Shape grammar implementations

The last 35 36 years

Scott C. Chase
Architecture, Design & Media Technology
Aalborg University

Shape grammar implementation: from theory to usable software
Design Computing and Cognition workshop, Stuttgart, 11 July 2010
Outline

- Overview & issues
- Early history
- Examples
  - Categorised by issue
Today’s presentations

- Li, Chau, Chen, Wang
  *A prototype system for developing two- and three-dimensional shape grammars*

- Trescak, Esteva, Rodriguez
  *Shape grammar interpreter for rectilinear forms*

- Hoisl, Shea
  *A 3D spatial grammar interpreter applet*

- Jowers, Earl
  *QI – a shape grammar interpreter for curved shapes*

- Ertelt, Shea
  *Shape grammar implementation for machining planning*

- Jowers, McKay
  *Shape grammar implementation with vision*

- Correia, Duarte, Leitão
  *MALAG: a discursive grammar interpreter for the online generation of mass customized housing*
Challenge

We want conceptual design tools that support designers’ ways of thinking and working and enhance creativity, e.g. offering design alternatives difficult or not possible without the use of such tools.
Shape grammars
Emergence
Conceptual design tool requirements

DCC 2010 workshop notes

- Ease of use
- Modeling capabilities
- Visualization capabilities
- Multiplicity
- Flexibility
- Simultaneity
- Environment
- Semantics

- Entity identity vs. emergence
- Entity linkages
- Abstract objects
- Diagram support
- History and Design Space exploration
- (Re)generativity
SG implementation research

- Representations & algorithms
  - geometry, other design attributes, control
- User interaction/interface
- Specific design problems
- Integration into design process
Issues
Gips 1999

1. Interface
2. Parametric grammars
3. Subshape problem
4. Curved elements
5. Representations
6. Extensions to SG
7. ‘Proof of concept’ vs. production software
8. The ‘big enchilada’ or ‘one piece at a time’

http://www.shapegrammar.org/implement.pdf
Idealised general SG implementation  
Chau et. al (2004)

1. Subshape recognition and emergence  
2. Shape recognition under Euclidean transformations  
3. Parametric shape rules  
4. Shape recognition for parametric grammars  
5. 3D shapes  
6. Curvilinear basic elements  
7. Intuitive user interface  
8. Aesthetic measures for ranking & selecting designs  
9. Surfaces and solids  
10. Unambiguous interpretation of designs to physical realisation

SG system tasks
Gips 1999

1. Generation (design)
2. Parsing (analysis)
3. Inference (grammar construction)
4. CAD program for SG development (designer’s aid)
History of implementations

- Early work (1970s & 80s)
  - Primarily general interpreters

- Middle period (1990s & early 2000s)
  - Broader work includes systems for specific design problems
  - Work includes systems that don’t support emergence

- Past decade: broad mix
  - General interpreters
  - Specific implementation issues
  - Specific design problems
## Implementations

