Interactive Informatics

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ENAC

GDR IHM - GT GSI - juillet 2025

https://lii.enac.fr/
stephane.conversy@enac.fr

E/C HMI


Brock, *M2IHM*


Garcia,


Letondal, (HDR, Head)


Matton,


Pauchet, *M2IHM*


Vo *M2IHM*

Head


Conversy (HDR)

E/C ISE


Ben Khalifa,


Garoche, (HDR, head)


Pesin,


Picard,


Prun

PhD students


Boonyard


Bornes


Dubessy


Dubus *M2IHM*


Duchevet *M2IHM*


Gaillet *MIATSED*


Gao


Grossetête


Latarche


Rocher


Bortolussi


Bellanger


Haag *IENAC*


Janak


Marcon *M2IHM*


Ongalé -Obeyi


Wilandy *MIATSED*

Associate Researchers and Engineers

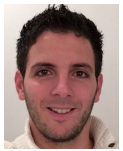

Chatty, (DSNA)


Saporito (Ens) *M2IHM*

PostDoc


Battut *M2IHM*

Engineers


Peyruqueou,

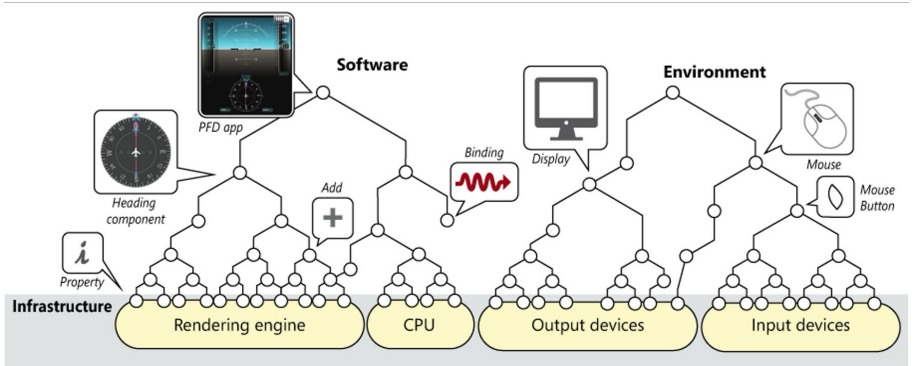

Poirier

10 E/C 2 **C€** 1**PD€** **1** IR **16** PhD **1** IR

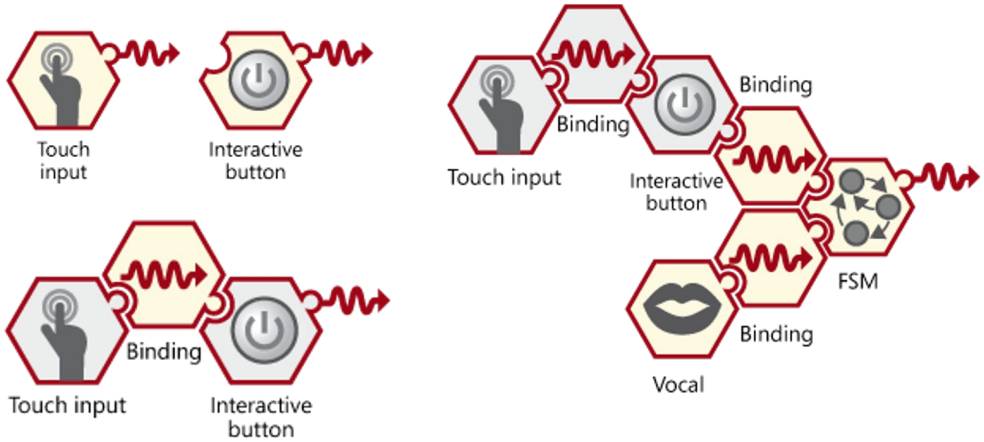
SESAR ASTAIR: Auto steer taxi at airport

