

L'Informatique du Temps et des Événements ***Cours Gérard Berry 2013***

Séminaire ***Modélisation Synchrone des Logiciels et Systèmes d'Avionique***

Collège de France, 16 avril 2013

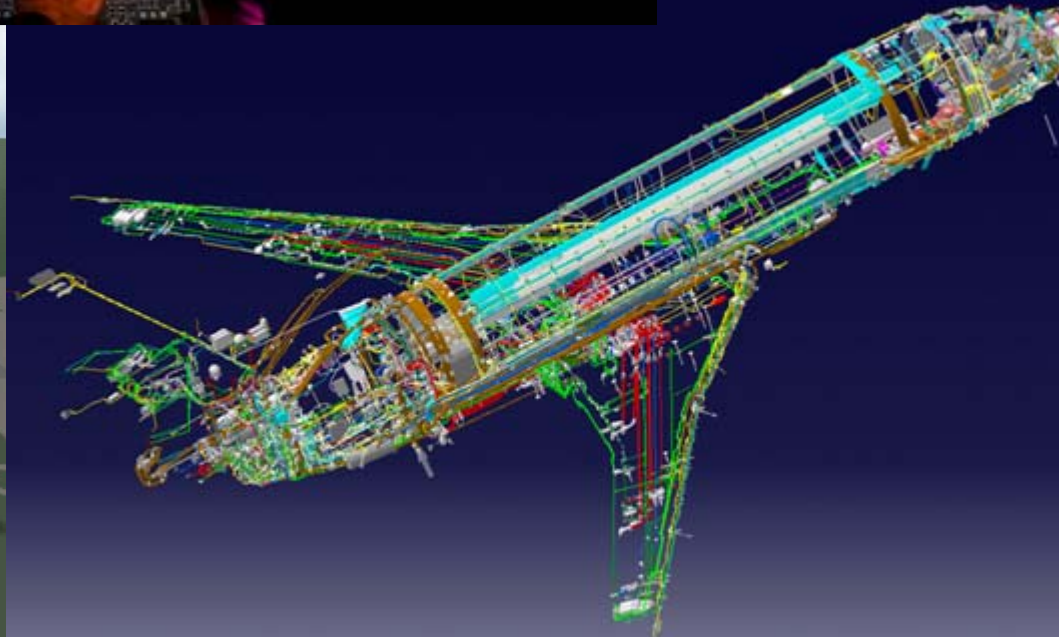
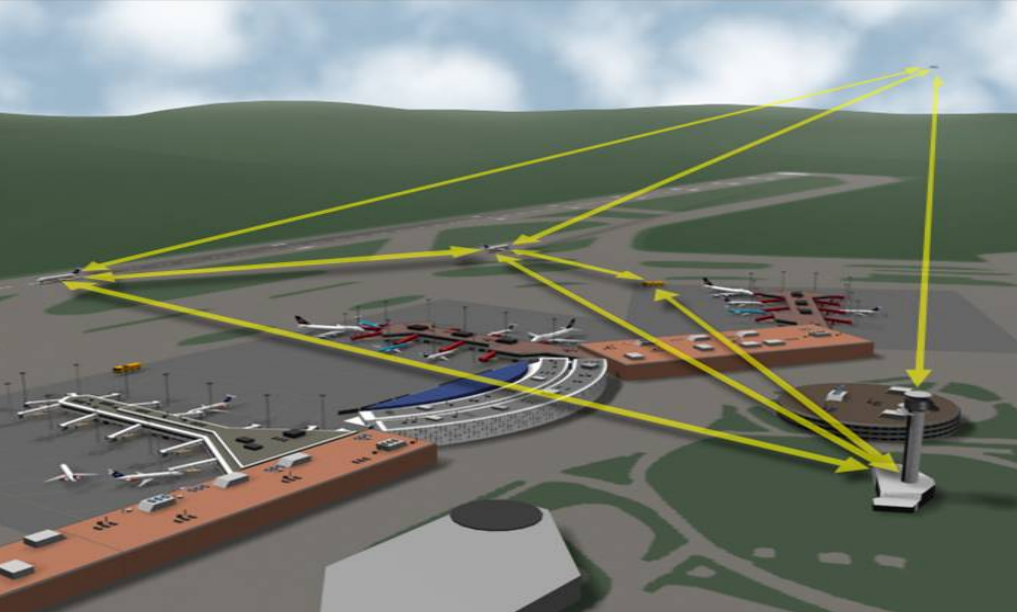
Emmanuel Ledinot
Direction de la Prospective
Dassault Aviation

