 2009, p. 182).

Table 1 presents description rules that explicate this description scheme. Only the rules that are necessary for the derivation (Table 2) of design 15 are included in Table 1. Four descriptions are considered: `spatial_relations`, `primitives`, `design_style`, and `similarity`.

Table 1. Description grammar rules used for design 15.

	design_style + primitives + similarity + spatial_relations
B4	Place three base markers symmetrically below the exterior columns spatial_relations: { `<sym, asy> <mon, fra> <sta, dir>` } → { `<sym - 1, asy> <mon, fra + 1> <sta - 1, dir>` }
C3	Place two pediment markers on top of the exterior columns spatial_relations: { `<sym, asy> <mon, fra> <sta, dir>` } → { `<sym - 1, asy> <mon, fra + 1> <sta - 1, dir>` }
D15	Apply base design 15 primitives: { `<rec, cur> <sym, asy> <sim, det>` } → { `<rec, cur + 2> <sym - 2, asy> <sim, det + 2>` }
E7	Apply pediment design 7 primitives: { `<rec, cur> <sym, asy> <sim, det>` } → { `<rec - 3, cur> <sym - 3, asy> <sim, det + 3>` }
F1	Apply column design 1 primitives: { `<rec, cur> <sym, asy> <sim, det>` } → { `<rec - 1, cur> <sym - 1, asy> <sim - 1, det>` }
F18	Apply column design 18 primitives: { `<rec, cur> <sym, asy> <sim, det>` } → { `<rec, cur + 1> <sym, asy + 1> <sim, det + 1>` }
G1a	Mark the design elements as Similar for the spatial relations descriptor Symmetric-Asymmetric similarity: { `<s1, s2, s3> prims` } → { `<"Similar", s2, s3> prims` } spatial_relations: { `<sym, asy?<=(sym / -3)> <mon, fra> <sta, dir>` } → { `<sym, asy> <mon, fra> <sta, dir>` }
G2b	Mark the design elements as Similar for the spatial relations descriptor Monolithic-Fragmentary similarity: { `<s1, s2, s3> prims` } → { `<s1, "Similar", s3> prims` } spatial_relations: { `<sym, asy> <mon, fra?>=(mon * -3)> <sta, dir>` } → { `<sym, asy> <mon, fra> <sta, dir>` }
G3a	Mark the design elements as Similar for the spatial relations descriptor Symmetric-Asymmetric similarity: { `<s1, s2, s3> prims` } → { `<s1, s2, "Similar"> prims` } spatial_relations: { `<sym, asy?<=(sym / -3)> <mon, fra> <sta, dir>` } → { `<sym, asy> <mon, fra> <sta, dir>` }
G5a	Mark the design elements as Similar for the primitives descriptor Symmetric-Asymmetric primitives: { `<rec, cur> <sym, asy?<=(sym / -3)> <sim, det>` } → { `<rec, cur> <sym, asy> <sim, det>` } similarity: { `rels <s1, s2, s3>` } → { `rels <s1, "Similar", s3>` }
G6b	Mark the design elements as Similar for the primitives descriptor Simple-Detailed primitives: { `<rec, cur> <sym, asy> <sim, det?>=(sym * -3)>` } → { `<rec, cur> <sym, asy> <sim, det>` } similarity: { `rels <s1, s2, s3>` } → { `rels <s1, s2, "Similar">` }
G9	Specify a high degree of balance design_style: { `unidiv balunb simcom domina` } → { `unidiv "High degree of balance" simcom domina` } spatial_relations: { `<sym?<0, asy?=0> <mon, fra> <sta, dir>` } → { `<sym, asy> <mon, fra> <sta, dir>` }
G12	Specify a high degree of complexity design_style: { `unidiv balunb simcom domina` } → { `unidiv balunb "High degree of complexity" domina` } similarity: { `<rec, cur> <sym, asy> <sim, det?>(sim * -3)>` } → { `<rec, cur> <sym, asy> <sim, det>` }

Table 2. Derivation of the descriptions for design 15.

