Interaction-Oriented Programming for Cockpits and Controller Working Positions

Stéphane Chatty
Stéphane Conversy
Jérémie Garcia
Sébastien Leriche
Mathieu Magnaudet
Célia Picard
Daniel Prun
Nicolas Saporito

The **Smala** programming language
The **Djnn** framework and run-time

Interaction-oriented programming


F-PALG

Those systems are all interactive

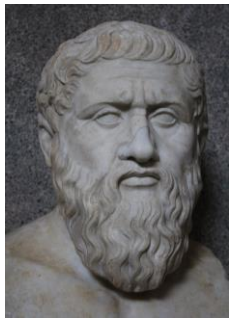
- Many interacting sub-systems
- Whose state depends on multiple asynchronous sources of event
- In which some sub-systems are human beings

djnn: a process-based approach

Plato

the world as *objects* with *properties*

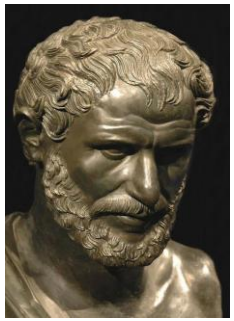
=> Object-Oriented Programming



Heraclitus

the world as *processes* that *change*

=> **Interaction-Oriented Programming**



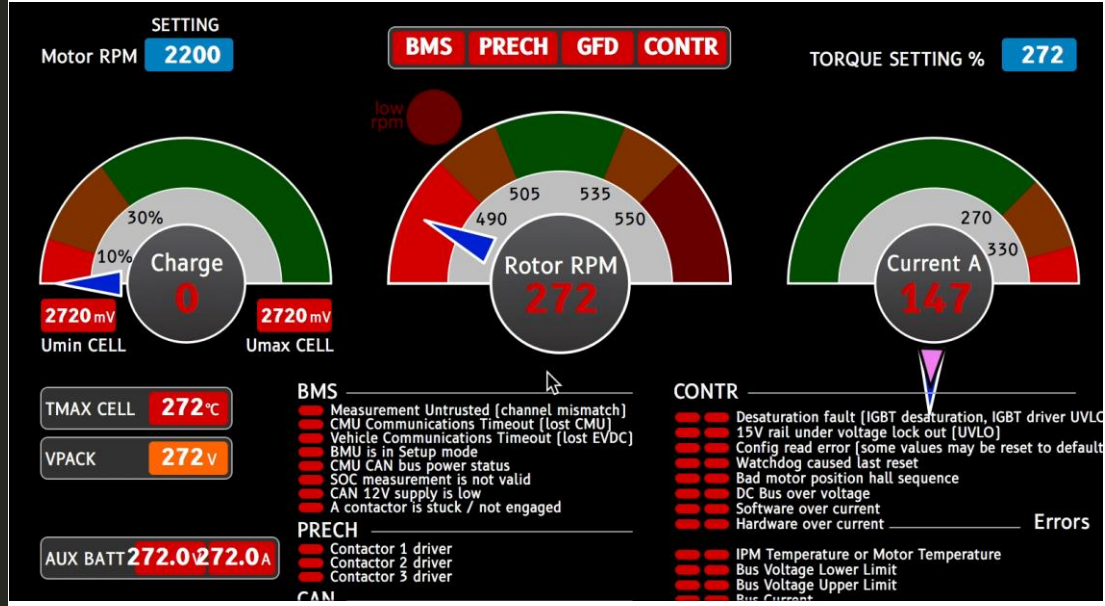
```
// page management
page = find(img.lower_panel.right.page)

addChildrenTo page {
  Component page_up
  Component page_down

  Switch status_control(page1) {
    page1 << svg.page_def.page1
    page2 << svg.page_def.page2
    page3 << svg.page_def.page3
  }
  FSM sc_fsm {
    State page2
    State page3
    State page1
    page1 -> page2 (page_up)
    page2 -> page3 (page_up)
    page3 -> page1 (page_up)
    page1 -> page3 (page_down)
    page2 -> page1 (page_down)
    page3 -> page2 (page_down)
  }
  sc_fsm.state => status_control.state
}

// mouse page down / page up
Component mouse_page_up_page_down {
  NoOutline noo
  FillColor black(0,0,0)
  Translation t(340, 265)
  Rectangle page_down(0, -5, 60, 20, 5, 5)
  Rectangle page_up(60, -5, 60, 20, 5, 5)
  page_down.press -> page.page_down
  page_up.press -> page.page_up
}

// gpio page down / page up
Component page_gpio {
  gpio_down = find(gpio:B4.in)
  gpio_up = find(gpio:B23.in)
  gpio_down -> page.page_down
  gpio_up -> page.page_up
}
```



