Context-Oriented Algorithmic Design

Bruno Ferreira
António Menezes Leitão
Algorithmic Design
BIM Tools

```lisp
(lisp
  (define beam-family-1 (load-family "C:\ProgramData\Autodesk")
  (define beam-family-2 (load-family "C:\ProgramData\Autodesk")
  (define beam-family-3 (load-family "C:\ProgramData\Autodesk")
  (define beam-family-element-1
    (family-element beam-family-1))
  (define beam-family-element-2
    (family-element beam-family-2))
  (define beam-family-element-3
    (family-element beam-family-3))
  (create-beam (xyz 0 0 0)
    (xyz 0 10 0))
  (create-beam (xyz 5 0 0)
    (xyz 5 10 0)
    ;#:family beam-family-element-1)

(idstr "#hasheqv() " 266712)
(idstr "#hasheqv() " 266719)
(idstr "#hasheqv() " 266806)
)```
BIM Tools

```lisp
(defend (bar pts)
  (tor/list ((p0 pts) (p1 (cdr pts)))
     (beam p0 p1)))

(define (rotated-bar pts angles)
  (tor/list ((p0 pts) (p1 (cdr pts)) (a angles))
     (beam p0 p1 a)))

(define (frames p)
  (tor/list ((f1 (division 2pi 0 n-frames-for-frames #f))
     (psi (division (/ pi -2) (+ (/ pi -2) skin-rotation)) n-fram
     (bar (list
       (+spf (rot-base-point f1 ro-f f1 (+ psi psiA))
       (+spf (rot-base-point f1 ro-f f1 (+ psi psiB))
       (+spf (rot-base-point f1 ro-f f1 (+ psi psiC))
       (+spf (rot-base-point f1 ro-f f1 (+ psi psiD)))))

;STEEL CONNECTIONS

(define (conections p)
  (define (ro-f x y) (sqrt (+ (sqr (/ x 2)) (sqr (/ y 2)))))
  (define psiA x y) (atan (/ y 2) (/ x 2))
  (define psiB x y) (+ pi (atan (/ y 2) (/ x 2)))
  (define psiC x y) (- (atan (/ y 2) (/ x 2)))
  (define psiD x y) (- (atan (/ y 2) (/ x 2)))