	design_style: { `"Partly unified and partly diversified" "Partly balanced and partly unbalanced" "Partly simple and partly complex" "Partly rectilinear and partly curvilinear"` } primitives: { `<0, 0> <0, 0> <0, 0>` } similarity: { `<"Dissimilar", "Dissimilar", "Dissimilar"> <"Dissimilar", "Dissimilar", "Dissimilar">` } spatial_relations: { `<0, 0> <0, 0> <0, 0>` }
B4, C3, D15, D15, D15, E7, E7, F18, F18, F1, F1, F1, F1	design_style: { `"Partly unified and partly diversified" "Partly balanced and partly unbalanced" "Partly simple and partly complex" "Partly rectilinear and partly curvilinear"` } primitives: { `<-10, 8> <-16, 2> <-4, 14>` } similarity: { `<"Dissimilar", "Dissimilar", "Dissimilar"> <"Dissimilar", "Dissimilar", "Dissimilar">` } spatial_relations: { `<-2, 0> <0, 2> <-2, 0>` }
G1a, G2b, G3a, G5a, G6b, G9, G12	design_style: { `"Partly unified and partly diversified" "High degree of balance" "High degree of complexity" "Partly rectilinear and partly curvilinear"` } primitives: { `<-10, 8> <-16, 2> <-4, 14>` } similarity: { `<"Similar", "Similar", "Similar"> <"Dissimilar", "Similar", "Similar">` } spatial_relations: { `<-2, 0> <0, 2> <-2, 0>` }

The description `spatial_relations` collects style descriptor values reflecting on the spatial relations between primitive shape elements: *Symmetric—Asymmetric*, *Monolithic—Fragmentary*, and *Stable—Directional*, in this order. The descriptor values are grouped in pairs, each enclosed with angle brackets. Similarly, the description `primitives` includes three pairs of style descriptor values reflecting on the primitive shape elements themselves: *Rectilinear—Curvilinear*, *Symmetric—Asymmetric*, and *Simple—Detailed*, in this order. The weight of each rule is encoded in the computation of the rule: the weight value is added or subtracted from the relevant descriptor values (e.g., rules D15 and E7).

The description `design_style` ranks the final design according to the style descriptors *Unity—Diversity*, *Balanced—Unbalanced*, *Simple—Complex*, and *Dominance*, in this order. Rules G7 through G14, in pairs, encode the respective conditions and identify the appropriate design style descriptions, as alphanumeric values. For example, rule G7 checks whether there is a high degree of unity and rule G8 whether there is a high degree of diversity. If neither applies, then the default description "Partly unified and partly diversified" is retained. Rules G11 and G12 rely upon the final description `similarity`, ranking the design as similar or dissimilar for each of the three spatial relations' descriptor ranks and primitives' descriptor ranks. Therefore, this description is composed of two triples—each triple enclosed with angle brackets—of the terms "Similar" and "Dissimilar" applying to the descriptor ranks in the order as specified within the descriptions `spatial_relations` and `primitives`, respectively. Rules G1 through G6 encode the similarity conditions and identify the appropriate term for each descriptor rank. Each rule has two variants, one for positive values and one for negative values. If neither variant applies, then the default term "Dissimilar" is assumed.

3. Al-Kazzaz's user guide grammar and evaluation metrics

Al-Kazzaz (2011) considers descriptions in shape grammars for hybrid design, where the descriptions provide feedback on rule application based on comparisons between the generated design and the antecedents in the corpus. One description scheme acts as a user guide for hybrid design and is specified as a set of antecedent labels. In order to ensure that subsequent shape rule applications are taken from different antecedents, each shape rule requires the label of its antecedent to be part of the user guide and subsequently removes the same label from the user guide.

Al-Kazzaz considers two more description schemes specifying evaluation metrics for rules and derivations, respectively. These provide feedback on the degree of innovation in hybrid design, specifically, whether the resulting design combines features from different antecedents and whether the design is sufficiently different from the antecedents in the corpus (Al-Kazzaz, 2011, p. 73). The evaluation metrics provide feedback both on the rule under application and on the design currently being derived. At the rule level, Al-Kazzaz distinguishes a rule prevalence value, a rule geometrical difference value, and a rule sequential difference value. The rule prevalence value depends on the number of antecedents the rule is derived from and, thus, the number of antecedent labels of the rule (divided by the total number of antecedents in the corpus). The other two values depend on the number of antecedents defining rules that share the same left-hand-side as the current rule while sharing the same geometry or sequence (Al-Kazzaz, 2011, p. 73–75).

