This paper provides an overview on the relation between computational ontologies and shape grammars regarding the development and production of multi-purpose Semantic Design Systems. The objective of the author's ongoing research is to assist the creation of generative design systems, applicable to design processes in general. Shape grammar rules and ontologies in these systems will be focusing on abstract, generic rules and generic descriptions. When combined through contextually specified relations, these assume semantic expressions and should be able to produce meaningful results. We collect here a short state of the art of the research developed in the fields of architecture, urbanism and computer science in the past ten years regarding the use of knowledge bases (ontologies) combined with generative design systems (with a particular focus on shape grammars). We expect to provide both insight about architectural and urban typologies and the production of meaningful designs using automated generative design systems.

**Keywords:** Ontologies, Shape Grammars, Semantic Design Systems, Architectural Design, Urban Planning

**INTRODUCTION**

The main objective of the research presented in this paper is the production of multi-purpose Semantic Design Systems that may be applicable to design processes in general. The research aims at defining a generative system that, using simple basic shapes and generic grammar rules, when combined with ontologies (adding specific meanings and relations to shapes) may be able to generate meaningful designs in the particular design domain defined by the ontology. In this paper we show the preliminary stage of surveying the present knowledge on the combined use of ontologies and shape grammars.

Generative design systems, namely shape grammars (Stiny and Gips 1971), allow the semi-automatic creation of a large quantity of varied designs. These designs are generated in a procedural fashion, through changes in parameters or the iterative application of a limited set of simple production rules, resulting in varied set of designs which define the grammar’s design space.

Despite being able to generate large quantities of syntactically correct designs, generative design systems do not guarantee the designs are semanti-
cally correct. This leads to the production of unfeasible or undesired solutions.

Semantic accuracy can be achieved through the combination of generative systems with knowledge bases, such as computational ontologies (Gruber 1993). According to Grobler et al. (2008), “the combinatory nature of the design rules captured by shape grammar and ontology offers the possibility that design knowledge can be explicitly represented, maintained and processed.”

In this paper, we will begin by looking at mechanisms introduced over time to control shape grammar derivations, especially those attempting to introduce semantic control over shape rule application. Over time, these mechanisms have been complementing the shape grammar formalism shifting this type of generative systems towards more complex compound grammars. (Knight 2003)

Later, we will move towards the combination of generative design systems based on shape grammars with ontologies. We will be looking at six approaches by different authors regarding both architecture (physical or virtual) and urbanism, over a period of time that spans from 2006 to 2015:

- Shape rules as descriptions for designs, replicated in natural language to create an ontology for a type of buildings (Andaroodi et al. 2006);
- Multi-level projection ontology design approach to create 3D models of Chinese Ancient Architectures (Liu et al. 2006; 2008), using approximation theory of granular computing to enhance its accuracy and performance (Liu et al. 2010). These approaches were later formalised as the Onto-Draw framework (Liu et al. 2012);
- Ontologies working in parallel with shape grammars, informing shape rule application through a lightweight BIM, to characterize existing buildings (Grobler et al. 2008; Aksamija et al. 2010);
- Ontologies defining the relevant concepts for a generative process, shape grammars as design generators, heuristics guiding the generation process and validations as a mechanism to test and evaluate designs (Trescak et al. 2010);
- Ontologies as a common representation protocol between different design modules (Beirão 2009; 2012);
- Ontologies as meta-models for generative design, using description rules to guide shape rule application (Stouffs and Tunçer 2015).

Also worth mentioning is the work of Tutenel et al. (2011), a framework to procedurally create buildings for computer games with consistent interrelated interiors and exteriors. In this case, the knowledge base is represented and stored in a purpose-built relational database, not an ontology.

CONTROLLING SHAPE GRAMMARS

Generative design systems based on shape grammars allow designers to work with large quantities of solutions within a design language, testing multiple outputs in search for the one(s) that meet the desired criteria (Stiny and Gips 1971).

Shape grammars are a formal rewriting mechanism which applies shape rules ($\alpha \rightarrow \beta$) algorithmically replacing a predicate condition, an Euclidean transformation of a shape ($t(\alpha)$), found in a design ($C$), by a consequent shape defined under the same transformation ($t(\beta)$), so that: $C' = C - t(\alpha) + t(\beta)$.

