

# MALAGUEIRA

ALVARO SIZA VIEIRA - 1977 / 99



## GRAMATICA

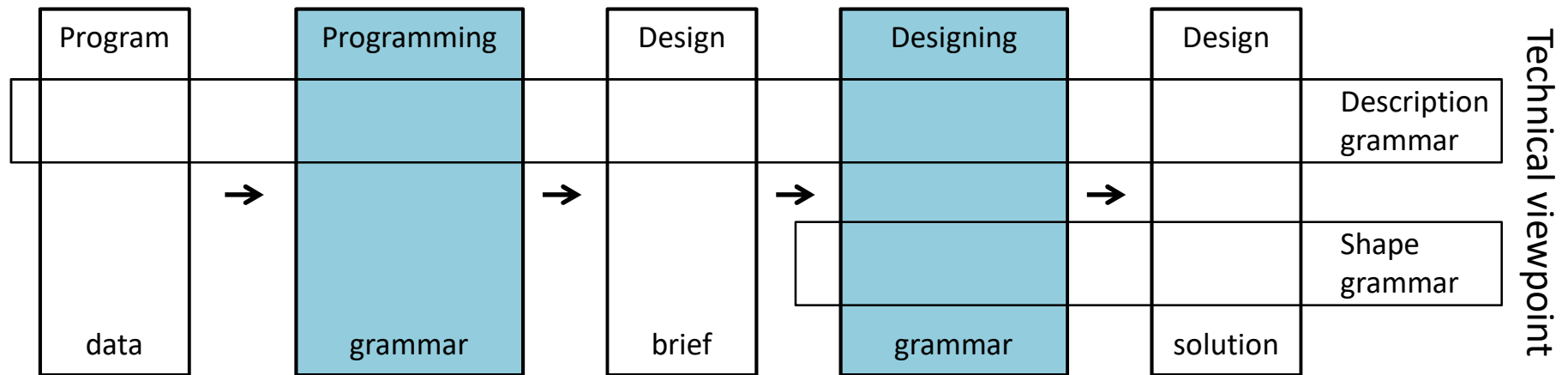
eCAADe 2012 - Prague

Rodrigo Correia  
José Duarte  
António Leitão

# Mass Customization Of Housing

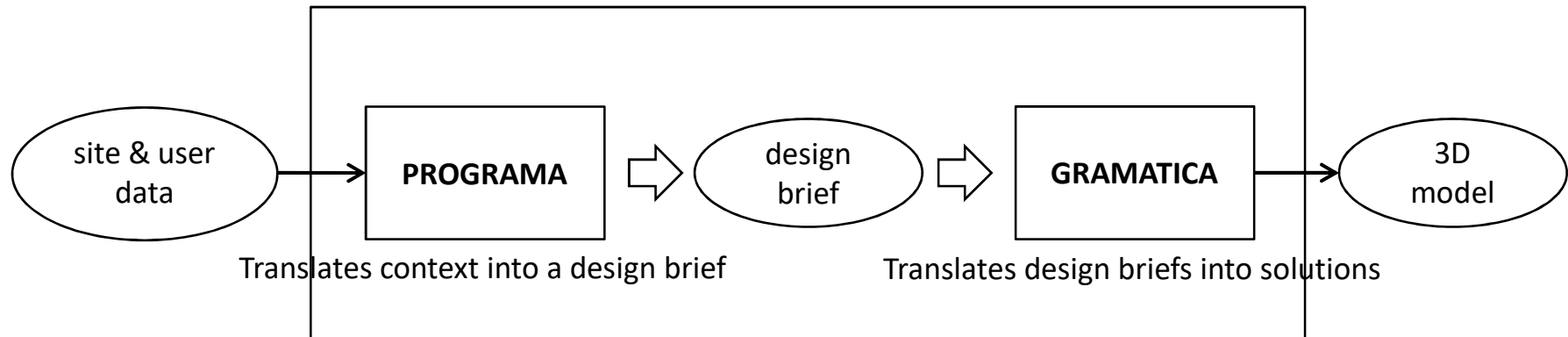
Discursive Grammar

Operative viewpoint



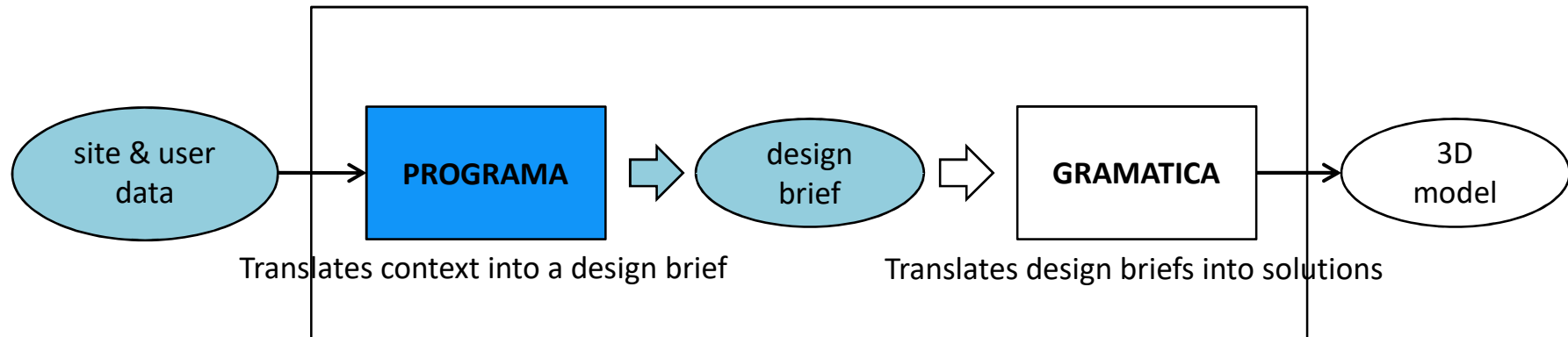
# Mass Customization Of Housing

## System Architecture



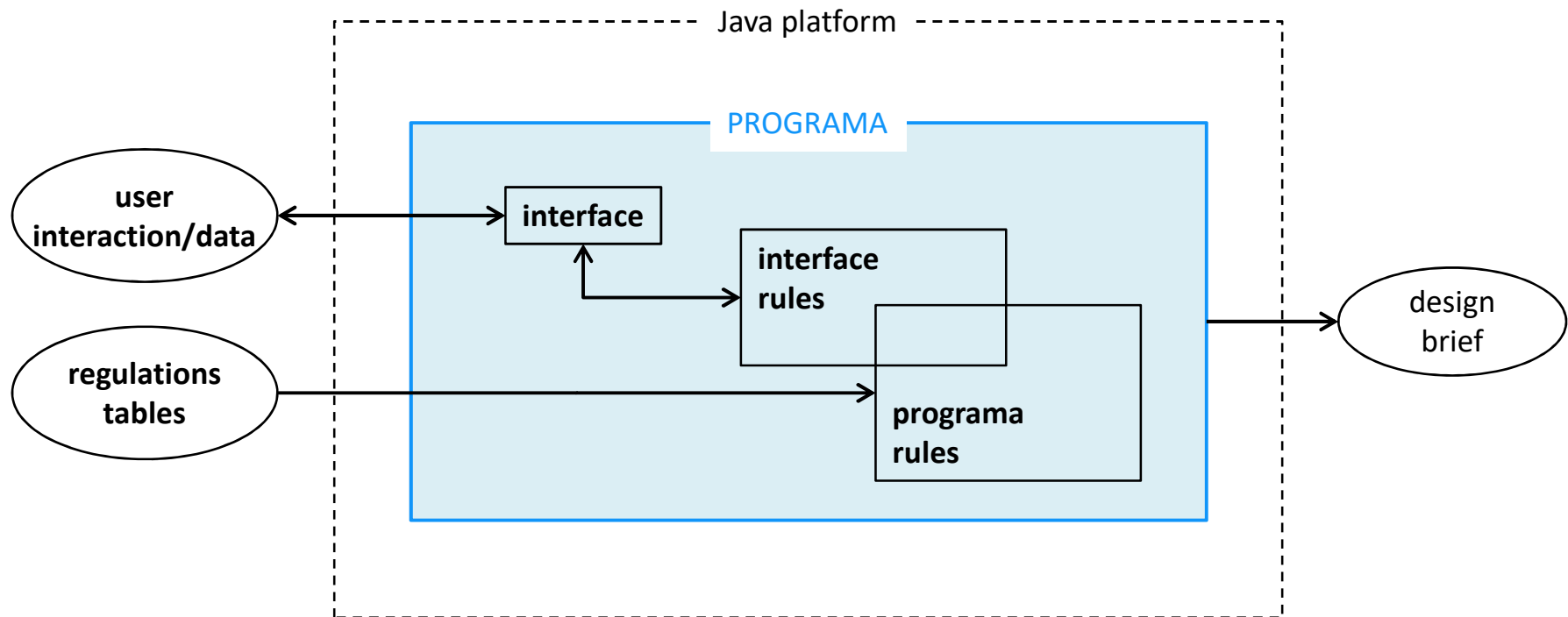
# Mass Customization Of Housing

## System Architecture



# Mass Customization Of Housing

## Programa Modules



# PROGRAMA

PAHPA Programmer