Chau et. al 2004

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Tool(s) used</th>
<th>Shape emergence</th>
<th>2D/3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple interpreter</td>
<td>Gips 1975</td>
<td>SAIL¹</td>
<td>No</td>
<td>2D</td>
</tr>
<tr>
<td>Shepard-Metzler analysis</td>
<td>Gips 1974</td>
<td>SAIL¹</td>
<td>No</td>
<td>2D/3D</td>
</tr>
<tr>
<td>Shape grammar interpreter</td>
<td>Krishnamurti 1982</td>
<td>Conventional language</td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Shape generation system</td>
<td>Krishnamurti and Giraud 1986</td>
<td>PROLOG²</td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Queen Anne houses</td>
<td>Flemming 1987</td>
<td>PROLOG</td>
<td>No</td>
<td>2D</td>
</tr>
<tr>
<td>Shape grammar system</td>
<td>Chase 1989</td>
<td>PROLOG</td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Genesis (CMU)</td>
<td>Heiserman 1991</td>
<td>C/CLP(R)³</td>
<td>No</td>
<td>3D</td>
</tr>
<tr>
<td>GRAIL</td>
<td>Krishnamurti 1992</td>
<td></td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Grammatica</td>
<td>Carlson 1993</td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Stouffs 1994</td>
<td></td>
<td></td>
<td>Yes</td>
<td>2D/3D</td>
</tr>
<tr>
<td>Genesis (Boeing)</td>
<td>Heisserman 1994</td>
<td>C++/CLP(R)³</td>
<td>No</td>
<td>2D/3D</td>
</tr>
<tr>
<td>GEdit³</td>
<td>Tapia 1996</td>
<td>LISP⁴</td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Shape grammar editor</td>
<td>Shelden 1996</td>
<td>AutoLISP</td>
<td>Yes</td>
<td>2D</td>
</tr>
<tr>
<td>Implementation of basic grammar</td>
<td>Simondetti 1997</td>
<td>AutoLISP</td>
<td>No</td>
<td>3D</td>
</tr>
<tr>
<td>Shape grammar interpreter</td>
<td>Piazzalunga and Fitzhorn 1998</td>
<td>ACIS Scheme</td>
<td>No</td>
<td>3D</td>
</tr>
<tr>
<td>SG-Clips</td>
<td>Chien et al 1998</td>
<td>CLIPS</td>
<td>No</td>
<td>2D/3D</td>
</tr>
<tr>
<td>3D Shaper</td>
<td>Wang 1998</td>
<td>Java/Open Inventor</td>
<td>No</td>
<td>3D</td>
</tr>
<tr>
<td>Coffee maker grammar</td>
<td>Michalek 1998</td>
<td>Java</td>
<td>No</td>
<td>2D/3D</td>
</tr>
<tr>
<td>MEMS grammar</td>
<td>Agarwal et al 2000</td>
<td>LISP</td>
<td>2D</td>
<td></td>
</tr>
<tr>
<td>Shaper 2D⁷</td>
<td>McGill 2001</td>
<td>Java</td>
<td>No</td>
<td>2D</td>
</tr>
<tr>
<td>U13 shape grammar implementation</td>
<td>Chau 2002</td>
<td>Perl</td>
<td>Yes</td>
<td>3D</td>
</tr>
</tbody>
</table>
A SYNTAX-DIRECTED PROGRAM THAT PERFORMS A THREE-DIMENSIONAL PERCEPTUAL TASK

JAMES GIPS
260 South Sycamore Avenue. Los Angeles, California 90036, U.S.A.

(Received 14 February 1974)

INTEGER PROCEDURE SAME.OR.MI (INTEGER ARRAY EQUIV);
BEGIN
INTEGER I, MINUSAXES, DIFFAXES;
MINUSAXES ← DIFFAXES ← 0;
FOR I ← 1, 2, 3 DO
BEGIN
IF ABS(EQUIV[I]) = 1 THEN DIFFAXES ← DIFFAXES + 1;
IF EQUIV[I] < 0 THEN MINUSAXES ← MINUSAXES + 1;
END;
IF (MINUSAXES = 1) ∨ (MINUSAXES = 3)
THEN RETURN (IF DIFFAXES = 2 THEN SAME ELSE M I)
ELSE RETURN (IF DIFFAXES = 2 THEN M I ELSE SAME);
END;

Fig. 1(a): Line drawings portraying identical objects rotated in picture plane.

Fig. 1(b): Line drawings portraying identical objects rotated to depth.

Fig. 1(c): A pair of more complicated lists are essentially analyzed and compared by the program.
Simple interpreter
Gips 1975

Shape grammar implementations: the last 36 years
Design Computing & Cognition workshop, 11 July 2010
SGI
Krishnamurti 1982

1. Who has referenced Krishnamurti’s 1982 report in their papers?

2. Who has actually seen the report?
Shape grammar implementations: the last 36 years

Design Computing & Cognition workshop, 11 July 2010

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>display shape</td>
<td>Activate and display the indicated shape. The subsequent commands refer to this active shape</td>
</tr>
<tr>
<td>B</td>
<td>reset the debugging switch</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>line/point no.</td>
<td>select the indicated line (or point) as the Current line (or point)</td>
</tr>
<tr>
<td>D</td>
<td>[line/point no.]</td>
<td>Delete the current [or indicated] line/point and renumber if necessary</td>
</tr>
<tr>
<td>E</td>
<td>[rule no.]</td>
<td>Enter a copy of the active shape as a side of the indicated shape rule. If no rule number is supplied, increment the highest rule entered by 1 to give the new rule number</td>
</tr>
<tr>
<td>F</td>
<td>[side] [rule no.]</td>
<td>Fetch a copy of the indicated side of the indicated shape rule onto the active shape. If no side is supplied, the entire shape rule is fetched onto the rule shapes. If no rule number is supplied, the last rule entered is fetched</td>
</tr>
<tr>
<td>G</td>
<td>production rules</td>
<td>enter the Generation phase. The indicated production rules are the only shape rules that can be referenced in the generation phase</td>
</tr>
</tbody>
</table>
SG interpreter
Chase 1987

