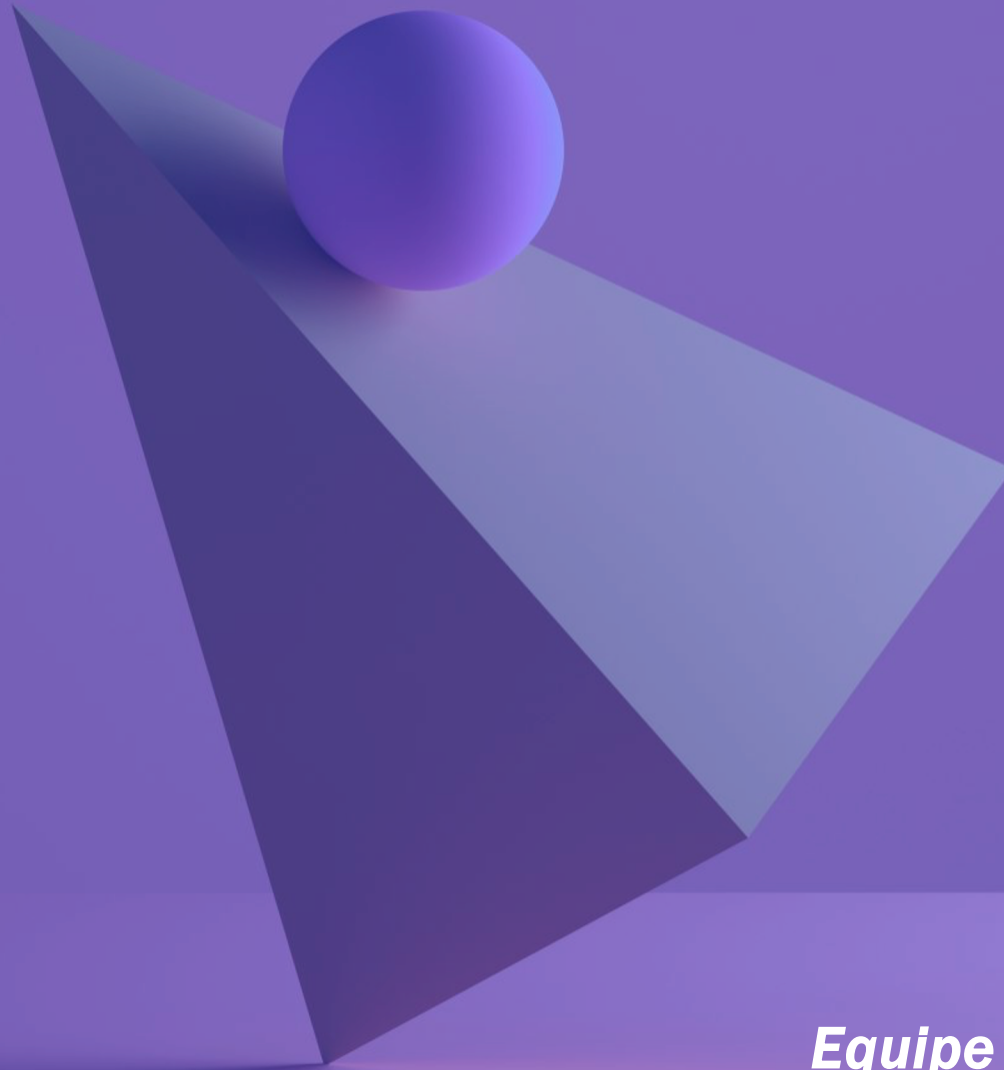