Temps Pilote

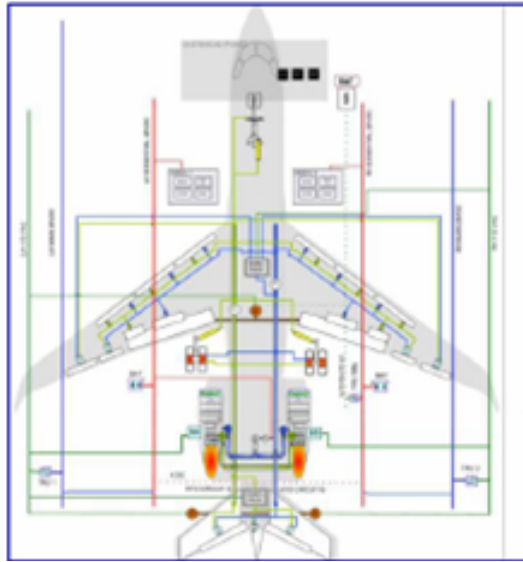
Temps Externe



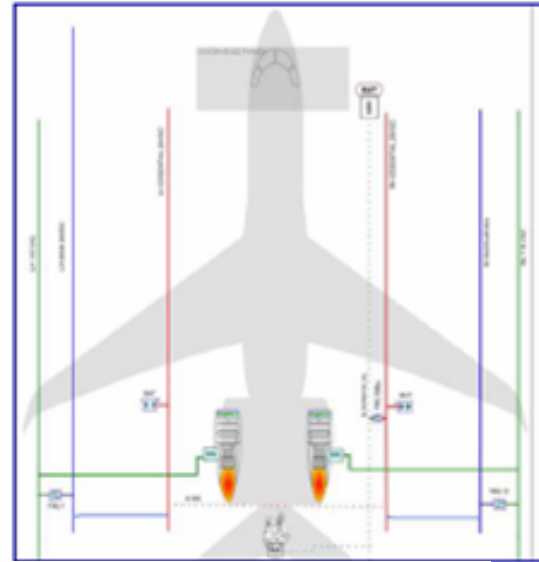
Temps Interne



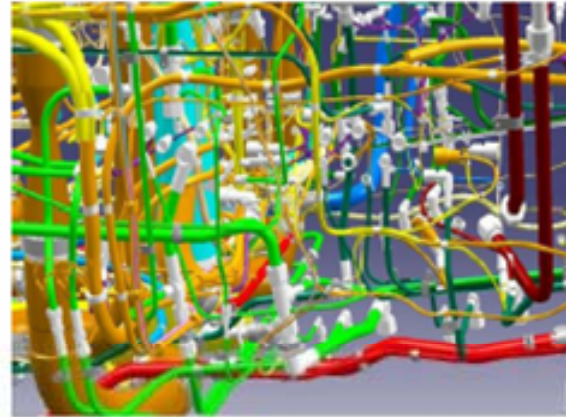
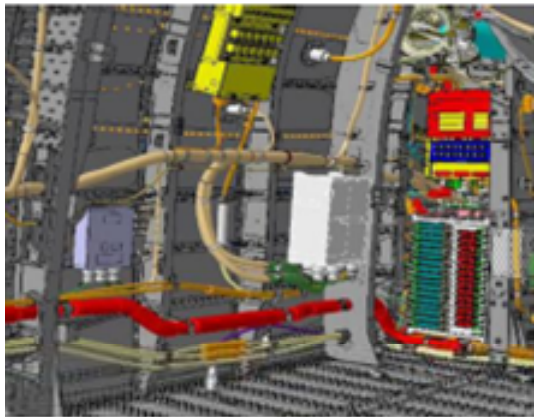
Les Systèmes Avion



