ELSEVIER

Automation in Construction 14 (2005) 265–275

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira

José P. Duarte*

Massachusetts Institute of Technology, USA

Abstract

The ultimate goal of the described research is a process for mass customizing housing based on computer-aided design and production systems. The current goal is the development of an interactive system for generating solutions on the Web based on a modeling approach called discursive grammar. A discursive grammar consists of a programming grammar and a designing grammar. The programming grammar generates design briefs based on user data; the designing grammar provides the rules for generating designs in a particular style, and a set of heuristics guides the generation of designs towards a solution that matches the design brief. This paper describes the designing grammar using Siza's houses at Malagueira as a case study.

Keywords: Mass customization, Housing; Grammars; Siza; Design automation

1. Introduction

The ultimate goal of this work is the design and production of mass-customized houses. The current focus is on design aspects. The purpose of mass customization is to provide high-quality housing at an affordable cost. Quality is defined as the satisfaction of user needs. Cost is controlled by using computer-aided manufacturing, which does not rely on exhaustive repetition. Traditionally, when a designer is faced with the design of a large development, the usual solution is to design a limited number of house types and then to repeat them based on market analysis. The envisaged process aims at overcoming such limitations by using computer-aided design and manufacturing processes. The idea is to give mass-produced houses some of the qualities associated with individually designed homes.

The design system includes an interactive program for generating housing solutions, and rapid prototyping and virtual reality techniques for visualizing these solutions. The user accesses the program on the Web. The program guides the user through questions that an architect would normally ask during an initial meeting, such as the family members' profile, their living habits, the rooms they want, the cost that they can afford, and so on. When the interview is over, the

* Now at Instituto Superior Técnico, Portugal.
E-mail address: jduarte@civil.ist.utl.pt.
URL: http://www.civil.ist.utl.pt/arquitectura/pessoas/perfil/jpduarte_english.html.

0926-5805/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2004.07.013

program generates the design brief or housing program, taking into account existing housing regulations. The user can then make changes to the initial requirements, and the program will update the design brief. Once the brief is approved, the program generates a housing solution that satisfies the requirements, which the user can assess. At this stage, the user might want to change the initial requirements and proceed through another iteration of the design process. Once a solution is accepted, an order can be automatically issued to the housing factory. This order will include a detailed list of parts and the digital information to manufacture the parts using computer-aided manufacturing techniques. At the end of the manufacturing process, these parts are transported to the site and assembled.

The research reported in this paper aims at the development of a mathematical model for the interactive program just described. This model needs to overcome three problems that correspond to the three computer-based methods identified by Radford and Gero [1]. First, it needs to provide a way of translating client data into design requirements and to verify whether a design satisfies these requirements—the simulation problem. In simulation, the computer manipulates a mathematical model that describes the design to evaluate the performance of a given design configuration against the design requirements. Second, it has to codify the rules of formal composition to design a house in a given style—the generation problem. In generation, the computer is used to produce design configurations according to a set of rules. And third, it needs a mechanism to translate the design requirements into a housing solution—the optimization problem. In optimization, the computer is used to generate design configurations that meet a performance goal. The number of solutions that satisfy multiple requirements is potentially very large. Thus, an important part of the model is a computational strategy capable of searching a potentially large design space and providing insight into function–form relations for multicriteria housing design.

The proposed model is illustrated with a case study that includes specific programming and designing grammars. This paper is mainly concerned with the designing grammar. Section 2 briefly explains the concepts of discursive, shape, and description grammars. Section 3 briefly describes the programming grammar. Section 4 describes the designing grammar, and Section 5 closes the paper.

2. Discursive grammar

The simulation and generation problems mentioned above can be solved with the use of a description grammar [2] and a shape grammar [3], respectively. A grammar consists of a set of substitution rules that apply recursively to an initial assertion to produce a final statement. In description grammars, the assertions are symbolic descriptions, whereas in shape grammars, they consist of shape descriptions. In addition, description grammars deal with semantics, and shape grammars address form. The third problem is solved with a set of heuristics. Heuristics are used to choose a rule for application at each step of the design generation or to constraint choice to a small number of rules. Other heuristics assess the designs that would result from the application of each of the available rules, and then choose the one that takes the evolving design closer to the goal description provided in the design brief. This process is deterministic. At a microscale, a specific design context will lead to the application of a specific rule; at a macroscale, a given context will lead to a given housing solution. I call this mathematical model a discursive grammar, because it allows the generation of formally and semantically correct designs. Each house is like a piece of speech in the language that is appropriate for the context.

From the technical viewpoint, a discursive grammar consists of a shape grammar, a description grammar, and a set of heuristics. From the operative viewpoint, a discursive grammar consists of a programming grammar and a designing grammar. The programming grammar processes user and site data to generate the housing program (design brief). The designing grammar uses the housing program to generate a housing solution (design). The programming grammar has a description part and an empty shape part, whereas the designing grammar has both a description and a shape part as diagrammed in Fig. 1. In theory, different programming grammars can be combined with different designing grammars to form various discursive grammars. In practice, one has to ensure that the contexts of both grammars match, so