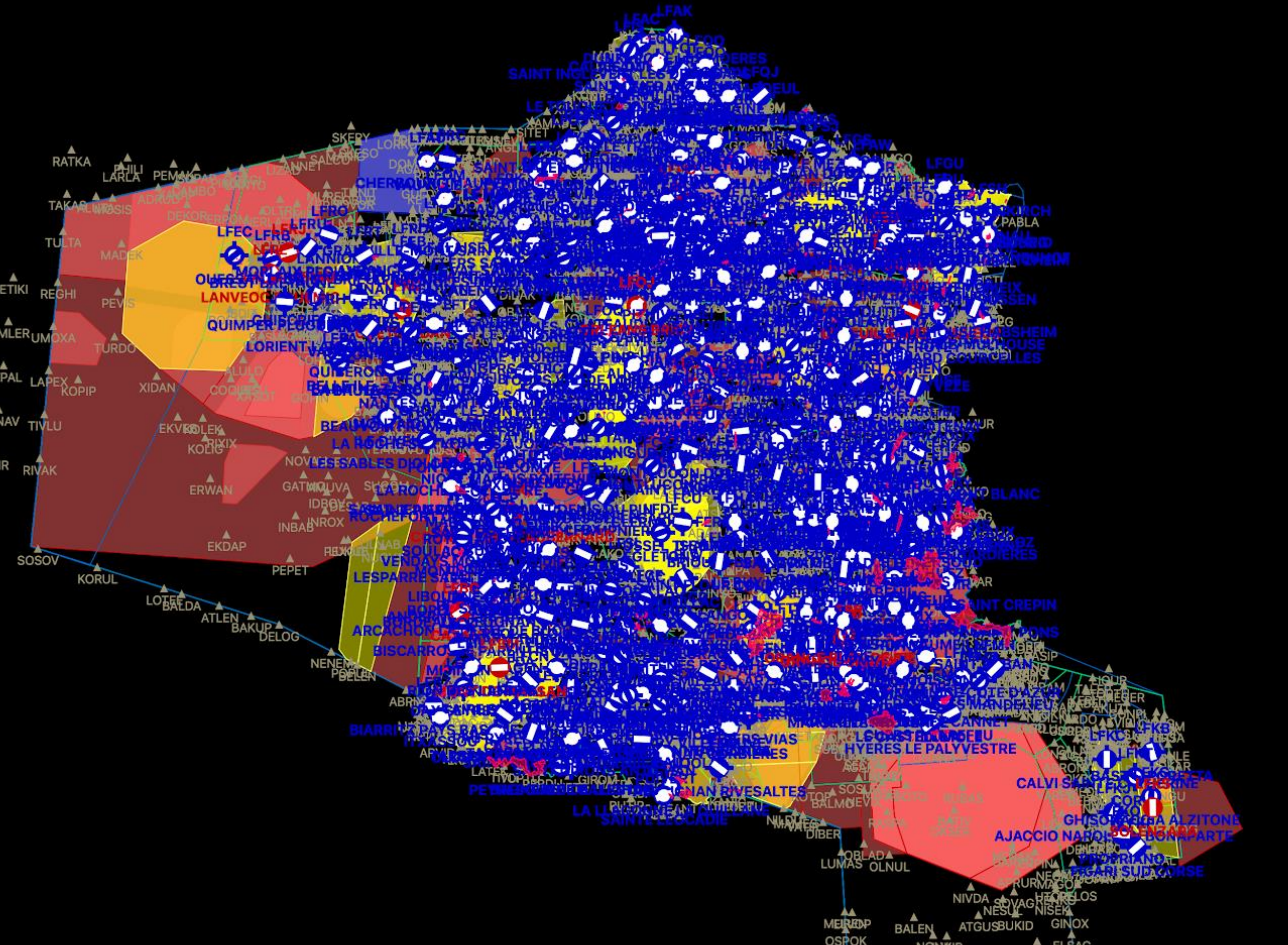
IOP: Input from graphics and GPIO
Unification: Interchangeable input
Iterative design: Switch Debug/Operation

```
1  ATT aka pfd.ATT
2
3  FSM pfd_mode {
4
5      State occidental {
6                      zero =: ATT.aeroplane_roll
7          -aeroplane_roll => ATT.horizon_roll
8          -aeroplane_roll => ATT.pointer_roll
9                      zero =: ATT.bankscale_roll
10         -aeroplane_roll => ATT.pitchscale_roll
11                 -radius =: ATT.slip_trans_y
12                    one =: ATT.pointer_opacity
13                    one =: ATT.slip_opacity
14     }
15
16     State russian {
17          aeroplane_roll => ATT.aeroplane_roll
18                     zero =: ATT.horizon_roll
19                     zero =: ATT.pointer_roll
20                     zero =: ATT.bankscale_roll
21                     zero =: ATT.pitchscale_roll
22               //-10+zero =: ATT.slip_trans_y // agnost
23                     zero =: ATT.pointer_opacity
24                     zero =: ATT.slip_opacity
25     }
26
27     State newpfd {
28                     zero =: ATT.aeroplane_roll
29         -aeroplane_roll => ATT.horizon_roll
30                     zero =: ATT.pointer_roll
31         -aeroplane_roll => ATT.bankscale_roll
32         -aeroplane_roll => ATT.pitchscale_roll
33                -10+zero =: ATT.slip_trans_y
34                     zero =: ATT.pointer_opacity
35                     one  =: ATT.slip_opacity
36     }
37
38 }
```