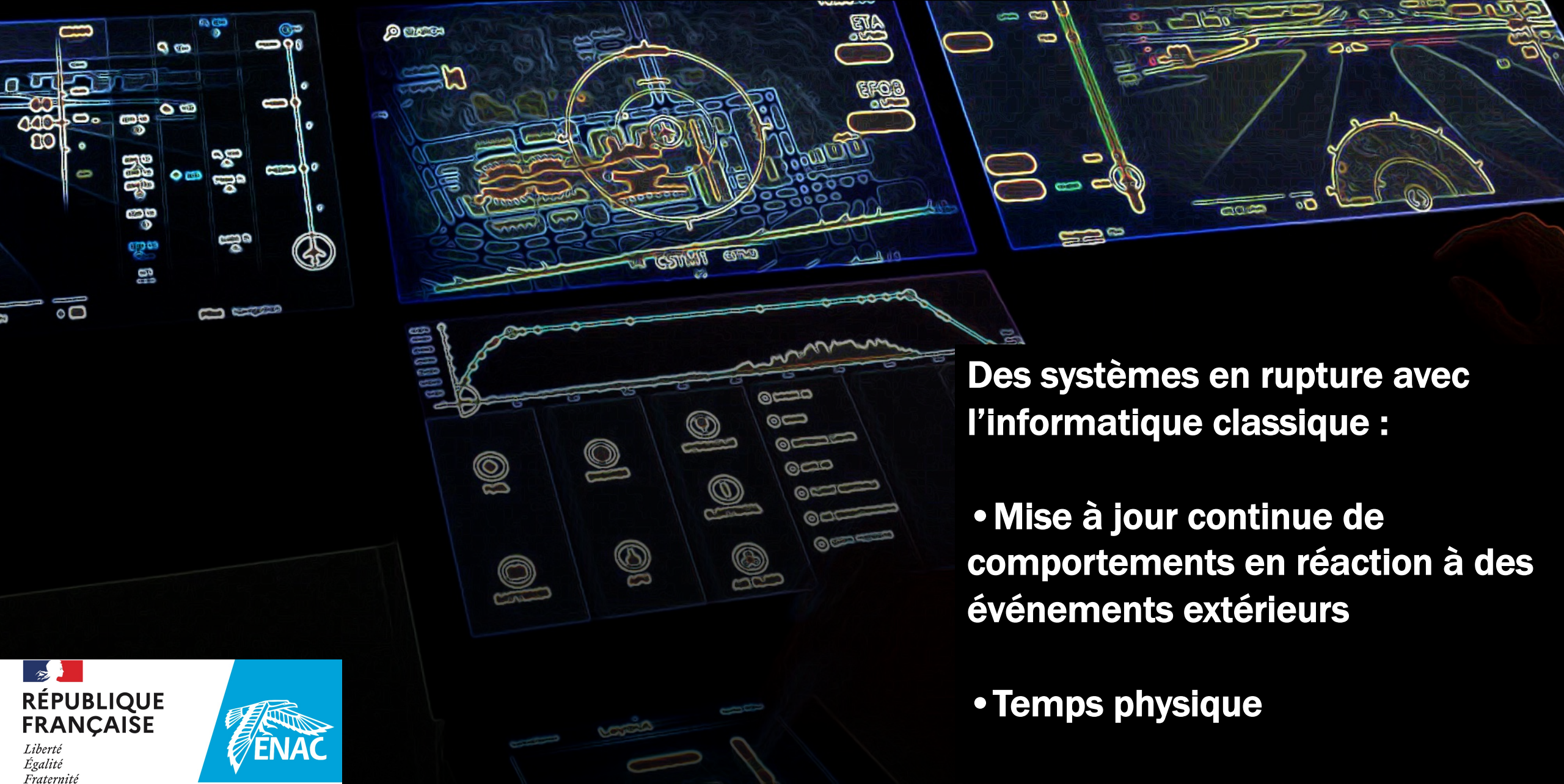