Context: Typology Morphology Spatiality Topology Aesthetics Weights info + view

Urban Context: Houses on both sides and back

Solar Orientation:

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	141	23.5	164.5	free
inhabitable	interior	exterior	useful	
0	0	0	0	used

$A_i / A_u$  (dwelling)       $A_u / A_g$

actual	minimum	actual	minimum
0	0.77	-	-

Cost

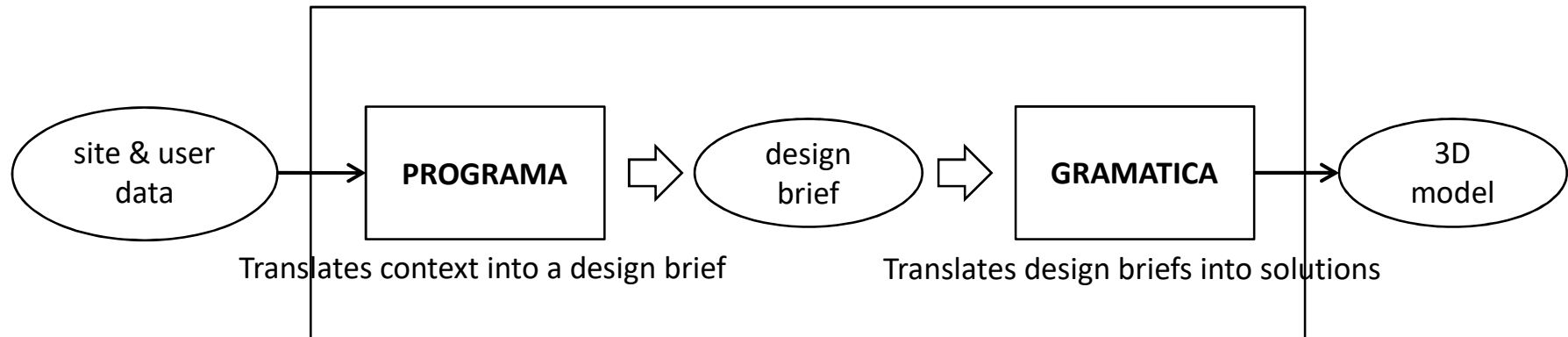
space type	cost / m2	current cost
	-	0.0

