Modern Programming for Generative Design
MSc in Computer Engineering and Information Systems

José Lopes

Instituto Superior Técnico
Technical University of Lisbon

July 18, 2012
Generative Design
Survey of currently used systems

- Textual Programming Languages
- Visual Programming Languages
- CAD Applications
Survey of currently used systems

- Functionality
- Linguistic constructs
- Geometric abstractions
Survey of currently used systems (Example)

Figure: Grasshopper program
Survey of currently used systems (Example)

Figure: Grasshopper program (excerpt)
Survey of currently used systems (Example)

Figure: *Grasshopper* program (excerpt)
Survey of currently used systems (Example)

**Figure:** Grasshopper program (complete)
Generative Design Principles

- Portability
- Parametric elements
- Functional operations
- ...

- Modern programming environment: Rosetta
Portability

- Programs are not portable
- Vendor lock-in
Portability in Rosetta

- Javascript
- Racket
- AutoLISP
- ...
Portability in Rosetta

```
; lang rosetta-racket

(define (roundc c)
  (xyz
    (round (xyz-x c))
    (round (xyz-y c))
    (round (xyz-z c))
  )
)

(define (random-interval-round a b)
  (round (random-interval a b))
)

(define (pipe-shape p0 v0 l n valid?)
  (if (~ n 0)
      (list)
      (let ([v1 (roundc
        (epsh 1
          (* (random-interval-round 0 4) p1/2)
          (* (random-interval-round 0 5) p1/2)))]
        (cond ((< v0 (c v1 -1))
          (pipe-shape p0 v0 l n valid?)
        (else
          (let ([p1 (~ p0 (* v1 (* (random-interval 0.2 1.0) l))))
            (cond ((valid? p1)
              (list
                (cylinder p0 (* 0.02 l) p1)
                (sphere p1 (* 0.04 l))
                (pipe-shape p1 v1 l (~ n 1) valid?)
              (else
                (pipe-shape p0 v0 l n valid?)
              ))))))
))
)

(define (pipes-cube p r l n)
  (pipe-shape p (xyz 1 0 0) 1 n)
  (lambda (p1)
    (let ([v (~ 0 p1)])
      (and (< (abs (xyz-x v)) v)
        (< (abs (xyz-y v)) v)
        (< (abs (xyz-z v)) v)))))
```
Portability in Rosetta

Modern Programming for Generative Design
Portability in Rosetta

```
(define (roundc c)
  (vector (round (nxtw* c 0)) (round (nxtw* c 1)) (round (nxtw* c 2))))

(define (random-interval-round a b)
  (round (random-interval a b)))

(define (pipe-shape p0 v0 l valid?)
  (if (~v0 0)
      (list)
      (let (v1 (roundc
                 (sph 1
                 (* (random-interval-round 0 1) p1/2)
                 (* (random-interval-round 0 1) p1/2)))))
       (cond ((~v0 0) (~v1 1))
             (pipe-shape p0 v0 l valid?)
            (else
             (let ((p1 (c p0 (~v1 1 (* (random-interval 0.2 1.0) l))))
                  (valid? p1)
                  (list
                   (cylinder p0 (* 0.02 1) p1)
                   (sphere p1 (* 0.04 1))
                   (pipe-shape p1 v1 l (~v0 1) valid?)))))
      (else
       (pipe-shape p0 v0 l valid?))))))

(define (pipe-cube p l n)
  (pipe-shape p (xyz 1 0 0) l n)
  (let ((p1)
        (let ((v (~p p1))
              (and (< (abs (xyz-x v)) x)
                   (< (abs (xyz-y v)) y)
                   (< (abs (xyz-z v)) 1))))))
```
Portability in Rosetta
Parametric elements

\[ \text{spiral}(t) = \begin{cases} 
\rho = \alpha t \\
\phi = \beta t \\
z = t 
\end{cases} \]

Figure: Conic spiral tower
Parametric elements

\[ \text{spiral}(t) = \begin{cases} 
\rho = \alpha t \\
\phi = \beta t \\
z = t
\end{cases} \]

function spiral(t) {
    return cyl(a * t, b * t, t);
}

Figure: Conic spiral tower
Parametric elements

**Figure:** Conic spiral sampling
function spiral(t) {
    return cyl(a * t, b * t, t);
}

; sampling
function spiralPoints(n) {
    var points = [];
    for (var i = 0; i < n; ++i) {
        points[i] = spiral(i / n);
    }
    return points;
}

sweep(spline(spiralPoints(n)), circle(1));
function spiral(t) {
    return cyl(a * t, b * t, t);
}

sweep(functionCurve(spiral), circle(1));
Mathematical and geometric strictness

Symmetric difference ($\Delta$)