Concepts et outils pour l'informatique de l'interaction



Les systèmes informatiques interactifs



Des systèmes en rupture avec l'informatique classique :

- Mise à jour continue de comportements en réaction à des événements extérieurs
- Temps physique



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*



Objectifs

Théorique :




Réfléchir à un cadre théorique dédié : c'est *quoi* le pendant de la machine de Turing pour l'informatique interactive ?










Pratique :

A partir du langage de programmation développé dans l'équipe, concevoir un outil dédié pour aider les programmeurs de systèmes interactifs

Résultat : un outil de visualisation de code

BDR design test

A11: T1 T2 T3 T4 09:53:41    connecté

<3'	 LION427	FL 230@230		TC:0s Dact: 4.8 Nm	<3'
<2'	 NJE540W	FL 155@230			<2'
<1'					<1'
<3'	 EZY29JC	FL 370@...		TC:31s Dact: 12.6 Nm	<3'
<2'	 AFR87JK	FL 370@...			<2'
<1'					<1'
<3'	 RYR5G	FL 254@150		TC:113s Dact: 24.7 Nm	<3'
<2'	 ADN22B	FL 162@...			<2'
<1'					<1'

Program3.sma

Strip.sma

```

2 use base
3 use gui
4
5 _define_
6 Strip ()
7 {
8   Translation pos (0, 0)
9   x aka pos.tx
10  ui = loadFromXML ("img/full_strip.svg")
11  strip << ui
12  String speed ("init")
13  i_s => s.speed
14  speed => ui.normal.g1165.data\~9.text

```

(a)

Program3.sma

Strip.sma

```

3 use gui
4
5 _define_
6 Strip ()
7 {
8   Translation pos (0, 0)
9   x aka pos.tx
10  ui = loadFromXML ("img/full_strip.svg")
11  strip << ui
12  String speed ("init")
13  planevp => i_s
14  speed => ui.normal.g1165.data\~9.text

```

(b)

Program3.sma

Strip.sma

```

5 _define_
6 Strip ()
7 {
8   Translation pos (0, 0)
9   x aka pos.tx
10  ui = loadFromXML ("img/full_strip.svg")
11  strip << ui
12  String speed ("init")
13  mybus.ivy.in.[2] => planevp
14  mybus.ivy.in.[1] => planevp
15  planevp => i_s
16  i_s => s.speed
17  speed => ui.normal.g1165.data\~9.text

```

(c)

Chrono.sma

Program.sma

```

96 FSM ctrl {
97   State pause_idle {
98     img << ui.pause
99   }
100   State pause_hover {
101     img << ui.pause_hover
102   }
103   State play_idle {
104     img << ui.play
105   }
106   State play_hover {
107     img << ui.play_hover
108   }
109   pause_idle->pause_hover (play_mask.enter)
110   pause_hover->pause_idle (play_mask.leave)
111   pause_hover->play_hover (play_mask.press, chrono.start_chrono)
112   play_idle->play_hover (play_mask.enter)
113   play_hover->play_idle (play_mask.leave)
114   play_hover->pause_hover (play_mask.press, chrono.pause_chrono)

```

(a)

Chrono.sma

Program.sma

```

96 FSM ctrl {
97   State pause_idle {
98     img << ui.pause
99   }
100   State pause_hover {
101     img << ui.pause_hover
102   }
103   State play_idle {
104     img << ui.play
105   }
106   State play_hover {
107     img << ui.play_hover
108   }
109   pause_idle->pause_hover (play_mask.enter)
110   pause_hover->pause_idle (play_mask.leave)
111   pause_hover->play_hover (play_mask.press, chrono.start_chrono)
112   play_idle->play_hover (play_mask.enter)
113   play_hover->play_idle (play_mask.leave)
114   play_hover->pause_hover (play_mask.press, chrono.pause_chrono)

```

(b)

Chrono.sma

Program.sma

```

96 FSM ctrl {
97   State pause_idle {
98     img << ui.pause
99   }
100   State pause_hover {
101     img << ui.pause_hover
102   }
103   State play_idle {
104     img << ui.play
105   }
106   State play_hover {
107     img << ui.play_hover
108   }
109   pause_idle->pause_hover (play_mask.enter)
110   pause_hover->pause_idle (play_mask.leave)
111   pause_hover->play_hover (play_mask.press, chrono.start_chrono)
112   play_idle->play_hover (play_mask.enter)
113   play_hover->play_idle (play_mask.leave)
114   play_hover->pause_hover (play_mask.press, chrono.pause_chrono)

```

(c)



RÉPUBLIQUE
FRANÇAISE

Liberté
Égalité
Fraternité



Bibliographie et production

RÉFÉRENCES

- Feitelson. 2014. On the effect of code regularity on comprehension.
- Goldin & Wegner. 2008. The interactive nature of computing.
- Ko & Myers. 2004. Designing the Whyline: A debugging interface for asking questions about program behavior.
- Lee. 2006. Are Computing Foundations Adequate ?
- Magnaudet et al. 2018. Djnn/smala: A conceptual framework and a language for interaction-oriented programming.
- Salvaneschi. 2016. Debugging for reactive programming.

CONFÉRENCES

- IACAP19'. Engineering Distributed Systems : How Efficient Is A Computational Model ?
- IHM21'. Towards Interactive Completeness: Requirements for an Interactive Abstract Machine. (paper published in the ACM Journal)
- Philosophy of Informatics VI 2022. Modelling Interactive Computing Systems: Do We Have a Good Theory of What Computers Are?
- ICPC22' (submitted). Causette: User-Controlled Rearrangement of Causal Constructs in a Code Editor