Quality

Current      [minimum]

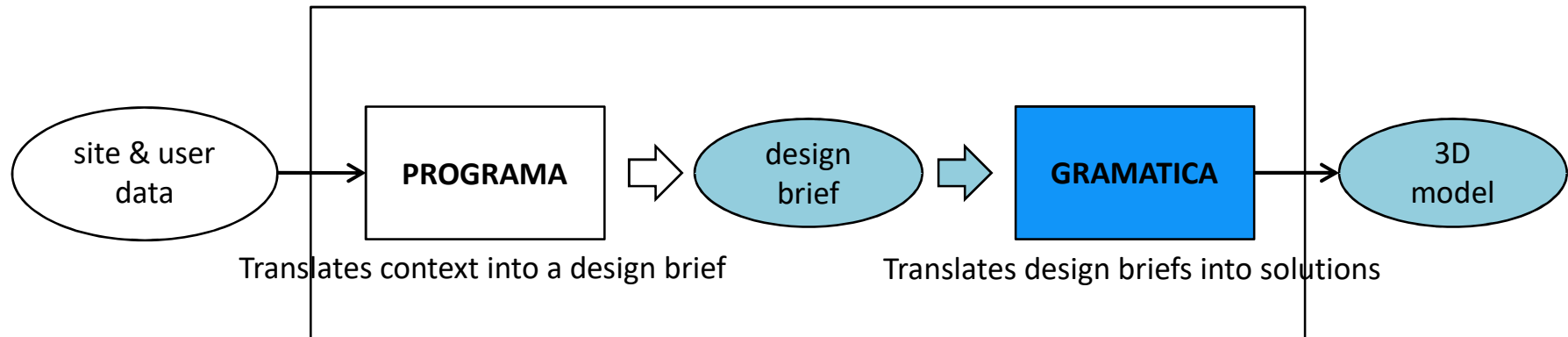
# Mass Customization Of Housing

## System Architecture



# Mass Customization Of Housing

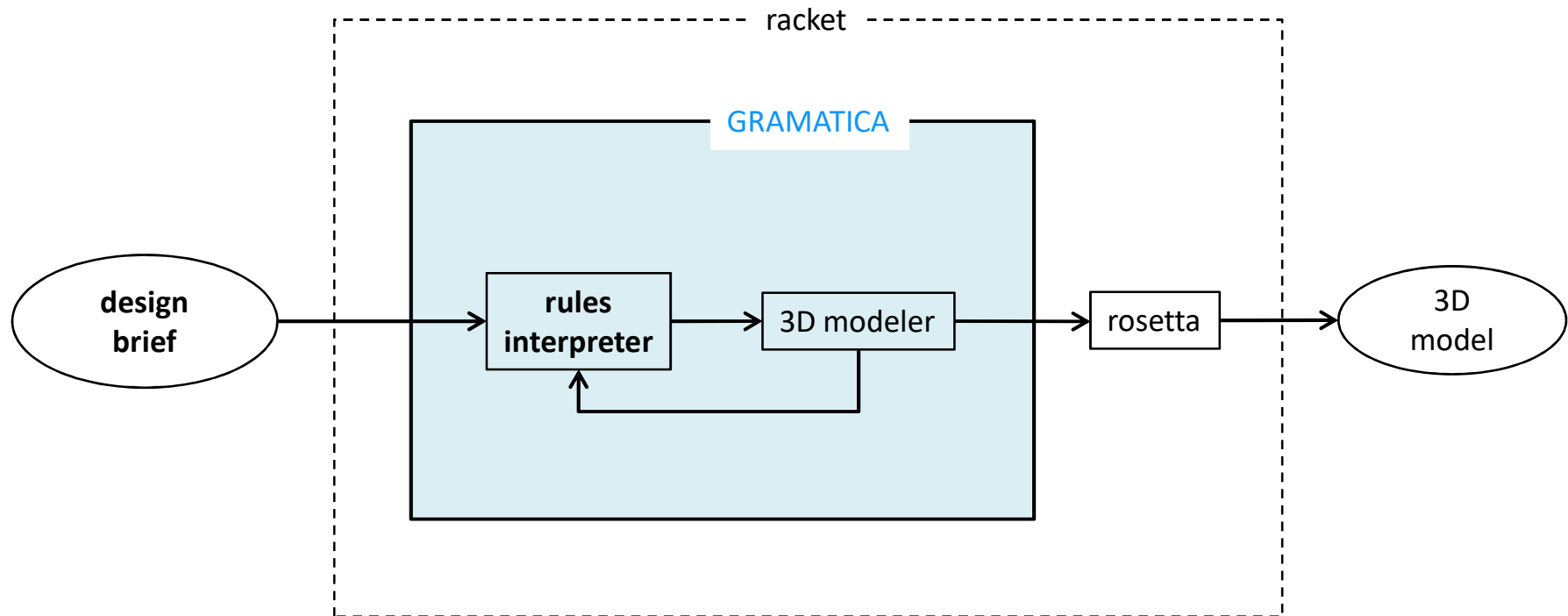
## System Architecture





# Mass Customization Of Housing

GRAMATICA Modules

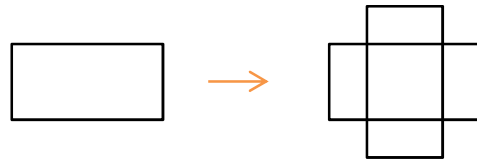