At the derivation level, Al-Kazzaz distinguishes the following metrics:

- Design diversity is the number of antecedents that the generated design is derived from, divided by the total number of antecedents in the corpus;
- Design abundance is the cumulative sum of the number of antecedents in each of the applied rules, divided by the number of applied rules;
- The matching degree equals the highest number of applied rules derived from a same antecedent, divided by the number of applied rules;
- The design geometrical difference is the sum of rule geometrical difference values of the applied rules, divided by the number of applied rules;

- The design sequential difference is the sum of rule sequential difference values of the applied rules, divided by the number of applied rules

Al-Kazzaz's hybrid minaret design grammar considers a corpus of 12 antecedents, identified as d1 through d12. Each rule in the minaret design grammar identifies the antecedents this rule is derived from. For example, rule OR7a applies to the design of antecedent d12 (only) and, therefore, has the antecedent label "d12". In order to ensure hybrid minaret designs, Al-Kazzaz stipulates that a rule only applies if at least one antecedent label is present in the user guide, and that all antecedent labels of the rule are removed from the user guide upon application of the rule. In the case of rule OR7a, a description rule { `"d12"` } → { } may be considered that ensures the presence of the label "d12" in the user guide, and subsequently removes it from the user guide. Unfortunately, this type of rule does not apply for more than one antecedent if some, but not all, antecedent labels are already absent from the user guide. Instead, we consider a description a_user_guide containing a list of numbers of applied rules, one for each antecedent d1 through d12. Thus, a rule applies if the number for applied rules is 0 for at least one antecedent, and the number of applied rules is increased by one for each antecedent of the rule. Expressing this condition requires a rule variant for each antecedent of the rule.

Table 3 explicates description rules for the first four rules of an example presented by Al-Kazzaz (2011, pp. 131–137) for a hybrid design derived using seven (original) rules. For the last rule, only one variant is presented. Table 4 presents the resulting derivation. A separate description is considered for each metric at the derivation level: diversity, abundance, matching_degree, geometrical, and sequential. They receive their values from other descriptions. The value of diversity is computed from an antecedents description that maintains a list of (unique) antecedents from which rules have been applied. The values of abundance, geometrical and sequential are computed from a design_values description, which contains a list of three values: the cumulative sum of the number of antecedents in each of the applied rules, the sum of rule geometrical difference values of the applied rules, and the sum of rule sequential difference values of the applied rules. The value of matching_degree is computed from a_user_guide, it is its maximum value divided by the number of applied rules. A description rules maintains the number of applied rules and a description rule_values contains the evaluation metrics at rule level.

We refer to Stouffs (2015) for any details on the notation used but would like to point out the following:

- Tuples are enclosed with angle brackets.

- `length`, `unique`, and `max` are functions operating on tuples, where `length` returns the number of elements in the tuple, `unique` returns a new tuple with duplicates removed, and `max` returns the maximum numeric value in a tuple.
- The append operator on a tuple is implicit, e.g., in `(da "d12")`, the alphanumeric value `"d12"` is appended to the tuple `da`.
- The literal '`e`' defines an 'empty' entity, that is, zero, an empty string, or an empty tuple.
- A reference to the value of another description takes the form `<description>.value` for the value before rule application and `rhs.<description>.value` for the value after rule application. In the latter case, the description name should precede the current description name alphanumerically, to account for the order in which they are processed.
- A reference to a parameter of another description rule takes the form `<description>.<parameter>`.