Interaction with Automation

Jérémie Garcia, Mathieu Cousy, Alexandre Battut, Dong-Bach Vo, Vincent Peyruqueou



« System » = fractal collection of interactive processes



Execution model = activation of coupled processes

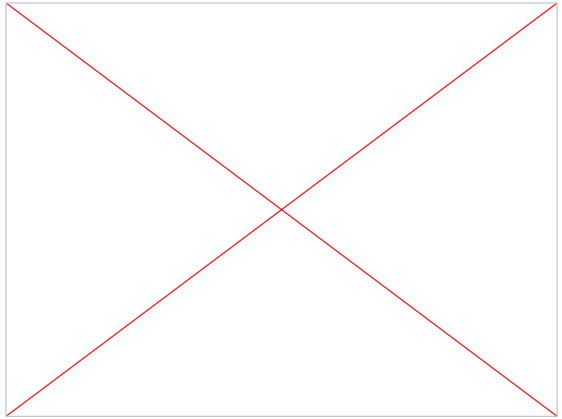
```
4     addChildrenTo page {
5       Component page_up
6       Component page_down
7
8       Switch status_control(page1) {
9         page1 << svg.page_def.page1
10        page2 << svg.page_def.page2
11        page3 << svg.page_def.page3
12      }
13      FSM sc_fsm {
14        State page2
15        State page3
16        State page1
17        page1 -> page2 (page_up)
18        page2 -> page3 (page_up)
19        page3 -> page1 (page_up)
20        page1 -> page3 (page_down)
21        page2 -> page1 (page_down)
22        page3 -> page2 (page_down)
23      }
24      sc_fsm.state => status_control.state
25    }
26
```

Smala dans ISI, applications "réalistes" dans IHM et ISI

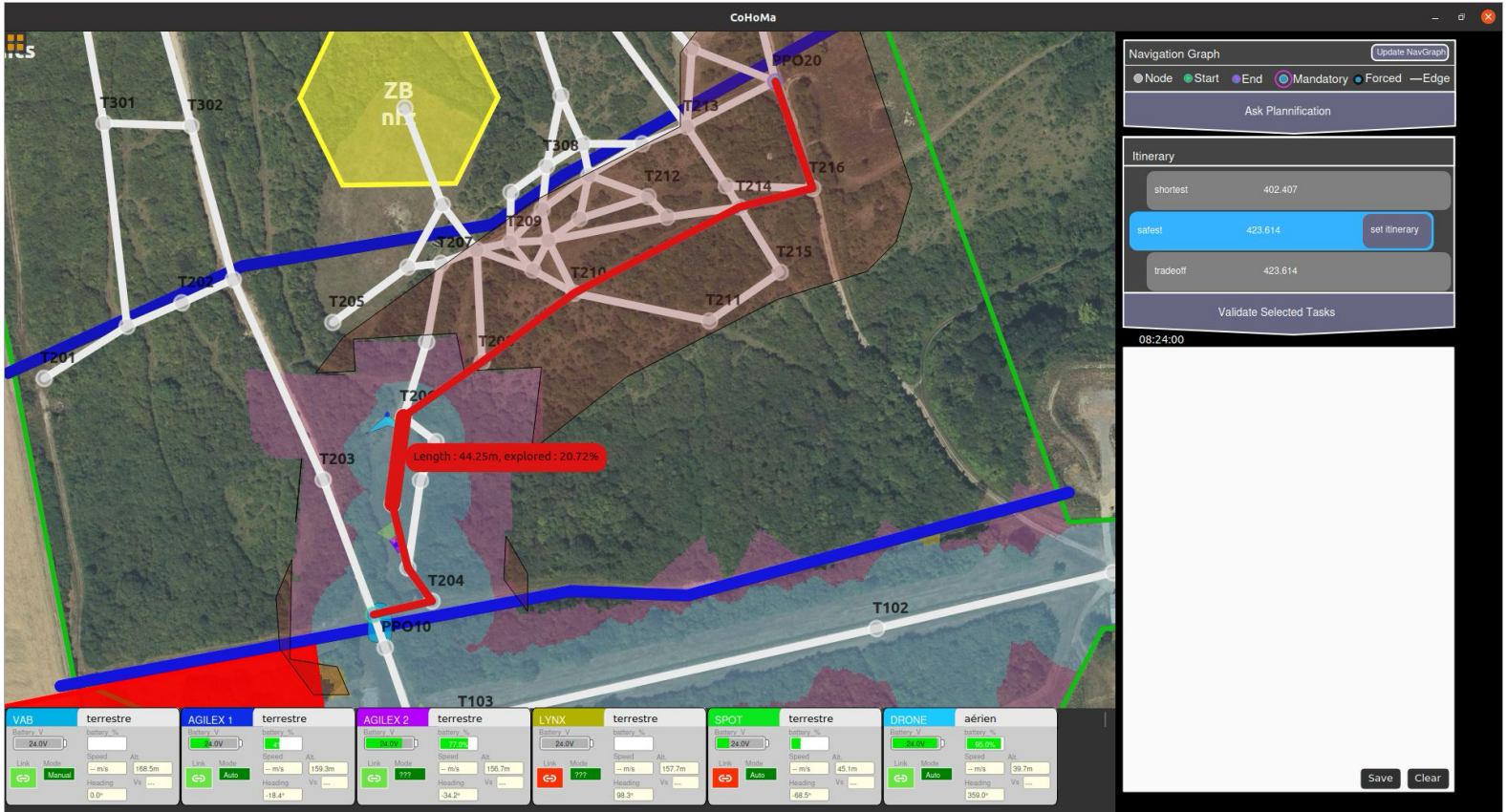
Réalistes : "suffisamment complexes" pour démontrer la validité écologique