Interface/Interaction
Grammar use & interaction
Chase 2002

Chase S C, 2002, "A model for user interaction in grammar-based design systems"  
*Automation in Construction* 11 161-172
Grammar interaction
Chase 1987 & 2002

a) Manual mode (Scenario 2)

- Select Rule
- Select 2 triples of points
- Calculate transform
- $a \leq b$?
  - no
    - New shape
  - yes
    - Apply rule

b) Semi-automatic mode (Scenario 4)

- Select Rule
- Calculate distinguishable points
- Calculate set of transforms
- $a \leq b$?
  - no
    - Store others if desired
  - yes
    - Apply rule
    - for each transform
    - Select one to continue
    - Set of new shapes
GEdit
Tapia 1996

Tapia M, 1999, "A visual implementation of a shape grammar system"
*Environment and Planning B: Planning and Design* 26 59-73
3D Shaper
Wang 1998

Designing With Vision

http://design.open.ac.uk/DV
Extensions
Yingzao fashi grammar
Li 2002

Non-geometric attributes

QI (curves)

Jowers 2006

Parametric SG interpreter
Krishnamurti 2010
Graph grammars

- Schmidt (from PhD 1995)
- Campbell
GraphSynth
Campbell 2010

http://www.graphsynth.com
Integration with design & production processes
Design Synthesis & Shape Generation
McKay et al. 2007-08

http://www.engineering.leeds.ac.uk/dssg

... we anticipate three intertwined cycles

The designer designing shapes

Communication between the two

The Shape Synthesis System generating shapes
Design Synthesis & Shape Generation
McKay et al. 2007-08

Shape grammar implementations: the last 36 years
Design Computing & Cognition workshop, 11 July 2010
Design Synthesis & Shape Generation
McKay et al. 2007-08
Industrial strength interpreters
Genesis-PhD
Heisserman 1991

Genesis-Boeing
Heisserman since 1991
EifForm
Shea from 1997

Planar truss grammar

SG interpreter
patents
McCormick & Cagan 2006/9

http://www.freepatentsonline.com/7050051.html
http://www.freepatentsonline.com/7502511.html
Specific design applications
Specific design applications

- Product development
  - Coffeemaker (Agarwal & Cagan, 1998)
  - Dove (Chau, 2002)
  - Buick (McCormack et al., 2004)
  - Coca-Cola (Chen, 2005)
  - General shampoo bottle grammar (Chen 2005)

- Architecture
  - MALAG (Duarte 2005)
Coffee maker grammar
Agarwal et al 1999

Duarte J P, 2005, "A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira"

*Automation in Construction* **14** 265-275
Recent general interpreters
3D interpreter
Chau 2002

SG development system
Li et al. 2010

Li, Andrew I-K, Chau H H, Chen L, Wang Y, 2009, "A Prototype System for developing two- and Three-Dimensional Shape Grammars", in Proceedings of the 14th International Conference on Computer Aided Architectural Design Research in Asia (CAADRIA, Yunlin, Taiwan) 717-726
SGI (2)
Trescak et al. 2009

http://sourceforge.net/projects/sginterpreter
Interactive 3D Spatial Grammar System
Hoisl & Shea 2010

http://sourceforge.net/projects/spapper
Shape Designer (v2)
Wong et al. 2004-5

In conclusion…

- We still have a long way to go to make an impact on industry methods using grammar based approaches

- Areas with a lot of activity; maturity?
  - Representations
    - Including extensions, e.g. curves, parametrics, non-geometric attributes
  - Interfaces

- Promising areas
  - New methods of interaction
  - Integration w/design & production processes
Demo time!