Hydraulique



Electricité

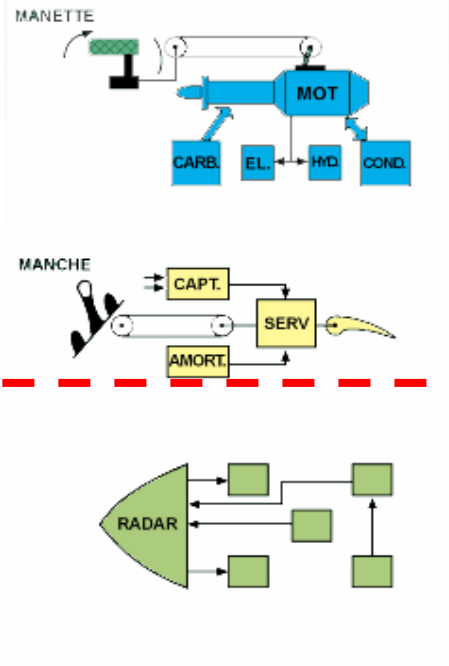


Maquette Numérique

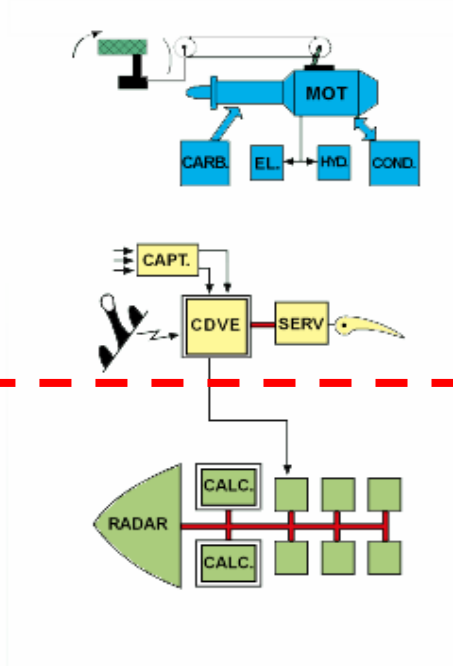
- Propulsion
- Auxiliary Power Unit (APU)
- Prélèvement d'air
- Conditionnement d'Air
- Oxygène
- Génération / distribution électrique
- Hydraulique
- Commandes de vol
- Trains / trappes
- Carburant
- Protection incendie
- Aménagement / support vie
- Anti-givrage / protection pluie
- Eau
- Avionique
- Maintenance embarquée