Table 3. Four description grammar rules for a hybrid minaret using 7 original rules.

a_user_guide + abundance + antecedents + corpus + design_values + diversity + geometrical + matching_degree + sequential + rule_values + rules	
OR7a	Add a circular minaret base
a_user_guide: { `<d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, 0>` } → { `<d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, 1>` } abundance: { `da` } → { `(design_values.sda + 1) / (rules.n + 1)` } antecedents: { `da` } → { `unique(da "d12")` } design_values: { `sda, tgdv, tsdv` } → { `sda + 1, tgdv + 1 - 1 / corpus.value, tsdv + 1 - 6 / corpus.value` } diversity: { `dd` } → { `length(rhs.antecedents.value) / corpus.value` } geometrical: { `gd` } → { `rhs.design_values.tgdv / (rules.n + 1)` } matching_degree: { `md` } → { `max(rhs.a_user_guide.value) / (rules.n + 1)` } sequential: { `sd` } → { ` rhs.design_values.tsdv / (rules.n + 1)` } rule_values: { `rpv, rgdv, rsdv` } → { `1 / corpus.value, 1 - 1 / corpus.value, 1 - 6 / corpus.value` } rules: { `n` } → { `n + 1` }	
OR4b	Add an octagonal minaret body above the base
a_user_guide: { `<d1, d2, 0, d4, d5, d6, d7, d8, d9, d10, d11, d12>` } → { `<d1, d2, 1, d4, d5, d6, d7, d8, d9, d10, d11, d12>` } abundance: { `da` } → { `(design_values.sda + 1) / (rules.n + 1)` } antecedents: { `da` } → { `unique(da "d3")` } design_values: { `sda, tgdv, tsdv` } → { `sda + 1, tgdv + 1 - 1 / corpus.value, tsdv + 1 - 6 / corpus.value` } diversity: { `dd` } → { `length(rhs.antecedents.value) / corpus.value` } geometrical: { `gd` } → { `rhs.design_values.tgdv / (rules.n + 1)` } matching_degree: { `md` } → { `max(rhs.a_user_guide.value) / (rules.n + 1)` } sequential: { `sd` } → { ` rhs.design_values.tsdv / (rules.n + 1)` } rule_values: { `rpv, rgdv, rsdv` } → { `1 / corpus.value, 1 - 1 / corpus.value, 1 - 6 / corpus.value` } rules: { `n` } → { `n + 1` }	
OR12b	Add an octagonal second minaret body above the first body

	a_user_guide: { `<d1, d2, d3, d4, d5, d6, d7, d8, 0, d10, d11, d12>` } → { `<d1, d2, d3, d4, d5, d6, d7, d8, 1, d10, d11, d12>` } abundance: { `da` } → { `(design_values.sda + 1) / (rules.n + 1)` } antecedents: { `da` } → { `unique(da "d9")` } design_values: { `sda, tgdv, tsdv` } → { `sda + 1, tgdv + 1 - 5 / corpus.value, tsdv + 1 - 1 / corpus.value` } diversity: { `dd` } → { `length(rhs.antecedents.value) / corpus.value` } geometrical: { `gd` } → { `rhs.design_values.tgdv / (rules.n + 1)` } matching_degree: { `md` } → { `max(rhs.a_user_guide.value) / (rules.n + 1)` } sequential: { `sd` } → { ` rhs.design_values.tsdv / (rules.n + 1)` } rule_values: { `rpv, rgdv, rsdv` } → { `1 / corpus.value, 1 - 5 / corpus.value, 1 - 1 / corpus.value` } rules: { `n` } → { `n + 1` }
OR4d_2	Add a circular balcony above the second body
	a_user_guide: { `<d1, d2, d3, d4, d5, 0, d7, d8, d9, d10, d11, d12>` } → { `<d1, d2, d3 + 1, d4, d5, 1, d7 + 1, d8 + 1, d9, d10, d11, d12>` } abundance: { `da` } → { `(design_values.sda + 4) / (rules.n + 1)` } antecedents: { `da` } → { `unique(da "d3" "d6" "d7" "d8")` } design_values: { `sda, tgdv, tsdv` } → { `sda + 4, tgdv + 1 - 5 / corpus.value, tsdv + 1 - 6 / corpus.value` } diversity: { `dd` } → { `length(rhs.antecedents.value) / corpus.value` } geometrical: { `gd` } → { `rhs.design_values.tgdv / (rules.n + 1)` } matching_degree: { `md` } → { `max(rhs.a_user_guide.value) / (rules.n + 1)` } sequential: { `sd` } → { ` rhs.design_values.tsdv / (rules.n + 1)` } rule_values: { `rpv, rgdv, rsdv` } → { `4 / corpus.value, 1 - 5 / corpus.value, 1 - 6 / corpus.value` } rules: { `n` } → { `n + 1` }