# Shape Grammars

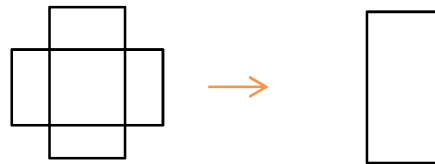
initial shape



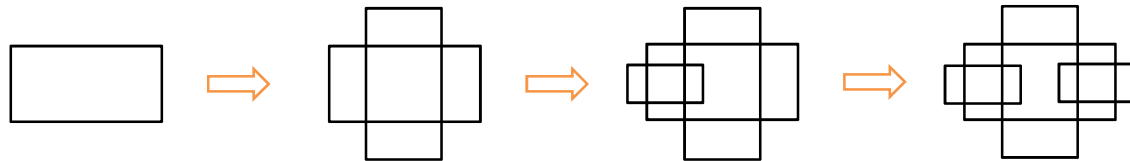
rule



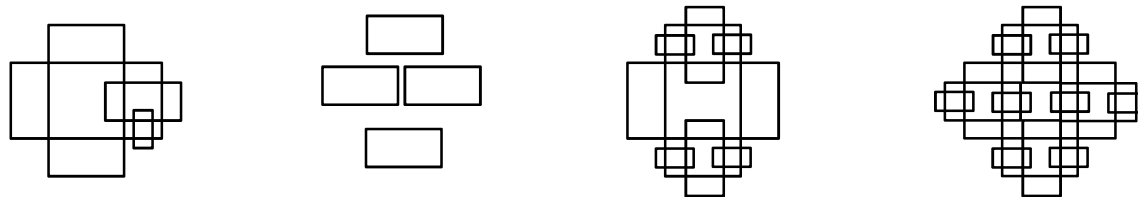
rule



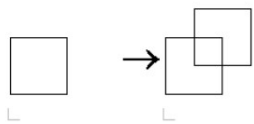
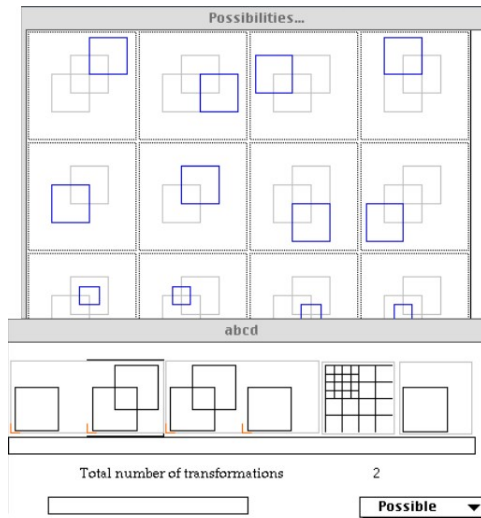
derivation



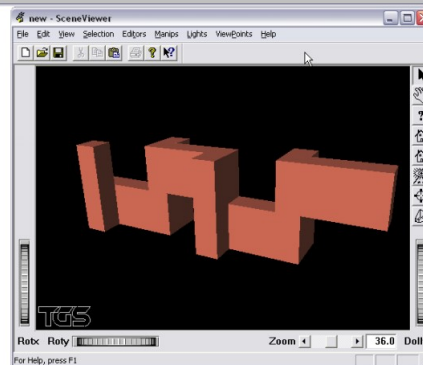
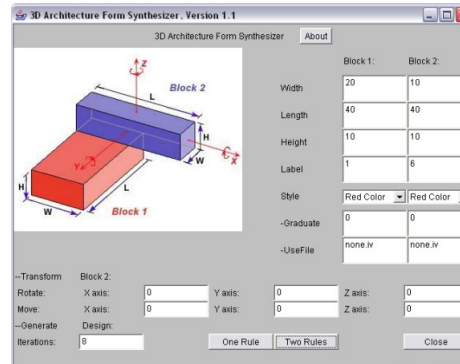
designs



