A MULTI-AGENT EXPERT SYSTEM SHELL FOR SHAPE GRAMMARS

FILIPE SANTOS, JOAQUIM REIS, PEDRO LOPES, ALEXANDRA PAIO, SARA ELOY AND VASCO RATO
Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
filipe.santos@iscte.pt

Abstract. This paper proposes a multi-agent rule based architecture of a computational system for supporting generic work with shape grammars. The key ideas and technologies involved are presented. This computational system is being developed and will give support to an urban and architecture research based on shape grammar theories.

Keywords. Shape grammars; rule based systems; multi agent systems.

1. Introduction

During almost 40 years research in Shape Grammars has been focused in conceptual and theoretical aspects (Stiny and Gips 1972, (Stiny 1980a, 1990, 1992, 2001, Knight 1993) and in analysis (Stiny 1977, Stiny and Mitchell 1978, 1980, Knight 1989, Koning and Eisenberg 1981). Work has also been done in synthesis (Stiny 1980b, Heisserman 1994, Agarwal and Cagan 1998, Duarte 2005) and in developing algorithms for shape manipulation in the rule application processes (Krishnamurti 1980, 1981, 1992, Krishnamurti and Stouffs 1997). Only more recently investigation has been oriented to more practical aspects of computer implementation including interfaces and generic interpreters of shape grammars (Tapia 1999, Liew 2004, McGill and Knight 2004, Li et al. 2009). For an historic and panoramic perspective on shape grammars see (Chau 2004) or (Chase 2010). However, as far as we understand, shape knowledge within these specific design applications is represented in a procedural ad hoc way and is therefore too rigid for further improvements.

We believe that a declarative knowledge-based approach would offer more flexibility to face new design situations and improve shape reasoning capabilities.
This paper proposes Generic Shape Grammars (GSG), a multi-agent rule based expert system shell to computationally support generic work with Shape Grammars. We start by describing the GSG project, with its key ideas and technologies involved and then propose a computational architecture. Finally we mention the Emerg.cities4all project, an on-going research on urbanism and architecture based in shape grammar theories that will have the computational support of GSG and which will also offer an opportunity to test GSG in a practical scope.

2. The GSG project

Within this project we are developing a configurable and extensible computational system tool to support generic work with shape grammars (applicable to different fields, e.g., design, architecture, creative composition, education). This system will provide support to analysis and synthesis of languages of designs, including style extraction and representation into shape grammars, automatic spatial relation driven, parsing/analysis and computer supported generation/synthesis. The main objective is to provide an expert system shell for developing shape grammars based systems, i.e., a tool providing the minimum software machinery to develop such systems and also with a configurable computational architecture allowing application based extensions according to different domain subjects, specialists and users.

Our actual research in GSG is centred on a generic shape grammar interpreter for shape computing, with reasoning capabilities about shapes and shape spatial relations and shape emergence, with an intuitive visual and symbolic interface and with advanced options for more complex work. Of course, the declarative and symbolic representation of shapes plays a key role.

3. Technologies involved and system architecture

The system relies heavily on Artificial Intelligence and Distributed Artificial Intelligence techniques and tools: heuristics and heuristic search, symbolic programming, symbolic knowledge representation and reasoning, logic, Rule Based Systems (RBS; both forward and backward reasoning) (Russell and Norvig 2003) and agents and Multi-Agent Systems (MAS) (Weiss 1999). Human-computer interaction will also be relevant since there are three types of users:

- system specialist: builds and expands the shell;
- shape grammar specialist: with the shell builds a system applied to a specific area;
- common user: applies the system to create solutions in the specific area.
The computational system architecture is depicted in figure 1. There are two core modules, the *geom* and the *rbs* module, and an additional *interface* module. Basically, the *geom* and *rbs* modules together provide the shape computing and reasoning capabilities. The *interface* module provides communication with the user, including facilities for shape and rule input/output and edition, a shape grammar interpreter, and system customisation and extension.

The *interface* has a visual component (a user interface) and a symbolic/programmatic component. The former is built on the latter, which is also available, as an API (Application Programmer’s Interface), for the user, mainly for the system specialist or the shape grammar specialist, in order to allow for system customisation and extension, and to develop specific shape grammar systems tailored to specific needs/applications.