IOP: States - assignment
Unification:
Iterative design:

Status

- Already working:
  - « Static » interfaces (no new objects during run-time)
  - Graphics, SVG, direct manipulation, multitouch, network
  - Suitable for research projects
- Work-in-Progress
  - Dynamic interfaces
  - Scalability, latency, dependability
  - Suitable for prototyping projects and/or non-critical applications (SWAL>=4)


LFAK
LFAC
SAINT INGLEVERT
SKERY
SITET
RATKA
LARLA
TAKAS
TULTA
MADEK
REGHI
PEVIS
UMOXA
TURDO
LAPEX
KOPIP
XIDAN
TIVLU
RIVAK
ERWAN
EKDAP
SOSOV
KORUL
PEPET
BALDA
ATLEN
DELOG
NENEM
LFEC
LFRB
LFRO
LANNION
LANVEOC
LORIENT
LESPARRE
ARCACHON
LFOU
PABLA
MULHOUSE
SAINT CREPIN
HYERES LE PALYVESTRE
PERPIGNAN RIVESALTES
SAINTE LEOCADIE
LFKB
LFKC
SOLENZARA
FIGARI SUD CORSE
DIBER
LUMAS
OLNUL
NIVDA
BALEN
ATGUS
BUKID
GINOX
OSPOK

smala.io

https://github.com/lii-enac/djnn-cpp
https://github.com/lii-enac/smala