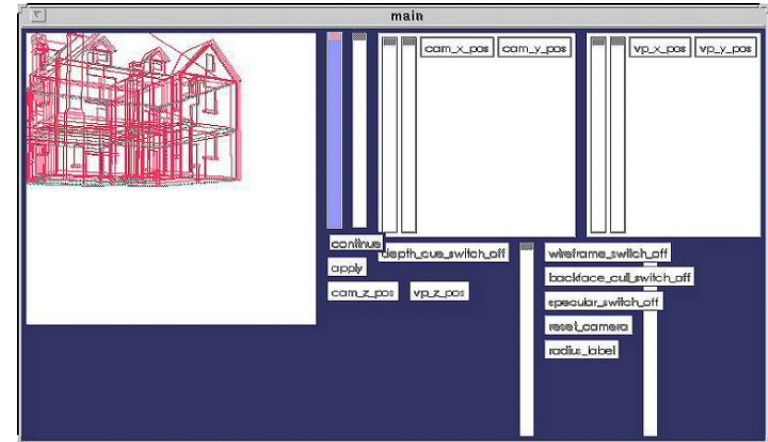
# Shape Grammar Interpreters



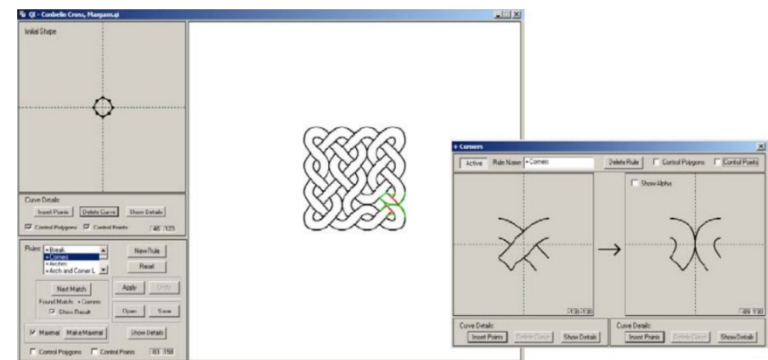
Gedit



3D shaper



Genesis



QI

# GRAMATICA

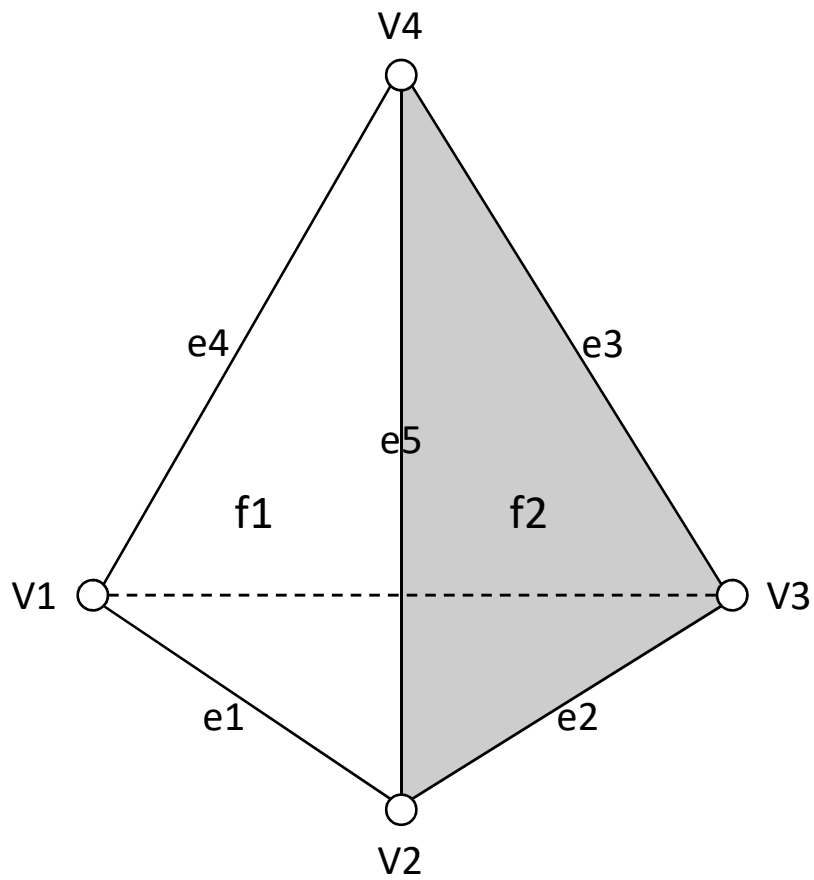
- 3D shape grammar interpreter

# GRAMATICA

- 3D shape grammar interpreter
- Focus:
  - Shape representation & generation
  - Rule representation, application & control
  - CAD bridge