Evolution des Architectures Numériques

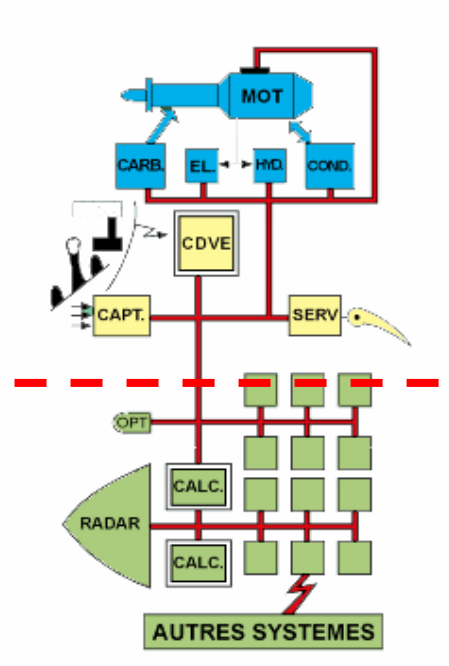
1970



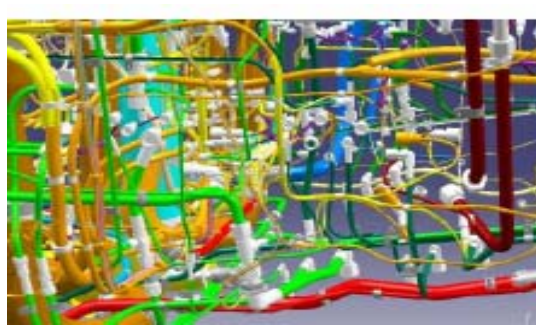
1980



1990



2000



Conduite par l'équipage



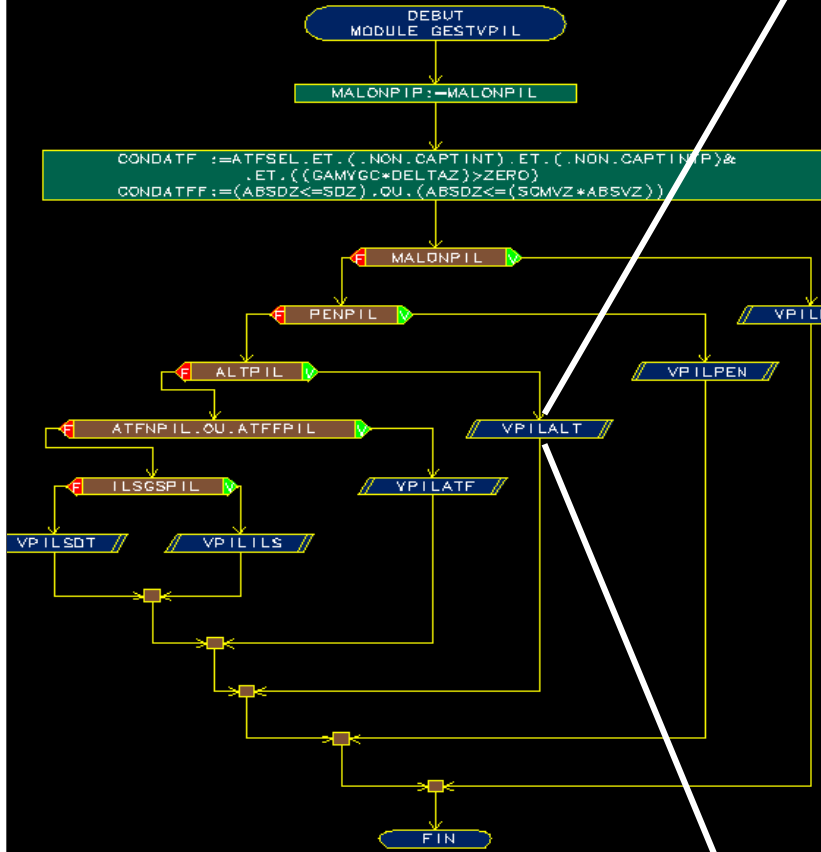
Multiplexage de ressources, fusion de capteurs
=> logiques d'activation complexes

