Computers & Graphics 34 (2010) 322–334

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

A GML shape grammar for semantically enriched 3D building models

Bernhard Hohmann [a], Sven Havemann [a,*], Ulrich Krispel [a], Dieter Fellner [a,b]

[a] Institute of Computer Graphics and Knowledge Visualization (CGV), Graz University of Technology, Inffeldgasse 16c/II, 8010 Graz, Austria
[b] Fraunhofer IGD & Darmstadt University of Technology, Fraunhoferstrae 5, 64283 Darmstadt, Germany

ARTICLE INFO

Keywords:
Shape grammars
Architectural buildings
Shape modeling
3D-reconstruction
Shape semantics
Generative modeling

ABSTRACT

The creation of building and facility models is a tedious and complicated task. Existing CAD models are typically not well suited since they contain too much or not enough detail; the manual modeling approach does not scale; different views on the same model are needed, and different levels of detail and abstraction; and finally, conventional modeling tools are inappropriate for models with many internal parameter dependencies. As a solution to this problem we propose a combination of a procedural approach with shape grammars. The model is created in a top-down manner; high-level changeability and re-usability are much less of a problem; and it can be interactively evaluated to provide different views at runtime. We present some insights on the relation between imperative and declarative grammar descriptions, and show a detailed case study with facility surveillance as a practical application.

1. Introduction

Three-dimensional building and facility models are becoming ever more important with the widespread availability of building information systems. They are the basis for home automation, sensor networks, technical supervision and inspection, building surveillance systems as well as all kinds of maintenance tasks in the context of *building lifecycle management*. A central problem is scalability: Especially for complex facilities, the creation of appropriate 3D models is an extremely tedious task. Conventional interactive modeling software a la 3D House Designer, or general purpose software such as Maya or 3D Studio Max is suitable only for comparably small facilities.

For a surveillance project we needed a 3D model of a sufficiently complex facility, four university buildings (cf. Fig. 1). We made the first attempt for interactive reconstruction using Google SketchUp. The biggest problem we encountered was that errors early in the modeling process can practically not be corrected at all later on. If, e.g., the floor height must be changed, essentially the whole construction needs to be re-done.

1.1. Existing architectural software is inappropriate

Another option we thought of was to use architectural software such as ArchiCAD, Autodesk Revit or Architectural Desktop. However, this software is targeted at construction rather than reconstruction, i.e., reverse engineering of buildings. But even these so-called "associative models" are low-level since almost all of the geometry is constructed manually. Only certain parameter associations are kept consistent automatically, i.e., when dimensions of the model are changed this is (somehow) propagated to sub-parts. Architectural software cannot easily be integrated into, e.g., a surveillance application. But when a building model is exported, it loses much of its semantics. The CAD export contains (too) detailed geometry of walls, doors, etc., but there is no geometry for rooms and corridors that are just empty space. But this space is just where people live and it is therefore the main unit for surveillance. One viable alternative might be exporting the CAD model to a format with rich semantics such as IFC International Alliance for Interoperability [2]. It is apparently the upcoming open exchange standard for building semantics, and it supports labels for rooms, corridors, etc.

1.2. Needed: procedures and complex parameter dependencies

The focus of our work, however, was on creating a *sustainable* building model with a minimum of initial effort in terms of taking manual measurements. The model should allow replacing anytime later guessed values by accurate measurements taken with our laser range finder (Fig. 2). So the idea was to start with a

* Corresponding author.
E-mail addresses: b.hohmann@cgv.tugraz.at (B. Hohmann), s.havemann@cgv.tugraz.at (S. Havemann), u.krispel@cgv.tugraz.at (U. Krispel), d.fellner@cgv.tugraz.at (D. Fellner).
URLS: http://www.cgv.tugraz.at/hohmann (B. Hohmann), http://www.cgv.tugraz.at/havemann (S. Havemann), http://www.cgv.tugraz.at/krispel (U. Krispel), http://www.cgv.tugraz.at/fellner, http://www.igd.fhg.de/igd-a0/staff/fellner (D. Fellner).