The *geom* module provides operations on shapes (sub-shape and shape sum, difference and intersection and reduction to maximal shape) and shape transformations (translation, rotation, reflection, scale). The *rbs* module provides knowledge based representation and reasoning with shapes, shape spatial relations and shape emergence, and is built on a rule based expert system shell, with its inference engine and a global Knowledge Base. As shape grammars systems are essentially (forward reasoning) rule based production systems (dealing additionally with shape computation and emergence) it seems natural to use the rule based system paradigm here. Within an option for the shape grammar specialist and for complex problems, the GSG architecture borrows also concepts and practice from the MAS paradigm. The MAS paradigm views the problem solving effort as distributed amongst interacting autonomous, intelligent/proactive and/or reactive program modules, or agents. These agents can collaborate with each other in multiple coordinated ways ranging from purely cooperative to purely competitive.

In the GSG computational architecture there are (optionally) groups $G_i$ of agents $a_j$, as shown in figure 1. Basically, these agents are RBSs through which complex shape grammar design problem data, computation and reasoning processes can be distributed. Which groups $G_i$ will exist, or which specialist agents $a_j$ will compose each group, in the system will depend on the application. For instance, in figure 2 we depict a possible instantiation.
of the architecture just described for the Emerg.cities4all case, described in the next section. This multi-agent rule based architecture has already been sketched in some previous work (Reis 2008).

There are two main advantages of using the MAS paradigm here. First, a kind of emergence can happen where unexpected design solutions are generated when autonomous agent interaction in exploring parts of solutions in huge solution spaces comes into play. Second, in practical applications, huge and complex problem data, computations and reasoning processes may be conveniently distributed and modularised by using agents as the main module units. Different agents may be used with different specific roles and expert knowledge in a natural way for different practical aspects, such as: different drawing views (as in compound and parallel grammars), different stages of building a design, different levels of complexity of the language of designs (as in the hierarchy urbanism/architecture/construction), different tasks (as in drawing and colouring in painting), different styles, different roles in a design task (e.g., generating, evaluating and criticising, analysing), different parts of the computational system (e.g., interface and presentation, interpreter and rule execution, shape operations, reasoning with shapes and spatial relations).

4. Emerg.cities4all

The Emerg.Cities4all is a research project being developed to generate urban and architecture designs. The project aims to produce a generative computer-aided planning system for cities in Portuguese-speaking countries in Africa and South America for low-income housing. It uses shape grammars as a descriptive method using GSG as a tool. Emerg.Cities4all will encompass several scales of intervention, ranging from the physical level of the city to the level of building construction. Modular, scalable, adaptable, customisable and affordable urban and housing unit solutions are the design goals. The proposed solutions must be environmentally sustainable and energy efficient, while respecting certain social and cultural qualities in the context of new Portuguese-speaking emerging urbanities. Several studies have been developed considering Architecture and Urbanism design but only few have been translated to a real computerised implementation using a shape grammar interpreter: on an architectural and construction level, the grammar of Queen
Anne houses (Flemming 1987), the Malagueira grammar (Duarte 2005), the Yingzao fashi grammar (Li 2002) and, on an urban level, the Marrakech Medina grammar (Duarte et al. 2007), CityZoom (Grazziotin et al. 2007) and City Induction (Beirão et al. 2010). These are some notable exceptions, which, nevertheless, present approaches different from ours. The main challenges of our approach are the ability to integrate flexibility and adaptability, supporting a larger number of conditions and constraints and being able to evaluate and control the results. In the Emerg.Cities4all project we are currently developing a declarative language that would be used to represent shape grammars and also agents, their behavior and interaction.

5. Conclusion

We described the key ideas, technologies and system architecture involved in the multi-agent rule-based expert system shell for shape grammars that we are developing. Symbolic knowledge representation and reasoning, Rule Based Systems and Multi-Agent Systems are the core in this system. These are important features, especially in applications where huge and complex problem data, computations and reasoning processes are involved.

References