Spécifier, Programmer, Vérifier des logiques d'activation complexes

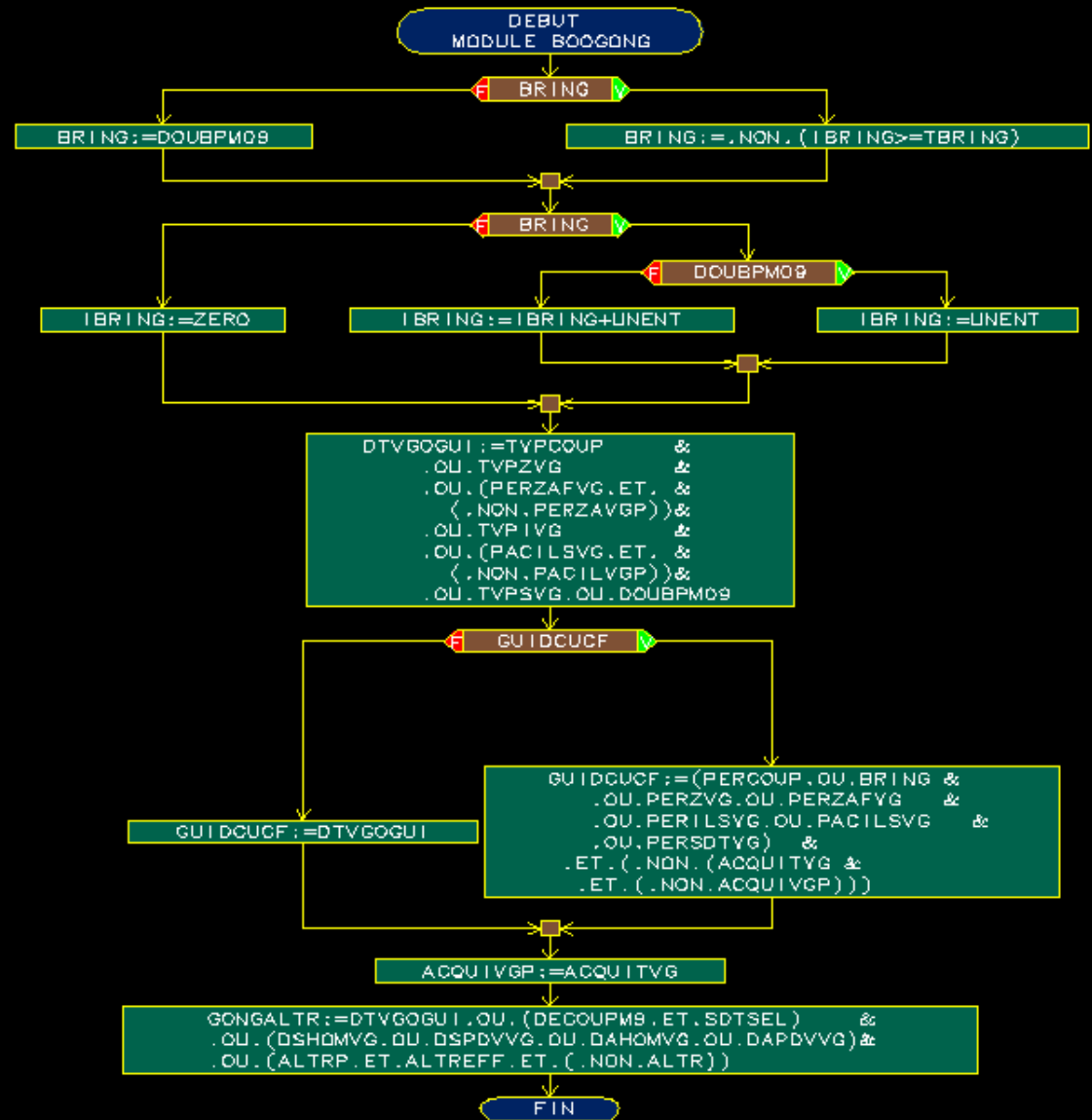
- **Avant: conception séquentielle**
- **Abstraction du séquençement**
 - Orthogonaliser / Hiérarchiser
 - Causalité
 - Confluence
- **Aperçu d'un vrai logiciel**
- **Après: les apports de SCADE 6**