Table 4. Partial derivation of the descriptions for a hybrid minaret using 7 original rules.

	a_user_guide: { `<0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0>` } abundance: { `e` } antecedents: { `e` } corpus: { 12 } design_values: { `0, 0, 0` } diversity: { `e` } geometrical: { `e` } matching_degree: { `e` } sequential: { `e` } rule_values: { `e, e, e` } rules: { `0` }
OR7a	a_user_guide: { `<0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1>` } abundance: { `1.0` } antecedents: { `"d12"` } corpus: { 12 } design_values: { `1, 0.917, 0.5` } diversity: { `0.083` } matching_degree: { `1` } geometrical: { `0.917` } sequential: { `0.5` } rule_values: { `0.083, 0.917, 0.5` } rules: { `1` }
OR4b, OR12b, OR4d_2	a_user_guide: { `<0, 1, 2, 1, 0, 1, 1, 2, 1, 0, 1, 1>` } abundance: { `1.75` } antecedents: { `"d12" "d3" "d9" "d6" "d7" "d8" "d4" "d11" "d2"` } corpus: { 12 } design_values: { `7, 3.0, 2.417` } diversity: { `0.5` } geometrical: { `0.75` } matching_degree: { `0.5` } sequential: { `0.604` } rule_values: { `0.333, 0.583, 0.5` } rules: { `4` }

4. Discussion

Both case studies only present one possible explication. For example, in the case of Ahmad's style mapping, we chose to include only the numeric values for the style descriptor ranks and collect these ranks into a `spatial_relations` and a `primitives` descriptor. Instead we could have opted to include the alphanumeric values for the style descriptor ranks alongside the numeric values and maintain a separate description for each style descriptor rank. The latter may better reflect on the way Ahmad chooses to present the results (Figure 1) but is far more verbose. In the case of Al-Kazzaz's user guide grammar and evaluation metrics, we adopted a number of auxiliary descriptions in order to calculate and derive the different metrics. There is certainly some flexibility in the way these auxiliary descriptions are defined and specified.

5. Conclusion

We have demonstrated for two case studies that the description grammar interpreter (Stouffs, 2015) and its notation for descriptions and description rules are able to support the development and explication of description grammars from concept to computation. Certainly, the given examples present only one manner of explication; others may exist that are as valid. Also, each present only a single derivation; other derivations may require additional rules that are not mere variants of the rules presented. Nevertheless, the case studies do point to the generality of the notation and the description grammar interpreter. We do hope these examples may inspire authors of description schemes to extend their development beyond the conceptual level.

References

Ahmad, S.: 2009, A framework for strategic style change using goal driven grammar transformations, Ph.D. dissertation, Dept. of Architecture, University of Strathclyde, Glasgow, UK.

Ahmad, S. and Chase, S.: 2006, Grammar Representations to Facilitate Style Innovation: With an Example From Mobile Phone Design, *in* V. Bourdakis and D. Charitos (eds.), *Communicating Space(s) - Proceedings of the 24th eCAADe Conference*, Volos, Greece, 320–323

Al-Kazzaz, D. A. A.: 2011, Shape grammars for hybrid component-based design, Ph.D. dissertation, Dept. of Architecture, University of Strathclyde, Glasgow, UK.

Al-Kazzaz, D., Bridges, A. and Chase S.: 2010, Shape Grammars for Innovative Hybrid Typological Design, *in* G. Schmitt at al (eds.), *Future Cities - Proceedings of the 28th eCAADe Conference*, ETH Zurich, 187–195.

Stouffs, R.: 2015, Implementing a description grammar interpreter: a notation for description and description rules, *in* B. Martens, G. Wurzer, T. Grasl, W.E. Lorenz, R. Schaffranek (eds.), *Real Time: Extending the Reach of Computation*, Vol. 1, eCAADe, Brussels, and Vienna University of Technology, Vienna, 471–480.