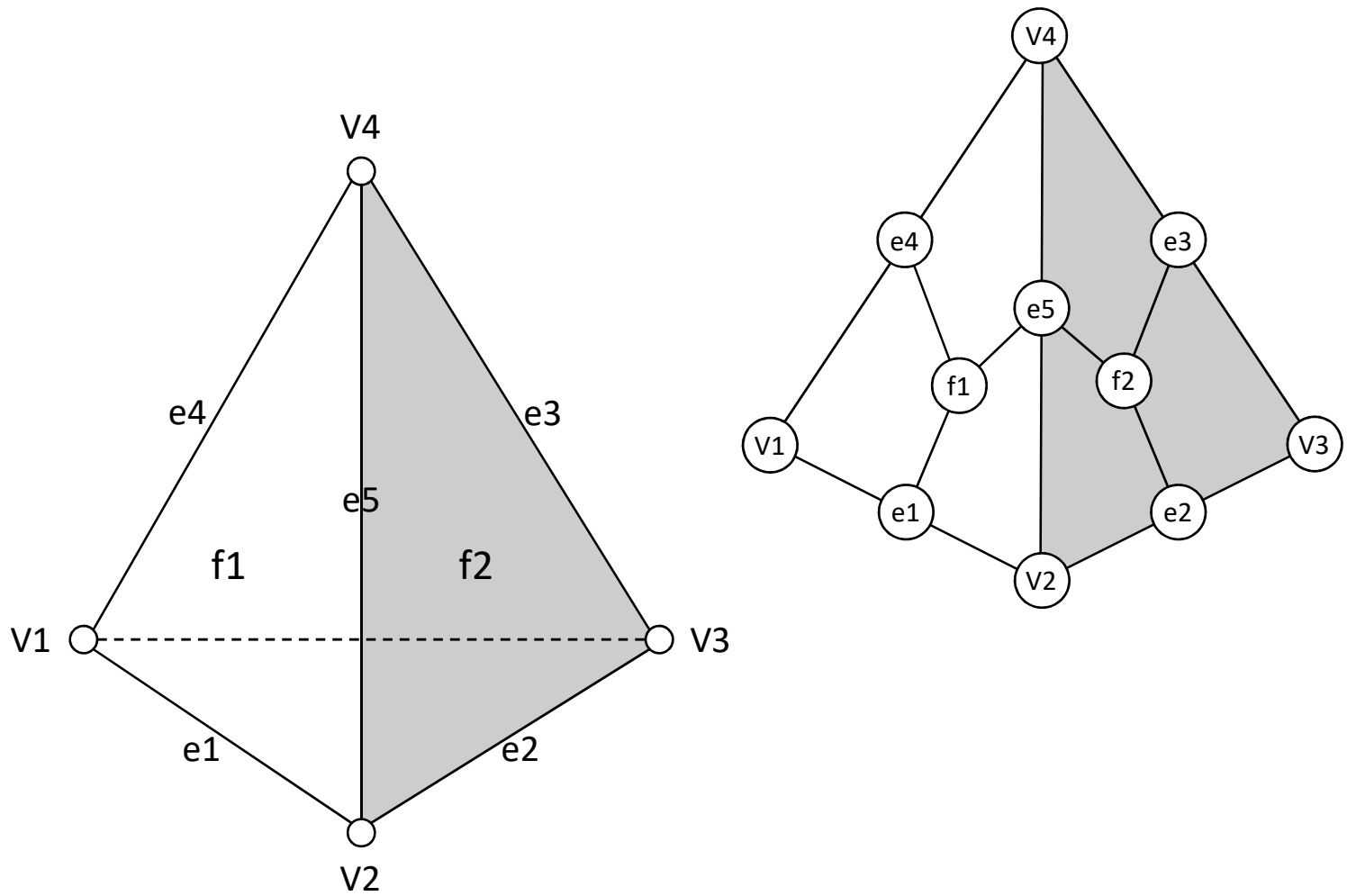
# GRAMATICA

Shape Representation



# GRAMATICA

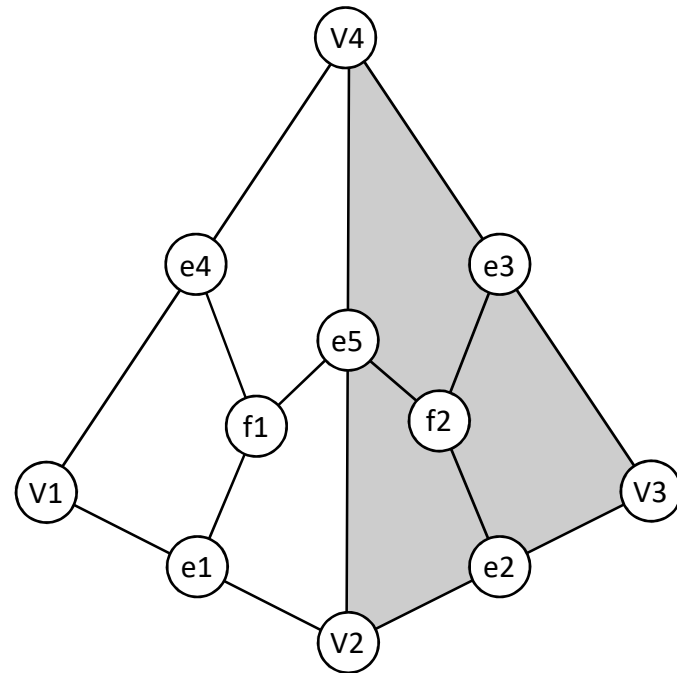
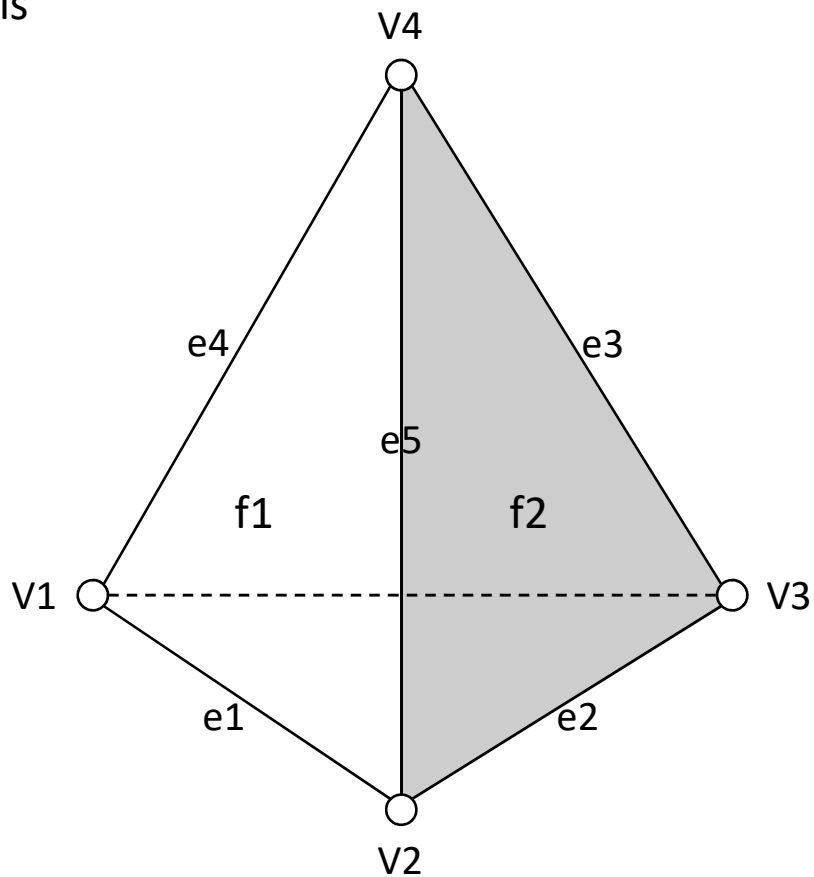
Shape Representation