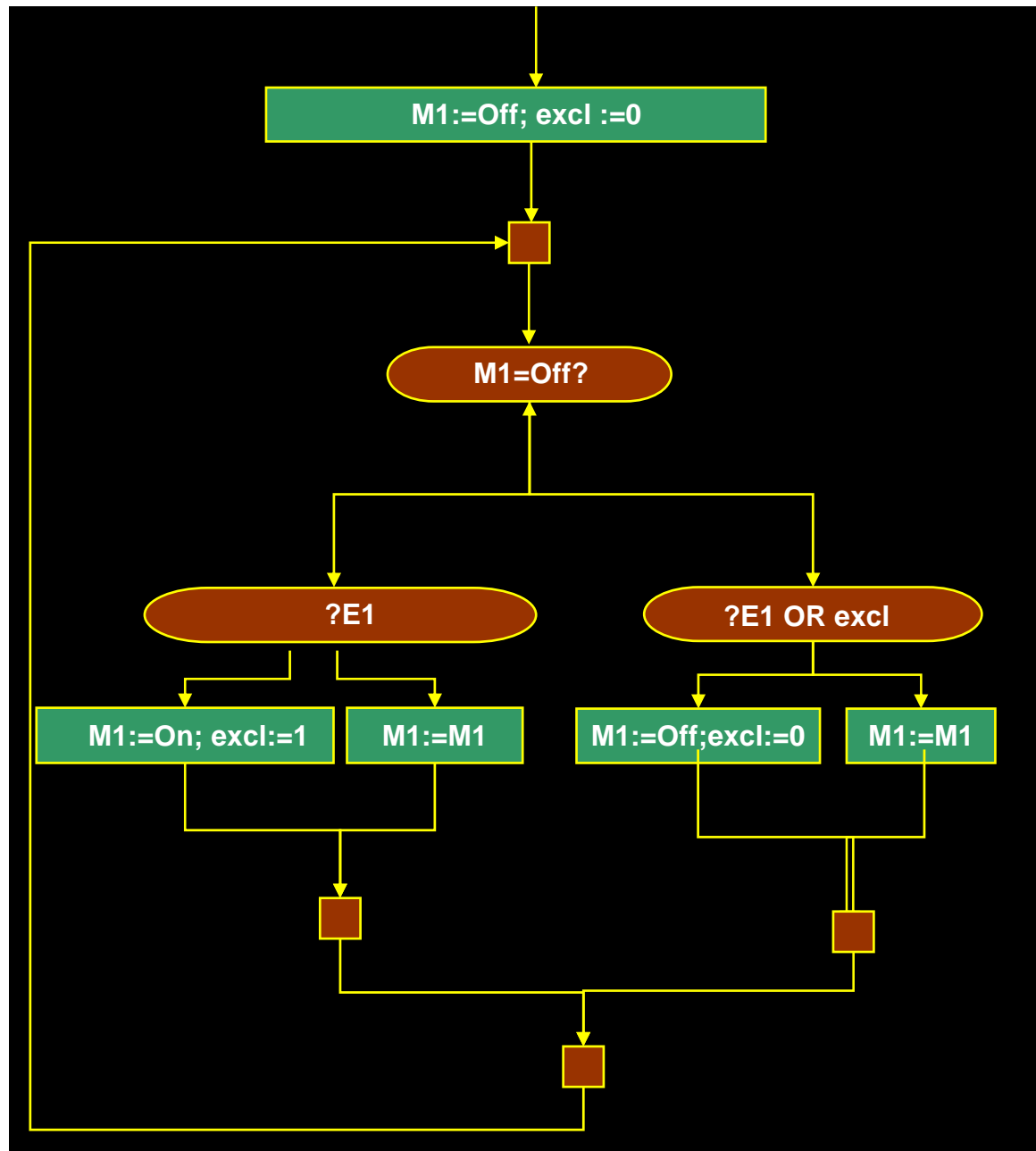
0 1 1 0 1 0 0 1 0 1 1 0 0 0 0 0 0



0 