(define (beam2column pt p2)
  (define (dif-almost? a b) (< -10 (- a b) 10))
  (define (dif-almost0? a b) (< -1 (- a b) 1))
  (if (string? current-backend-name) "ArchiCAD"
    (if (and (dif-almost? (cx pt1) (cx p2))
            (dif-almost0? (cy p1) (cy p2)))
      "..."))}
```
Complex Buildings
Complex Buildings
Need For Analysis
Several Tools
Khepri

- Python
- JavaScript
- OpenGL
- R
- A

Diagram showing these technologies connected to a central object.
Khepri
Khepri
Khepri
Code can become messy
One script for all models
Context-Oriented Programming
Behavioral Variations
Layers
Activation mechanism
Context
Scoping
Objectives

> Present and compare COP

> Combine COP with AD
Related Work
Aspect-Oriented Programming

- Cross-cutting concerns
- Aspects
- Pointcuts, Join Points, Advice
Subject-Oriented Programming

> Subjects

> Subject-activation

> Subjective Dispatch
Context-Oriented Programming

> Contexts

> Contextual Dispatch
> Common Lisp

> Dynamically Scoped Layer Activation

> Layered Generic Functions

> Layer-in-class and class-in-layer
(deflayer employment-layer)

(with-active-layers (employment-layer)
    ... contained code ...)

(define-layered-class employer
    :in-layer employment-layer ()
    ((name :initarg :name
        :layered-accessor employer-name)))
> Python

> Dinamically Scoped Layer Activation

> Decorators

> Layer-in-class
class Slab:

    @around(a3DLayer)
    def generate(self):
        return extrusion(surface_from(self.path),
                         self.thickness)

    @around(a2DLayer)
    def generate(self):
        return self.path
ContextJ

> Java (source-to-source compiler)

> Dynamically Scoped Layer Activation

> Reflection API

> Layer-in-class
class Employer{
    String toString() {
        return "Name: " + name;
    }
}

layer Address {
    String toString() {
        return proceed() + "; Address: " + ...
    }
}

Lambic

> Common Lisp

> Predicate Dispatch

> Activation with Predicates

> Different modularization
(defgeneric factorial (n)
  (:predicates < = >))

(defmethod factorial (n)
  (:when (> n 0))
  (* n (factorial (- n 1))))

(defmethod factorial (n)
  (:when (= n 0))
  1)
<table>
<thead>
<tr>
<th>Comparison</th>
<th>Base Language</th>
<th>Implementation</th>
<th>Layer Activation</th>
<th>Modularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContextL</td>
<td>Common Lisp</td>
<td>Library</td>
<td>DSA</td>
<td>LIC, CIL</td>
</tr>
<tr>
<td>ContextScheme</td>
<td>Scheme</td>
<td>Library</td>
<td>DSA</td>
<td>CIL</td>
</tr>
<tr>
<td>ContextErlang</td>
<td>Erlang</td>
<td>Library</td>
<td>Per-Agent</td>
<td>Erlang Modules</td>
</tr>
<tr>
<td>ContextJS</td>
<td>JavaScript</td>
<td>Library</td>
<td>Open Implementation</td>
<td>LIC, CIL</td>
</tr>
<tr>
<td>PyContext</td>
<td>Python</td>
<td>Library</td>
<td>DSA, Implicit</td>
<td>CIL</td>
</tr>
<tr>
<td>ContextPy</td>
<td>Python</td>
<td>Library</td>
<td>DSA</td>
<td>LIC</td>
</tr>
<tr>
<td>ContextJ</td>
<td>Java</td>
<td>S2S</td>
<td>DSA</td>
<td>LIC</td>
</tr>
<tr>
<td>JCop</td>
<td>Java</td>
<td>S2S and Aspect</td>
<td>DSA, declarative, conditional</td>
<td>LIC</td>
</tr>
<tr>
<td>EventCJ</td>
<td>Java</td>
<td>S2S and Aspect</td>
<td>DSA</td>
<td>LIC</td>
</tr>
<tr>
<td>JavaCtx</td>
<td>Java</td>
<td>Library and Aspect</td>
<td>DSA</td>
<td>LIC</td>
</tr>
<tr>
<td>ContextR</td>
<td>Ruby</td>
<td>Library</td>
<td>DSA</td>
<td>LIC</td>
</tr>
<tr>
<td>ContextLua</td>
<td>Lua</td>
<td>Library</td>
<td>DSA</td>
<td>CIL</td>
</tr>
<tr>
<td>ContextS</td>
<td>Smalltalk</td>
<td>Library</td>
<td>DSA, indefinite</td>
<td>CIL</td>
</tr>
<tr>
<td>Ambience</td>
<td>AmOS</td>
<td>Library</td>
<td>DSA, global</td>
<td>CIL</td>
</tr>
<tr>
<td>Lambic</td>
<td>Common Lisp</td>
<td>Library</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Subjective-C</td>
<td>Objective-C</td>
<td>Preprocessor</td>
<td>Global</td>
<td>LIC</td>
</tr>
</tbody>
</table>
Context-Oriented Algorithmic Design
Implementation

- COP Library
- Khepri
  - Visualization Tool
  - Rendering Tool
  - Analysis Tool
  - ...

...
Implementation

> ContextScheme (adapted to Racket)

> Khepri
Case Study
Case Study
(define (shop-2d ...) 
  (... 
    (line ...) 
    (rectangle ...)) 
  ...))

(define (shop-3d ...) 
  (... 
    (right-cuboid ...)) 
  ...))
(define (mall ... shop ...) 
  (... 
    (shop ... )
  ...))
(define (shop p v l w)
  (...)
  ((wall) p0 p1 wall-thickness wall-height)
  ...)
  ((door) ((wall) p2 p3 wall-thickness wall-height)
    p4 p5 wall-thickness door-height))))
(with-layers (2D)
  (mall (xy 0 0) 100000 12000 25000 7000 7000 4))
COP 2D
COP 3D
COP Analysis
Partial Modelling
Partial Modelling
Advantages

> Reduces the code

> Dynamic context change
Evaluation
Evaluation

> Higher-order or not?

> Implicit or Explicit Activation?

> Performance?
Future Work
Thank you!

Questions?

bruno.b.ferreira@tecnico.ulisboa.pt
antonio.menezes.leitao@tecnico.ulisboa.pt

https://algorithmicdesign.github.io/