One issue that arises when working with shape grammars is related with the system’s ability to produce valid solutions. This is a particularly important feature in automated systems, where they may produce large quantities of syntactically correct derivations of a shape grammar but which are, in the end, semantically incorrect and therefore unfeasible or undesirable. Therefore, providing generative systems with semantic support tools able to distinguish the semantically robust from the semantically meaningless derivations is a mandatory procedure on any implementation aiming at developing a semantically robust generative design system. The semantic difficulties in relation to shape grammars
have been expressively pointed by Fleisher (1992) even after some of the semantic formalisms developed for shape grammars had already been developed.

To regulate shape grammar derivations, some control mechanisms were introduced over time. From the very beginning, with the definition of shape grammars, labels were introduced to control the iterative application of shape rules (Stiny and Gips 1971). Labels can be used, for instance, as symbols associated with shapes on the left-hand side (α) of a shape rule (α → β), breaking the symmetry group of (α) and, therefore, conditioning the number of possible solutions (comparing with unlabelled shapes) (Stiny 1980). In this case, labels are used as spatial relation markers. Labelled shapes can be used to reduce ambiguity in shape rule application or distinguish shape rules in the construction of different parts of a design. In some situations, labels add a particular classification to a shape. For instance, in Duarte (2001) labels are used to distinguish types of space according to their functions, a "kitchen" as something different from a "bedroom" even though they might be both represented by a rectangle. In this case labels work as classifiers.

Description grammars followed, operating in parallel with shape grammars and constraining the application of shapes and symbols with description rules (Stiny 1981). After that came colours (Knight 1989), weights (Stiny 1992) and discursive grammars (Duarte 2001). Discursive grammars combined shape grammars, description grammars and heuristics in order to give a meaning to the derivation of solutions based on a design brief.

Later, Haldane Liew (2003) proposed a Shape Grammar Meta-Language (SGML), creating layers of abstraction in the shape grammar formalism and developing descriptions that can organize and control the application of rules. The rule application process is expanded into six phases: control, context, transformations, parameters, descriptors and application. Particularly interesting in the SGML approach are the descriptor, the context and the control phases. These three phases allow the conditioning of rule application to shapes that are only composed of maximal lines and/or have void spaces (descriptor), for instance, filtering specific labelled shapes or focusing on a particular shape when applying a rule (context), by linking a series of rules together forming macros or by determining the availability of a set of rules at any point in the grammar derivation (control).

Graph grammars may also be used as a control mechanism, working in parallel with shape grammars to ensure the maintenance of topological relations between shapes (Grasl and Economou 2010; 2013). Graphs can also be used to "group rules according to the topological transformations involved, without considering specific shapes". (Eloy and Duarte 2011)

When the rules in the abovementioned grammars are linked between them, constraining each other's application, they become compound rules and the grammar can be designated as a compound grammar. (Knight 2003)

These control mechanisms impose restrictions to the application of rules in a shape grammar, being able to provide some semantic control - by associating qualitative labels with descriptions, for instance. By using such mechanisms, shape grammars contain knowledge about families of designs and how these can be generated. This knowledge is not, however, explicit or easy to work with, extend or maintain. (Duarte 2001; Beirão 2012).

As we will see in the next section, knowledge regarding families of designs (namely building types and typology) is better represented using knowledge bases such as ontologies.

Ontologies can be described as "an explicit specification of a conceptualization" (Gruber 1993), structuring information using taxonomy (classes and sub-classes: "is a") and meronomy ("part of") relations. This allows ontologies, through the use of reasoners, to infer relations which were not explicitly stated between its elements, according to their object or data properties. Also known as knowledge-based models, ontologies are perfectly suited to contain infor-
Ontologies are particularly suited for well-known and well-defined historical architectural types, and may contain within it many different types of information: either regarding its physical (and geometrical) characteristics, conceptual definitions about the type, etc. The development of an ontology is meant to be a collaborative and adaptive process, changing or redesigning the ontology as new knowledge about a subject is introduced, until consensus about the subject is reached.

Furthermore Stouffs and Krishnamurti (2001) propose a formalism called sorts for representing different data types. Conceptually, elementary data types are used to define primitive sorts and these combined can be used to define composite sorts under formal compositional algebraic operations. As algebraic forms, sorts are particularly fit to define and combine grammars of different types as well as establish explicit relationships between sorts providing therefore semantic representations of concepts in any representation domain.

In this case the goal was not to generate new designs within the type, but to use this schema to extract computer-based semantics of the rules in order to provide a cultural heritage database with enhanced documentation, using ontologies.

**Extensive Semantics System**

Grobler et al. (2008) identified certain similarities between knowledge-based models and shape grammars: both of them contain design rules, though the nature of the rules varies. In their case, ontologies were used in parallel with shape grammars as a way to inform shape rule application, giving it contextual information specifically focused on existing architectural types by capturing knowledge about the architectural type.