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1

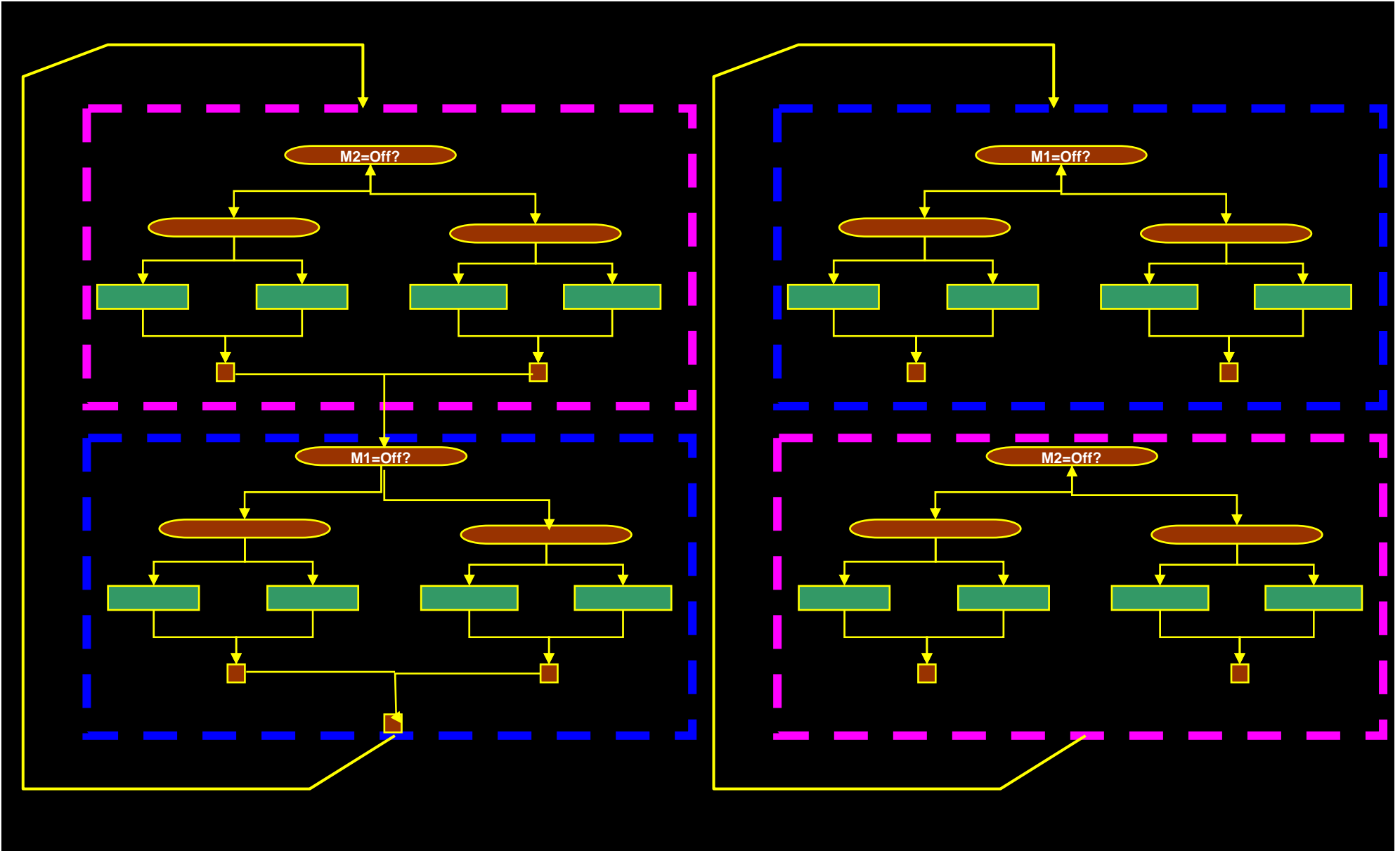


Penser séquentiel ...