# GRAMATICA

## Shape Representation

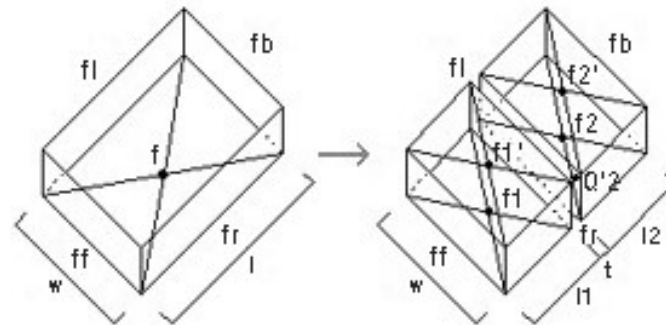
- Topology
- Geometry
- Labels





# GRAMMATICA

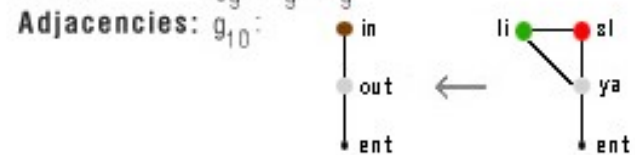
## Grammar Rule



R9:  $\langle F_1; f_b, f_r, f_f, li; o; Z \rangle \rightarrow$   
 $\langle F_1; f_b, f_r, f_f, f_l; ya, sl; Z - \{ya, sl\} \rangle$

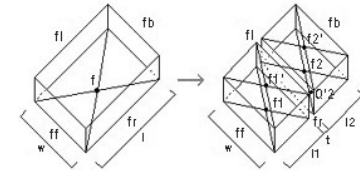
**Context:**  $g_4 : a_4 \leftarrow a_4$   
**Housetype:**  $g_5 : a_5 \leftarrow a_5$   
**N. rooms:**  $g_6 : a_6 \leftarrow a_6$   
**Balconies:**  $g_7 : a_7 \leftarrow a_7$   
**Zones:**  $g_8 : a_8 \leftarrow a_8 \leftarrow (in, (x,y), w, l, a),$   
 $+ (li, (x,y), w, l, a),$   
 $+ (sl, (x,y), w, l, a),$

**Room:**  $g_9 : a_9 \leftarrow a_9$



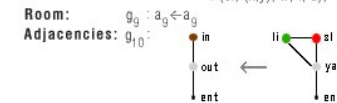
# GRAMATICA

## Rule Representation



R9:  $\langle F_1; f_b, f_r, f_l, l; o; Z \rangle \rightarrow$   
 $\langle F_1; f_b, f_r, f_l, f_j; ya, sl; Z - \{ya, sl\} \rangle$

**Context:**  $g_4 : a_4 \leftarrow a_4$   
**Housetype:**  $g_5 : a_5 \leftarrow a_5$   
**N. rooms:**  $g_6 : a_6 \leftarrow a_6$   
**Balconies:**  $g_7 : a_7 \leftarrow a_7$   
**Zones:**  $g_8 : a_8 \leftarrow a_8 \leftarrow (in, (x,y), w, l, a),$   
 $(li, (x,y), w, l, a),$   
 $(sl, (x,y), w, l, a).$