In this case, ontologies work as a knowledge representation about a subject, describing individuals as basic objects and organizing collections or types of objects under classes. They also define properties and characteristics and, more importantly, relations between objects.

Based on the requirements, a lightweight Building Information Modelling (BIM) was constructed, containing extensive information about the building. This information (data) was then used as initial input
for the shape rules.

An ontology of architectural design drivers was then used to provide additional data to the shape grammar rules. Including specific information about the building (type, location, culture, environment, structural system and context) the ontology described the relations between its different elements, such as size and dimensions, form, structure, circulation and movement patterns.

Collaborative Multi-Expert Semantics System

In the digital heritage project for ancient architectures in southeast China, Liu et al. (2006; 2008) were required to generate large quantities of 3D models of ancient Chinese buildings. Their approach focused on a semiautomatic modelling process at semantic level by using accumulated domain knowledge extracted from the buildings' architectures. This allows the user to focus on the semantic relations between different components rather than focusing on their geometric details. The modelling system had to be capable of distinguishing different elements and styles among multiple buildings. The system should also be able to generate building parts in the style or structure based on the semantic knowledge extracted from existing buildings. The modelling process in this system can be described as designing an ontology for the mentioned architectures, which could then be used to interpret its styles and structure.

The authors consider a multi-level projection ontology design approach, with three participants involved in ontology design: machine, programmer and architect. Different ontologies are required to accommodate the differences in domain knowledge between the participants. Though the three participants are all represented with the same domain concepts of architecture models, each has their own vocabularies and relationships.

Since all of the rules of the grammar were manually extracted by architects, most of them are empirical and imprecise. This might lead to a highly redundant knowledge library, with the following drawbacks: generation of unreasonable buildings from incorrect rules; generation of hybrids from multiple styles due to rule sharing between different styles; and a decrease in the performance of the modelling system due to redundant and incorrect rules.

To enhance the accuracy and performance of their system, Liu et al. (2010) combined the ontology-design method with a granular computing method. Granular computing simplified the design of the auto-modelling system and increased the hit ratio (which describes the performance of the input knowledge) when generating complex architectures with a similar style structure.

Because the selection of appropriate rules from a large knowledge library can be time consuming, a Domain-Knowledge-based-Heuristic-Selection (DKHS) algorithm was used, pruning the spatial search tree. DKHS was considered to be an effective knowledge-selection algorithm, used to compute the best knowledge subset from another knowledge set.

A style-check module evaluates the generated architecture instance to confirm if it belongs to the desired ontology domain. Since a house generated by the system is produced by the grammar, each house can be represented individually as a sequence of grammar terms - reducing the judging of the style of the house to grammar checking. Components and control rules are described in XML [1] and later verified with a technique based on Document-Type Definition (DTD).

The methods described above were later formalised as the Onto-Draw framework. The Onto-Draw engine is a unified platform and interface which supports that multiple participants design the ontology of a complex problem in visual collaboration. Onto-Draw combines varied domain knowledge from multiple experts from different fields and provides visual feedback from their designs. (Liu et al. 2012)
**Automated Semantically Responsive System**

Trescak, Esteva and Rodriguez (Trescak et al. 2010) define a Virtual World Grammar (VWG) to automatically generate, in real time, virtual auction houses with semantic content. The Virtual World Grammar is a compound grammar consisting of two shape grammars (one for the institution building and its inner activities; the other creates a separate building for each activity), an ontology, validations and a set of heuristics.

The ontology contains concepts related to (a) the description of the activities that will take place in the Virtual World and (b) concepts defining the properties of the Virtual World. When combined with the shape grammar, descriptions of the activities (a) will determine the layout of the Virtual World, while the second (b) will define the properties of the Virtual World elements.

The execution of a shape grammar is tested and evaluated using validations. During the generation process, the validations can be evaluated either after each generation step (step validations), preventing the selection of invalid paths of execution of a shape grammar, or at the end of the generation process (final validations), evaluating the final design.

Heuristics are used to guide the generation of the Virtual Worlds. They decide the order in which to process the elements from the specification and finding possible execution nodes in the execution tree for the selected specification element. Information regarding the generation process is stored in a tree structure, holding execution states which are either defined by a rule or a shape.

Definition and execution of the Virtual World Grammars is done via the Virtual World Builder Toolkit (VWBT), through visual interfaces. The Virtual World Builder Toolkit allows designers to explore different designs based on a shape grammar. The designer of the grammar can browse possible designs or fine tune parts of the grammar to obtain desirable results. (Trescak et al. 2010; 2012)

One interesting aspect of this system is its ability to dynamically react to changes in the specification - that is, changes in the ontology - and automatically regenerate the 3D environment incorporating the new definitions, at runtime.