$$\Delta(R_0, R_1) = (R_0 \cup R_1) - (R_0 \cap R_1)$$
Mathematical and geometric strictness

\[ \Delta(R_0, R_1) = (R_0 \cup R_1) - (R_0 \cap R_1) \]

```javascript
function delta(r0, r1) {
    return subtract(
        union(r0, r1),
        intersect(r0, r1));
}
```
Mathematical and geometric strictness

\[ \Delta(R_0, R_1) = (R_0 \cup R_1) - (R_0 \cap R_1) \]

function delta(r0, r1) {
    var r0Copy = copy(r0);
    var r1Copy = copy(r1);
    return subtract(
        union(r0, r1),
        intersect(r0Copy, r1Copy));
}
Mathematical and geometric strictness

\[ \Delta(R_0, R_1) = (R_0 \cup R_1) - (R_0 \cap R_1) \]

```javascript
function delta(r0, r1) {
    var r0Copy = copy(r0);
    var r1Copy = copy(r1);
    if (isCurve(r0) && isCurve(r1)) {
        return subtractCurves(
            unionCurves(r0, r1),
            intersectCurves(r0Copy, r1Copy));
    } else if (isSurface(r0) && isSurface(r1)) {
        ...
    } else if ...
} 
```
Mathematical and geometric strictness

\[ \Delta(R_0, R_1) = (R_0 \bigcup R_1) - (R_0 \bigcap R_1) \equiv (R_0 - R_1) \bigcup (R_1 - R_0) \]

```javascript
function delta(r0, r1) {
  var r0Copy = copy(r0);
  var r1Copy = copy(r1);
  if (isCurve(r0) && isCurve(r1)) {
    return subtractCurves(
      unionCurves(r0, r1),
      intersectCurves(r0Copy, r1Copy));
  } else if (isSurface(r0) && isSurface(r1)) {
    ...
  } else if ...
}
```
Mathematical and geometric strictness

\[ \Delta(R_0, R_1) = (R_0 \cup R_1) - (R_0 \cap R_1) \equiv (R_0 - R_1) \cup (R_1 - R_0) \]

function delta(r0, r1) {
    var r0Copy = copy(r0);
    var r1Copy = copy(r1);
    if (isEmptyIntersection(r0, r1)) {
        return union(
            subtract(r0, r1),
            subtract(r1Copy, r0Copy));
    } else if (isCurve(r0) && isCurve(r1)) {
        return subtractCurves(
            unionCurves(r0, r1),
            intersectCurves(r0Copy, r1Copy));
    } else if (isSurface(r0) && isSurface(r1)) {
        ...
    } else if ...
}
Mathematical and geometric strictness in Rosetta

- Functional operations
- Operations implement algebraic equivalences
- Dimension independent operations
Shape morphing
Traceability

- Relationship between program and model
- Understanding, maintaining, debugging
Traceability in Rosetta

Figure: Traceability: from program to model
Traceability in Rosetta

Figure: Traceability: from model to program
Immediate feedback

- Interactive input adjustment
- CAD applications designed for interaction
Immediate feedback in Rosetta

![Images of 3D models: Orthogonal cones, Möbius truss, and Scriptecture]

<table>
<thead>
<tr>
<th>Example/Application</th>
<th>AutoCAD</th>
<th>Rhinoceros</th>
<th>OpenGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthogonal cones</td>
<td>1022</td>
<td>191</td>
<td>1</td>
</tr>
<tr>
<td>Möbius truss</td>
<td>28837</td>
<td>9235</td>
<td>4446</td>
</tr>
<tr>
<td>Scriptecture</td>
<td>21920</td>
<td>5088</td>
<td>210</td>
</tr>
</tbody>
</table>

**Table:** Time (in milliseconds) to regenerate the model
Immediate feedback in Rosetta
Evaluation

- Program development
- Programming environment extension
- Program analysis and conversion
New backend: TikZ
New frontend: RosettaFlow
New frontend: RosettaFlow
New frontend: RosettaFlow
Program analysis and conversion
Conclusion

Generative Design needs:

- Portability
- Mathematical and geometric strictness
- Correlation between programs and models
- Multiple paradigms and techniques
- Modern and pedagogic system
Conclusion

- Devise set of Design Principles
- Rosetta implements these principles
- Rosetta is being used by designers
Contributions

- Programming Languages For Generative Design: A Comparative Study
  *journal* International Journal of Architectural Computing

- Portable Generative Design for CAD Applications
  *conference* ACADIA 11: Integration through Computation

- Essential Language Features for Generative Design
  *conference* III Simpósio de Informática (INForum 2011)

- Collaborative Digital Design (accepted)
  *conference* eCAADe 2012: Digital Physicality, Physical Digitality
Modern Programming for Generative Design
José Lopes

Questions?