Mais cela a un grand coût de développement/ressources…

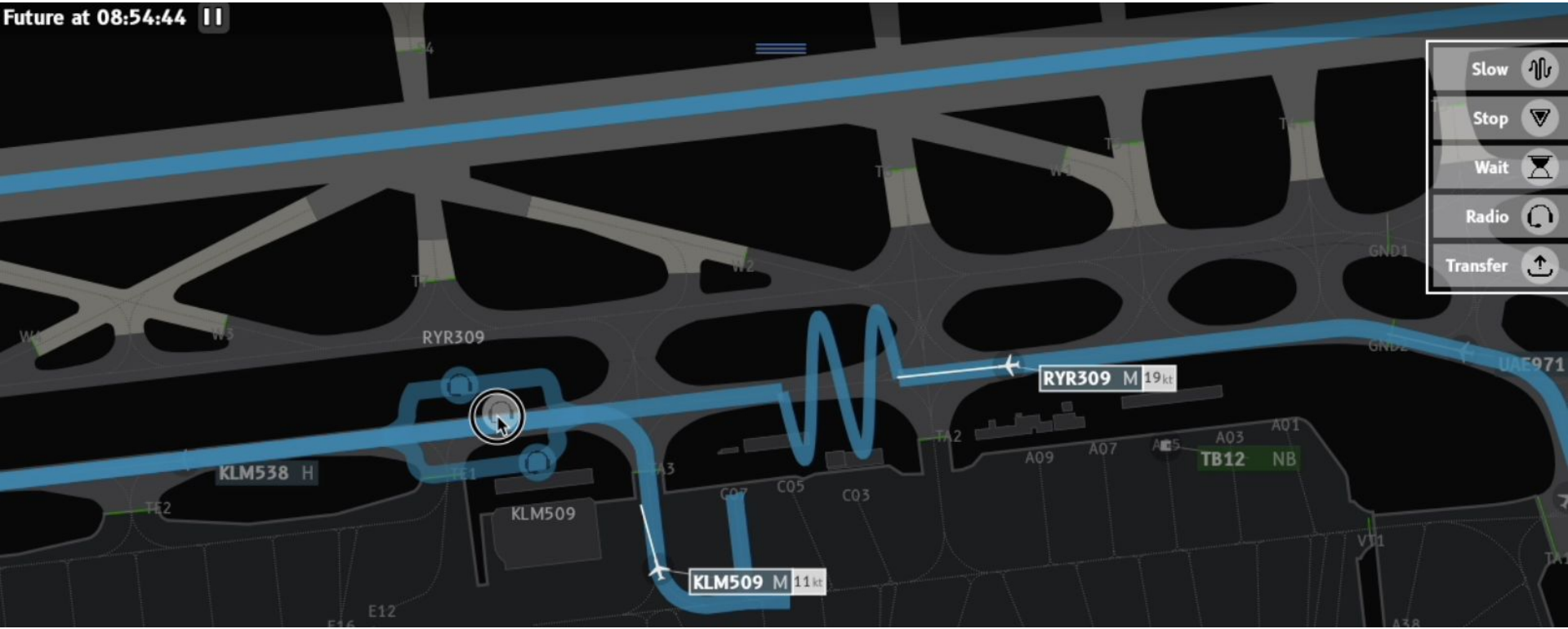
Perspectives : maintenant qu'on a les apps, s'en servir pour raisonner…