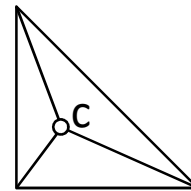
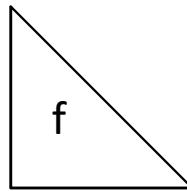


```
; house -> list of houses(new)
(define (rule:locate-inside-outside h)
  ; polyhedron facet house-extra -> list of houses(new)
  (define (locate-inside-outside p f e)
    (let* ((p2 (house-floor-2 h))
           (f2 (car (filter-facets-4-label p2 'f2))))
      ;;;
      (list (cond ((and is-frontyard has-balconies)
                   (new-house (intr-2-front p f 5 'out 'use)
                               (new-floor-2 (intr-2-front p2 f2 5 'out 'in))
                               e))
            ((and is-backyard has-balconies)
             (new-house (intr-2-front p f 7 'in 'out)
                       (new-floor-2 (intr-2-front p2 f2 7 'in 'out))
                       e))
            (is-backyard
             (new-house (intr-2-front p f 6 'in 'out)
                       (new-floor-2 (intr-2-front p2 f2 6 'in 'out))
                       e))
            (is-frontyard
             (new-house (intr-2-front p f 6 'out 'use)
                       (new-floor-2 (intr-2-front p2 f2 6 'out 'in))
                       e))))))
  ;
  (gen h 'f1 locate-inside-outside))
```

# GRAMATICA

## Rule Application & Control

grammar rule



$$c = f_{\text{center}} + f_{\text{normal}} * d$$

rule application  
depth first



rule application  
breadth first



original tetrahedron



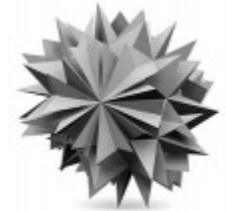
d = 0.5



d = 0.3



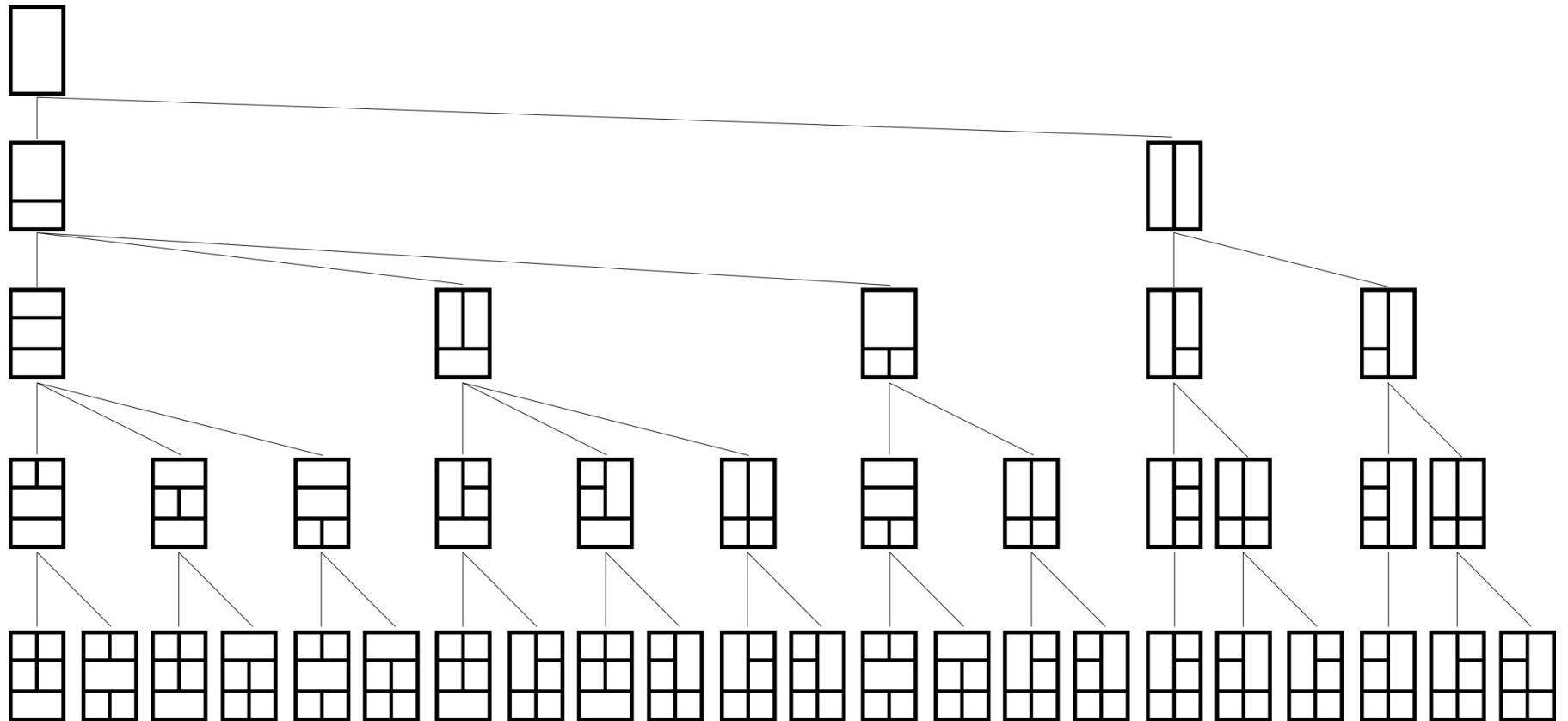
d = 0.1



d = 0.5

# GRAMATICA

Design Derivation Tree



# GRAMATICA

3D Model

# GRAMATICA

## Conclusions

- Shape/rule representation
- Portable across CAD applications

# GRAMATICA

## Conclusions

- Shape/rule representation
- Portable across CAD applications
- **Shape grammars interpreters are difficult to implement**

Thank you

Questions?