The workflow is divided into three main parts:

1. Preliminary definition: where the user defines the ontology and the shape grammar;
2. Instance definition: loading the specification, creating and defining all specification and shape grammar elements, as well as specifying mappings between them. Heuristics and evaluations are also introduced here;
3. Execution: browsing random designs and modifying instance parameters to reproduce 2D drafts, followed by its transformation into 3D.

**Purely Semantic System**

Stouffs and Tunçer (2015) take a different approach, using typological descriptions as generative guides for historical architecture. The typology of classical period Ottoman mosques is described by an ontology and description grammars are used as the mechanism to generate an instance of the typology from the ontology. The ontology, represented as an XML document, defines a meta-model that is able to create a model within a domain.

As in previous researches, the physical architectural object is decomposed in a hierarchical manner and its components structured as an ontology. Structural and organizational aspects of the mosques are classified under the "physical" class of the ontology. Perceptual, spatial, philosophical and contextual aspects are classified under the "conceptual" class.

In their case, the ontology prescribes the design descriptions the authors want to arrive at, at least in their hierarchical decomposition. It also drives the rule application process using a compound grammar, constraining the selection of rules that can be applied at any step in the process.

Their compound grammar consists of a description grammar and a shape grammar, with rules each
consisting of one or more description rule components and one or more shape rule components. By combining the description grammar with shape production rules, the compound grammar is able to generate instances of the typology both as an ontological description and as a geometric model.

Stouffs and Tunçer (2015) suggest the use of the term **shape grammar** to denote their compound grammar, in reference to Stiny's (1992) definition, considering their grammar to apply over an algebra that is composed of both shape algebras (extended with labels and/or weights, following its definition), and one or more description algebras.

**Generative with Semantic Control System**

Regarding urban design, an ontology describing street systems was used as a common representation protocol between three interrelated modules of an urban design tool developed in the context of the City Induction project: a formulation module, a generation module and an evaluation module. (Duarte et al. 2012; Beirão 2012; Beirão et al. 2012)

In this tool, the representation of new urban plans is generated by a compound grammar composed of several discursive grammars, each one using in their shape set one of the object classes found in the city ontology. The city ontology describes several complementary representations of a city: the city seen as a street system; the city as a built system; the city as a property system and so on. Each view of the city is described by a taxonomy that hierarchically describes its system branching from object classes describing generic or large concepts and progressively detailing each concept into sub-classes. In a way, each system can be seen as a sub-ontology of the larger ontology describing the city.

An instance in a class is composed of a shape part and a semantic part (a label, a description, or any type of classifier). Depending on the class the shape part or the semantic part may be an empty set.

The compound grammar is built out of smaller grammars in the form of discursive grammars that replicate common urban design operations and are therefore called urban design patterns - they integrate simultaneously the design pattern concept developed by Gamma et al. (1995) and the idea developed by Alexander et al. (1977) of defining languages of design by repeating well known and time tested design operations.

**DISCUSSION**

In the previous sections we looked into six systems for semantic design: **Inductive, Extensive Semantics, Collaborative Multi-Expert Semantics, Automated Semantically Responsive, Purely Semantic and Generative with Semantic control**.

These relate generative design systems (shape grammars) with knowledge bases (ontologies), introducing semantic control in the design generation process. The Inductive System's sole objective is to extract knowledge, as an ontology, from the generative rules of a shape grammar that represents an historical architectural type. The remaining systems take different approaches to the semantic generation of designs, producing meaningful results within a design language. Apart from the **Inductive System**, all systems share characteristics between one another; their identification tries to reflect their most prominent characteristics.

The systems reveal the complexity of the subject of semantic design systems and the difficulties inherent to its computational implementation.

Table 1 presents an overview of these systems, looking into some characteristics that may be used as a reference for future development of semantic design systems. It highlights specific characteristics in each system, such as the ontology language used or the final output of the system.

One interesting thing to notice is a slight preference for the XML language to represent the ontologies, over OWL [2] and/or RDF/S [3][4]. The use of heuristics and validations in the derivation process is widespread in all systems. Heuristics appear to be fundamental to coordinate the semantic application of shape rules, using validations to confirm the semantic accuracy of the derivation.
Table 1
Overview of the six Semantic Design Systems.