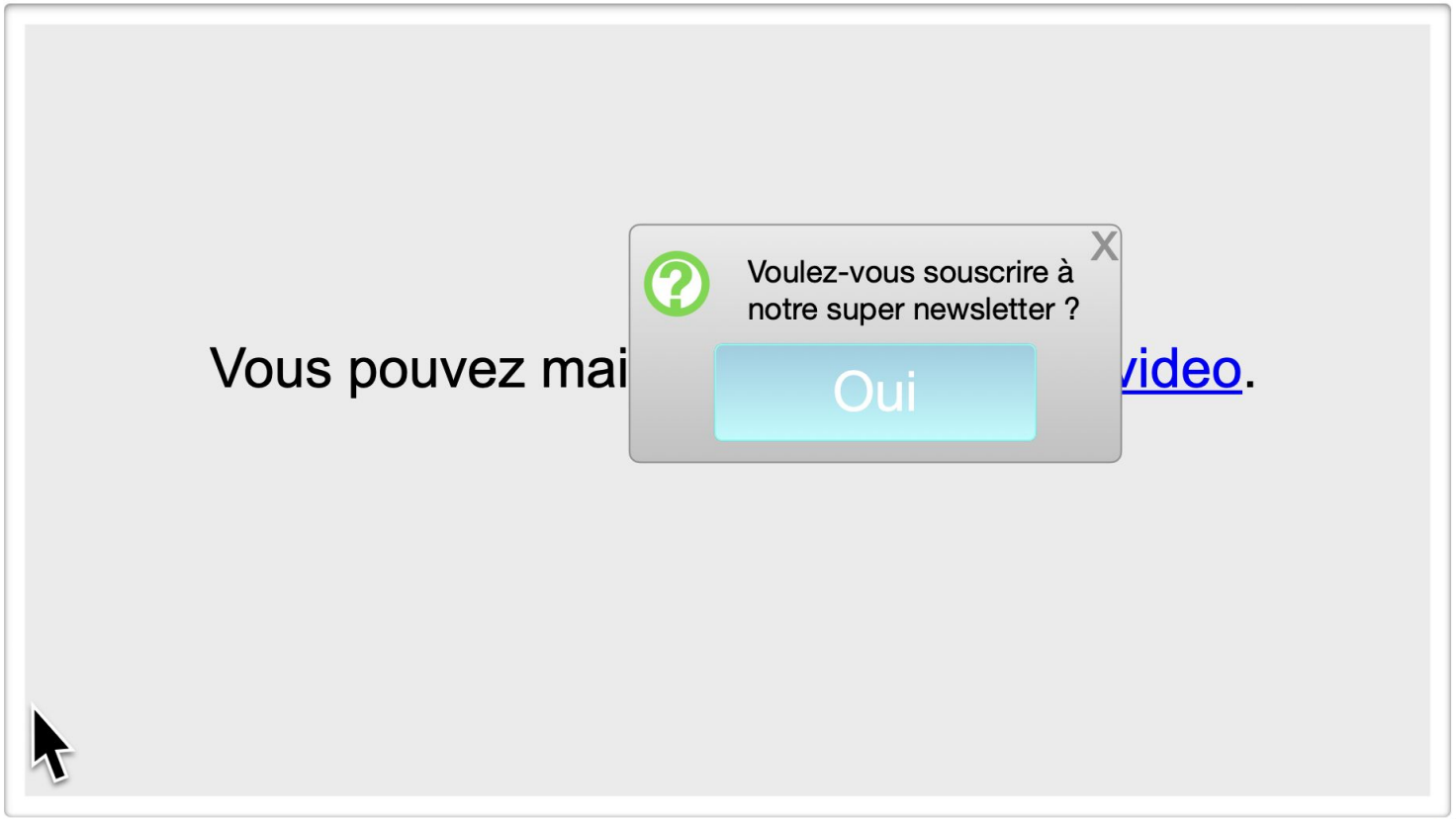
Pb Example: How to reuse a "diffuse" chatbox?