0097-8493/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2010.05.007

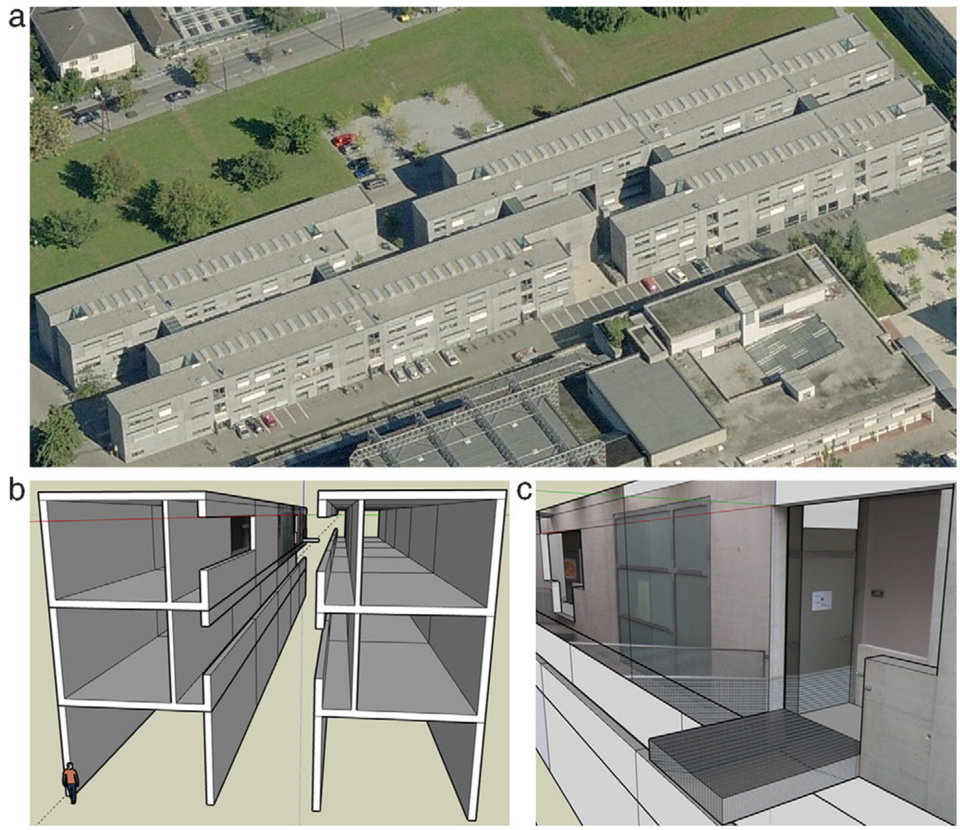

Fig. 1. (top) University facility: four buildings, three floors, 405 rooms. Photo: Bing Maps, Bird's Eye view [1]. (bottom) Reconstruction attempt from photographs using Google SketchUp, with obvious scalability problems.

Fig. 2. Leica Disto laser range finder with 1–2 mm precision over 60 m.

model that may be geometrically inaccurate but is *qualitatively correct*, meaning that all of the major building features (walls, doors, windows, etc.) are present. It should be possible to generate them quickly by just instantiating parametric templates. High-level changeability was to be granted by the ability to express any sort of parameter dependencies explicitly. For example, in our building a sequence of rooms along a corridor all have the same height and width, but the room length (along the corridor) can vary. So it needs to be possible to define the model in such a way that only one parameter per room is required, namely its length, and only a single width value per corridor, which then each room refers to as its width parameter.

We found also more complex dependencies that require, e.g., changing the reference frame when a measurement does not correspond directly to a parameter. There are surprisingly many possible parametrizations of a shape as simple as a box: p_{min}, p_{max}; p_{mid}, (r_x, r_y, r_z); or just the midpoint of a bottom edge, and orientation and extents are inherited from higher levels. This example shows that no limited number of pre-defined parametrizations can ever be sufficient; instead it must be possible to re-parameterize the boxes whenever needed. A more involved example is given in Section 5.3.

1.3. Main contributions

- *Simple but powerful split grammar formalism*: Our GML shape grammar toolkit develops considerable expressiveness out of about two dozen functions.
- *Unified view on grammars and imperative modeling*: Grammars are typically perceived as declarative, but we use them within an imperative paradigm. This permits us to overcome some inherent limitations.
- *Practical reconstruction of a complex facility with interiors*: The original motivation was to rapidly create a complex facility model that should still be sustainable.
- *A solution to the problem of propagating reference frames*: In many cases measurements do not directly correspond to model parameters. We present a simple, general solution to this problem.

2. Related work

Since ancient times complex architectural buildings were designed in a top-down, coarse-to-fine manner: The overall structure is divided into sections and floors, which are further refined into rooms, hallways, stairways, etc. A more formal view on this process suggests a grammar-based approach, which directly leads to the so-called shape grammars. They were first introduced by Stiny and Gips as early as 1972 [3]. Their very general idea was to simply replace a shape (or part of a shape) that carries a label by one or more (usually smaller) shapes carrying other labels. This hierarchical replacement process is specified by a finite set of shape grammar rules.

Many variations to this basic idea have been developed, from non-determinism over conditional rules to L-systems [4], e.g., for