<table>
<thead>
<tr>
<th>References</th>
<th>Inductive</th>
<th>Extensive Semantics</th>
<th>Collaborative Multi-Expert Semantics</th>
<th>Automated Semantically Responsive</th>
<th>Purely Semantic</th>
<th>Generative with Semantic Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Implementation</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Unknown</td>
<td>Yes</td>
</tr>
<tr>
<td>Ontology Language</td>
<td>OWL/RDFS</td>
<td>OWL/XML</td>
<td>XML</td>
<td>XML</td>
<td>XML</td>
<td>Natural Language</td>
</tr>
<tr>
<td>Uses Heuristics</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Uses Validations</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Interface</td>
<td>-</td>
<td>PILOT (Thetus Publisher)</td>
<td>Onto-Draw Engine</td>
<td>Virtual World Builder Toolkit</td>
<td>-</td>
<td>Grasshopper 3D (adapted from shape grammars)</td>
</tr>
<tr>
<td>Multiple Ontologies</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Automated Design Generation</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Semi-Automatic</td>
</tr>
<tr>
<td>Public Access to the Grammar(s)</td>
<td>(Andaroodi et al., 2006)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>(Şener and Görgül, 2008)</td>
<td>(Beirão, 2009, 2012)</td>
</tr>
<tr>
<td>Output</td>
<td>Ontology</td>
<td>3D Buildings (BIM)</td>
<td>3D Historical Scenes</td>
<td>3D Virtual Auction Houses</td>
<td>2D Building Plans (Ottoman Mosques)</td>
<td>3D Urban Plans</td>
</tr>
</tbody>
</table>


Most systems rely heavily on knowledge descriptions, which are then used as a control mechanism for shape grammar derivation. Knowledge in the ontologies is also used to restrict parameters in the shape grammars.

Regarding computer implementation, only the Generative with Semantic Control System uses an interface which is familiar in architectural design: the Grasshopper 3D [6] plugin for Rhinoceros [7].

One important topic for future developments of semantic design systems would be the agreement upon the structure of ontologies representing building or urban types and typology. The works presented here already pave the way for (universal) building or urban knowledge descriptions.

CONCLUSION
This paper presented an overview of approaches to Semantic Design Systems in the fields of architecture and urbanism (for both real and virtual environments), more specifically regarding the combination of computational ontologies and shape grammars.

A review of control mechanisms to regulate shape grammar derivation was presented, suggesting the need for more flexible and adaptive knowledge representations than just descriptions. Knowl-
edge bases such as ontologies are a suitable solution to introduce semantic controls in shape grammar derivations.

Following this, six strategies were presented proposing different approaches to the same problem: how generative systems and knowledge-bases may be combined to produce feasible, meaningful results within a design language.

Inversely, we also saw how a generative system (shape grammar) may be used to inform a knowledge base (ontology) about a type of building, helping to its classification. This illustrates the symbiotic relation between knowledge bases and generative design systems.

Pairing ontologies with shape grammars can be used for the (collaborative) creation of structured knowledge about families of designs, as well as a (semi-)automatic mechanism to generate meaningful solutions within a family of designs - the underlying goal of the research.

None of the examples presented in this survey offers a complete solution for the creation of semantic design systems but suggest partial mechanisms to further explore.

ACKNOWLEDGEMENTS
The authors are thankful for the financial support given by Portuguese Foundation for Science and Technology (FCT). This work was partially supported by national funds through FCT with reference TECTON-3D PTDC/EEI-SII/3154/2012.

REFERENCES
Aksamija, A., Yue, K., Kim, H., Grobler, F. and Krishnamurti, R. 2010, 'Integration of knowledge-based and generative systems for building characterization and prediction', AI EDAM, 24, pp. 3-16
Beirão, J., Duarte, J. and Stouffs, R. 2009 'Grammars of designs and grammars for designing - grammar-based patterns for urban design,' CAAD Futures 2009
Grasl, T. and Economou, A. 2010 'Palladian Graphs: Using a graph grammar to automate the Palladian grammar', Proceedings of eCAADe 2010, pp. 275-283
Gruber, T. R. 1993, 'Toward principles for the design of ontologies used for knowledge sharing?', International Journal of Human-Computer Studies, 43, pp. 907-928


Liu, Y., Xu, C., Zhang, Q. and Pan, Y. 2006 'Ontology based semantic modeling for chinese ancient architectures', *Proceedings of the National Conference on Artificial Intelligence*, pp. 1808-1813


Stiny, G. and Gips, J. 1971 'Shape Grammars and the Generative Specification of Painting and Sculpture', *IFIP Congress (2)*


[1] https://www.w3.org/XML/