Pb Example: How to D'n'D over hidden object?



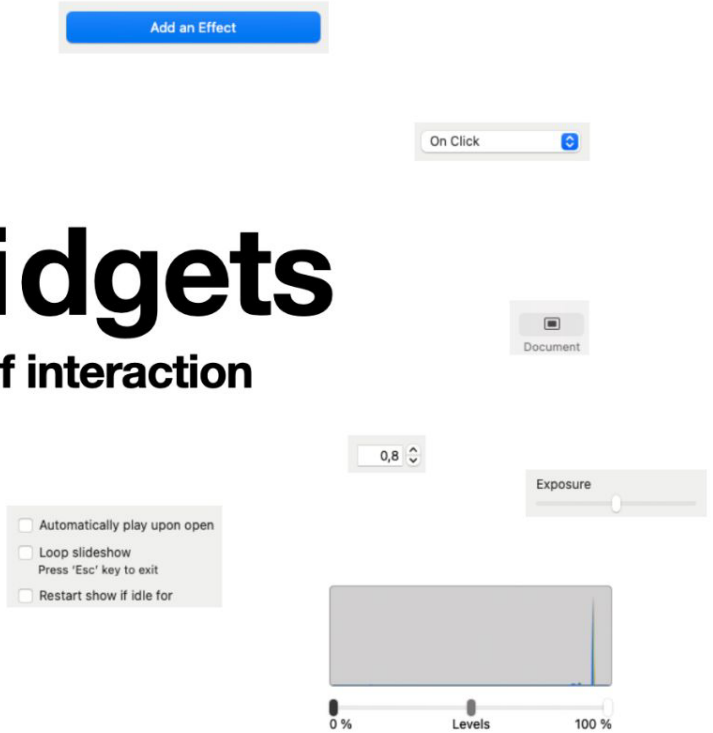
Pb example: interaction interference



knowdget

Knowledge widgets

Widgets supporting knowledge of interaction



Knowledge of…

1. Spatial and temporal ecosystem
    - Not limited to straight rectangle, with a background and clipping…
    - What's next to me (as a widget)? N,S,W,E ? Above ? Below ?
    - What was here before?
2. Human capacities (motor, perception, cognition)
    - Including (transient) disabilities, device accuracy etc.
3. How to communicate ways of interacting
4. Interaction context
5. Interaction history

Aim/purpose, mission, objectives (finalité, mission, objectifs)

Aim / purpose:

- Provide more usable interactive systems to end-users

Mission:

- Provide more usable and more reusable interactive components to designers/programmers

Objectives:

- Redefine the notion of atomic elements that compose GUIs to evolve beyond Widgets (Conceptual contribution) - T1
- Invent new control structures of Knowdget and provide programming APIs and code examples (Technical contribution) - T2
- Provide evidence of the benefits of the Knowdgets approach (empirical contribution) - T3

interaction composition

```
def add_interaction(self, interaction):
    self.interaction = (self.interaction + interaction).owned_by(self)
    self.hasClickListener = True
    self.canvas.update()

def mouse_down(self):
    self.interaction.press()

def mouse_up(self):
    self.interaction.release()
```

no adhoc event handling

Consortium

- Centre Inria de l'Université de Lille (coordinator)

  - Loki




- ENAC (Toulouse)

  - LII



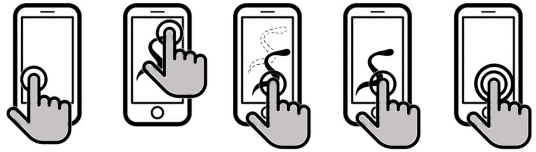
+ 1 Ph.D., 1 engineer (18 months), 1 apprentice (24 months) and interns

Concepts and tools for interactive computing

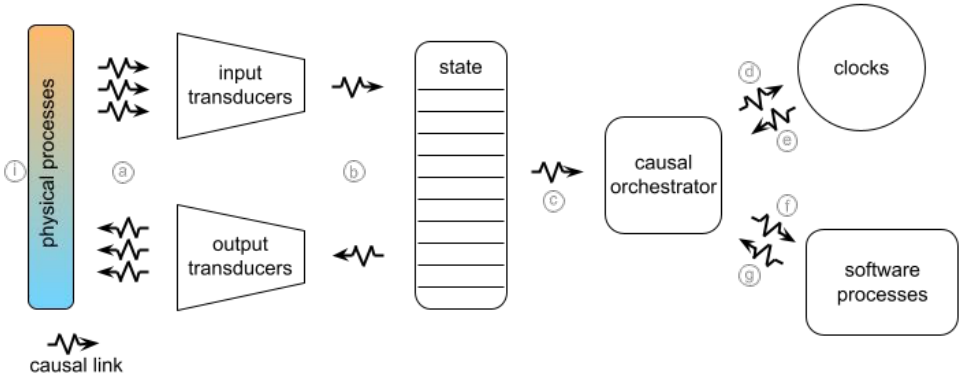
Alice Martin

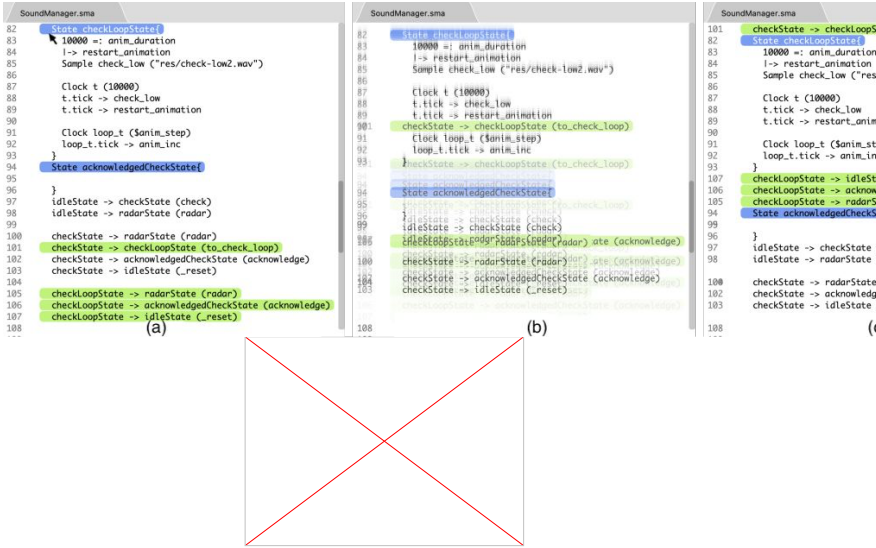
(Informatique + Philosophie)

RQ : qu'est-ce que l'informatique interactive ?



Thèse : l'informatique interactive est concernée par les relations de **causalité**.

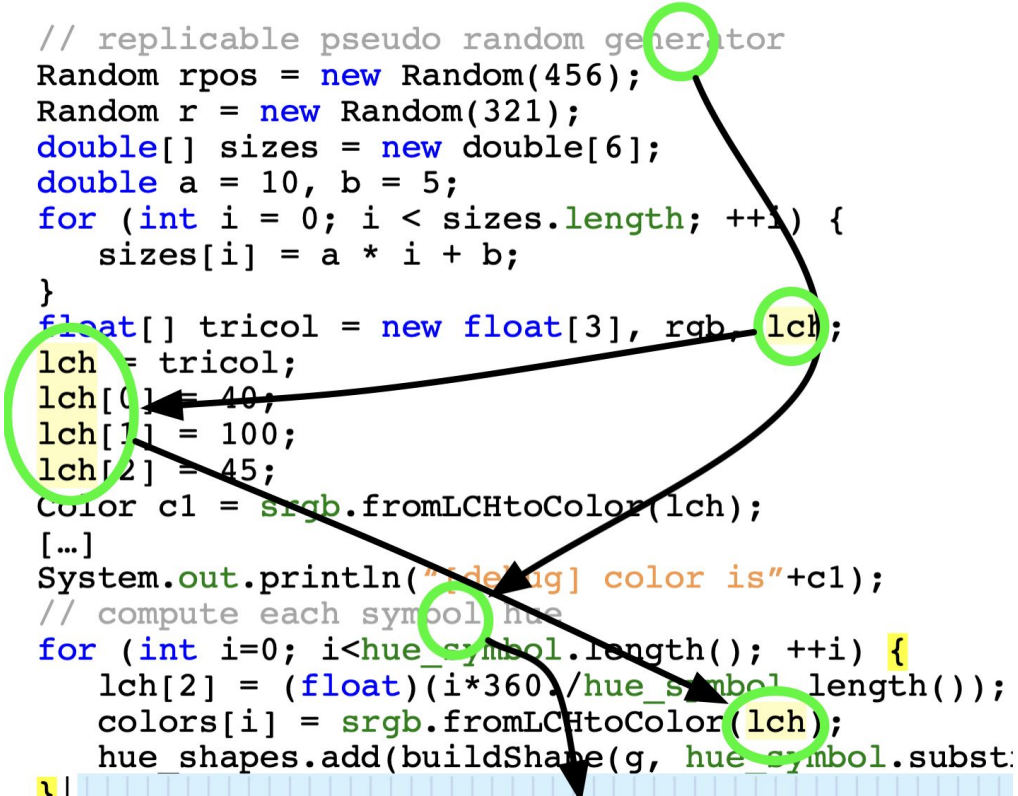




Il faut un modèle de son **exécution:**
une **machine interactive**

Il faut des outils pour comprendre les relations de causalité**: causette**

Unifying textual and visual programming with Semiology of Graphics



color is selective

task: «find all occurrences of variable ‘lch’» ?!

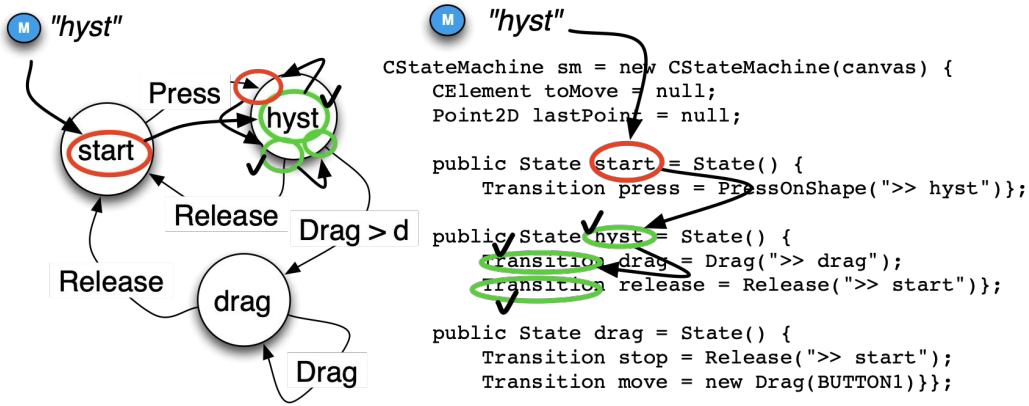
luminosity is selective

task: «seek and navigate among code and comment»

luminosity is ordered

task: «find important text, ignore other text»

Unifying textual and visual programming with Semiology of Graphics



task: “what are the ’out’ transitions for state ’hyst’?”

Stéphane Conversy. 2014. Unifying Textual and Visual: A Theoretical Account of the Visual Perception of Programming Languages. In Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software (Onward! 2014). Association for Computing Machinery, New York, NY, USA, 201–212. https://doi.org/10.1145/2661136.2661138

Preuves sur les langages pour l'interaction

1ère approche :

Identification des bigraphes de Milner comme un modèle sémantique approprié pour la description d'interactions graphiques

un graphe de composition d'une scène interactive (activation/désactivation hiérarchique)

un graphe de relations causales entre composants

Traduction de programmes smala (ou QML) en bigraphes, et utilisation des propriétés des bi-graphes pour prouver des comportements (à moyen terme)

à + court terme, Rocq-ification des preuves de bi-graphe

=> Célia Picard, Cyril Allignol, thèses de Nicolas Nalpon et Cécile Macron

Preuves sur les langages pour l'interaction

2ème approche :

Compilateur smala "vérifié"

Écrit en OCaml pour l'instant

À terme preuves sur les règles de transformation avec Rocq

=> sujet de thèse, recherche de candidat.e

=> Basile Pesin, Célia Picard, Cyril Alignol, Stéphane Conversy

State Update – Rule

$$\frac{\begin{array}{c} T = \{event\} \cup \{ev \mid E, A, T, E', A' \vdash^{\text{proc}}_{\text{prop}} proc \Downarrow ev\} \\ E, T, E', A' \vdash^{\text{proc}}_{\text{safe}} proc_\epsilon \\ E, A, T \vdash_{\text{update}} proc \Downarrow E', A' \end{array}}{E, A \vdash_{\text{react}} proc(event) \Downarrow E', A', (T \cap \{\texttt{Trigger}\ p \mid \texttt{Spike}\ p \in proc\})}$$

$$\frac{\begin{array}{c} \forall p\ v,\ \texttt{Assign}\ v\ p \in T \implies E'(p) = \lfloor v \rfloor \\ \forall p,\ (\forall v,\ \texttt{Assign}\ v\ p \notin T) \implies E'(p) = E(p) \\ \forall p,\ \texttt{Activate}\ p \in T \implies p \in A' \\ \forall p,\ \texttt{Deactivate}\ p \in T \implies p \notin A' \\ \forall p,\ (\texttt{Activate}\ p \notin T \wedge \texttt{Deactivate}\ p \notin T) \implies (p \in A' \iff p \in A) \end{array}}{E, A, T \vdash_{\text{update}} proc \Downarrow E', A'}$$

Some contradictory programs:

```
Int x 0;
a -> a1; a1: 42 =: x;
a -> a2; a2: 84 =: x;
```

```
Component c { ... }
a -> c;
a ->! c;
```

Rejected by static analysis