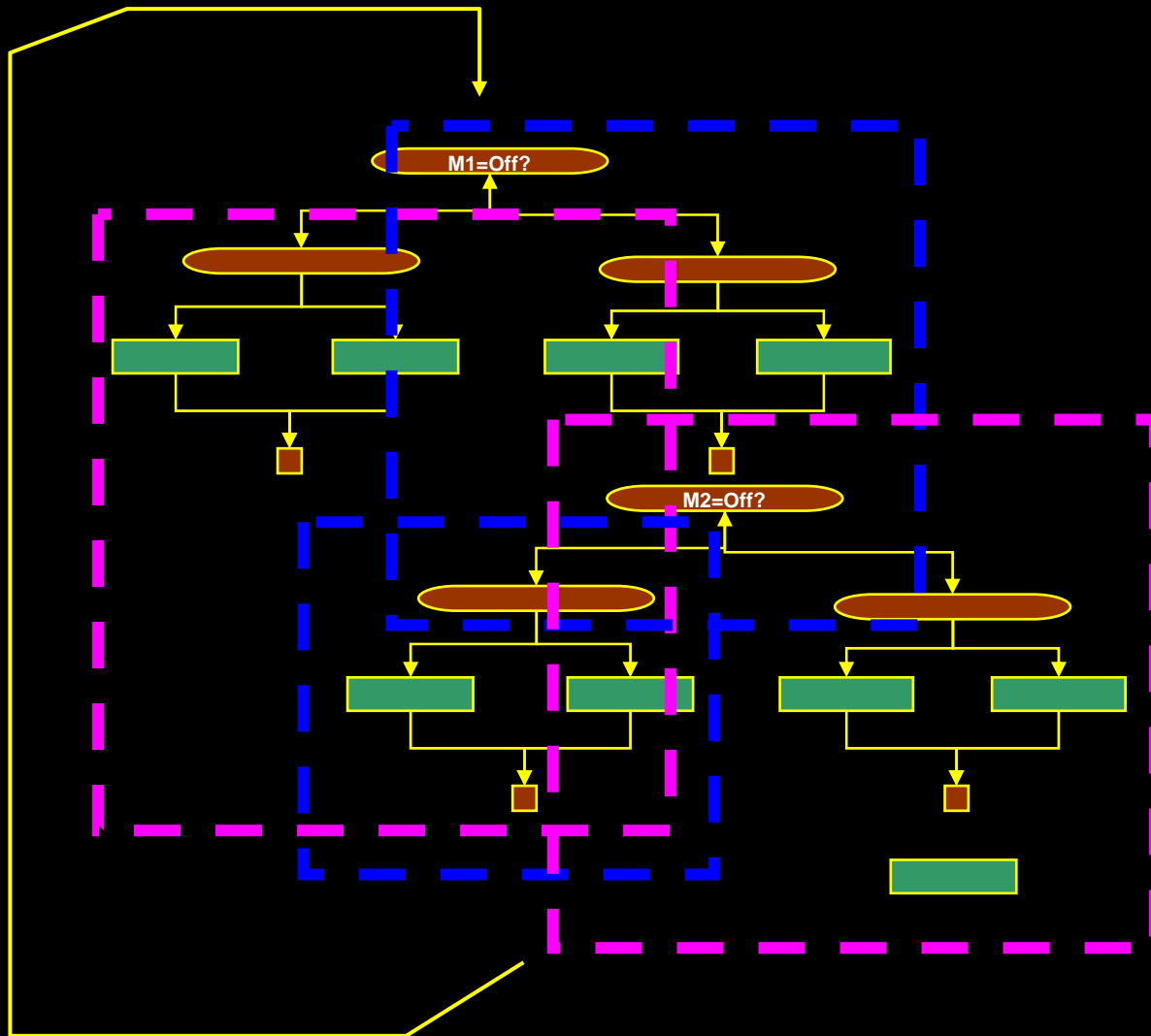


... penser global

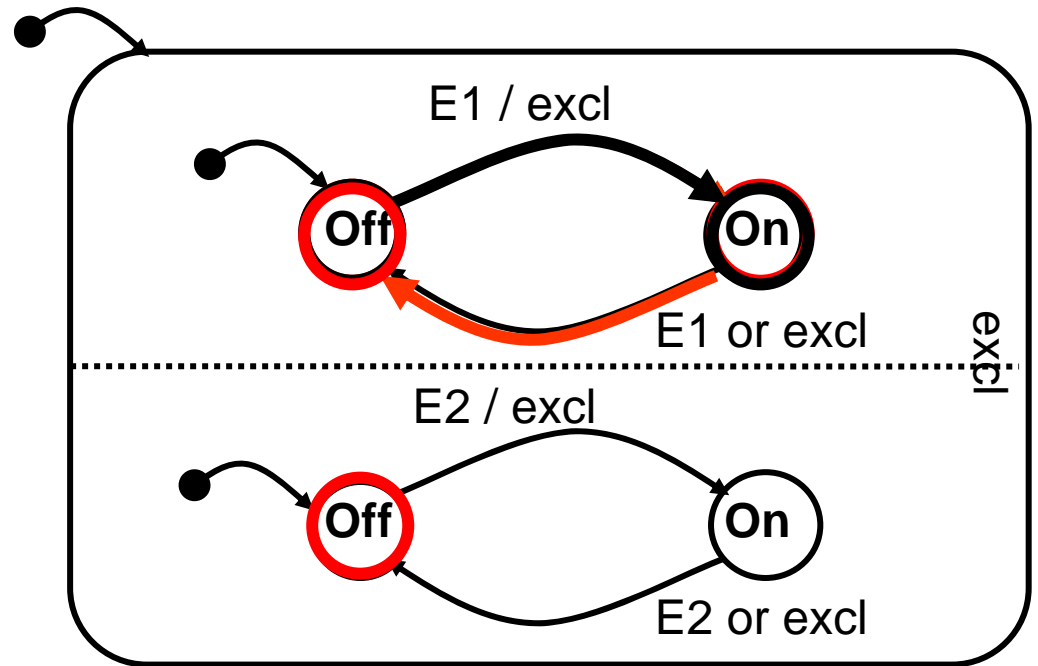
Pas de séquençement modulaire



Séquencement à la main ...



En Esterel



signal excl in

```
loop await E1; emit excl; await E1 or excl) end
```

```
//
```

```
loop await E2; emit excl; await (E2 or excl) end
```

```
end
```

de l'ordre aussi, mais partiel

Causalité



? Absent !

```
input TOP_HORLOGE INDICATEUR_ON;  
relation TOP_HORLOGE # INDICATEUR_ON;
```

Inactif !

```
signal FIN REARMEMENT in  
  abort  
    await TOP_HORLOGE do emit FIN end  
  when REARMEMENT  
  ||  
  do  
    ?  
    every INDICATEUR_ON do emit REARMEMENT end  
  upto FIN  
end
```

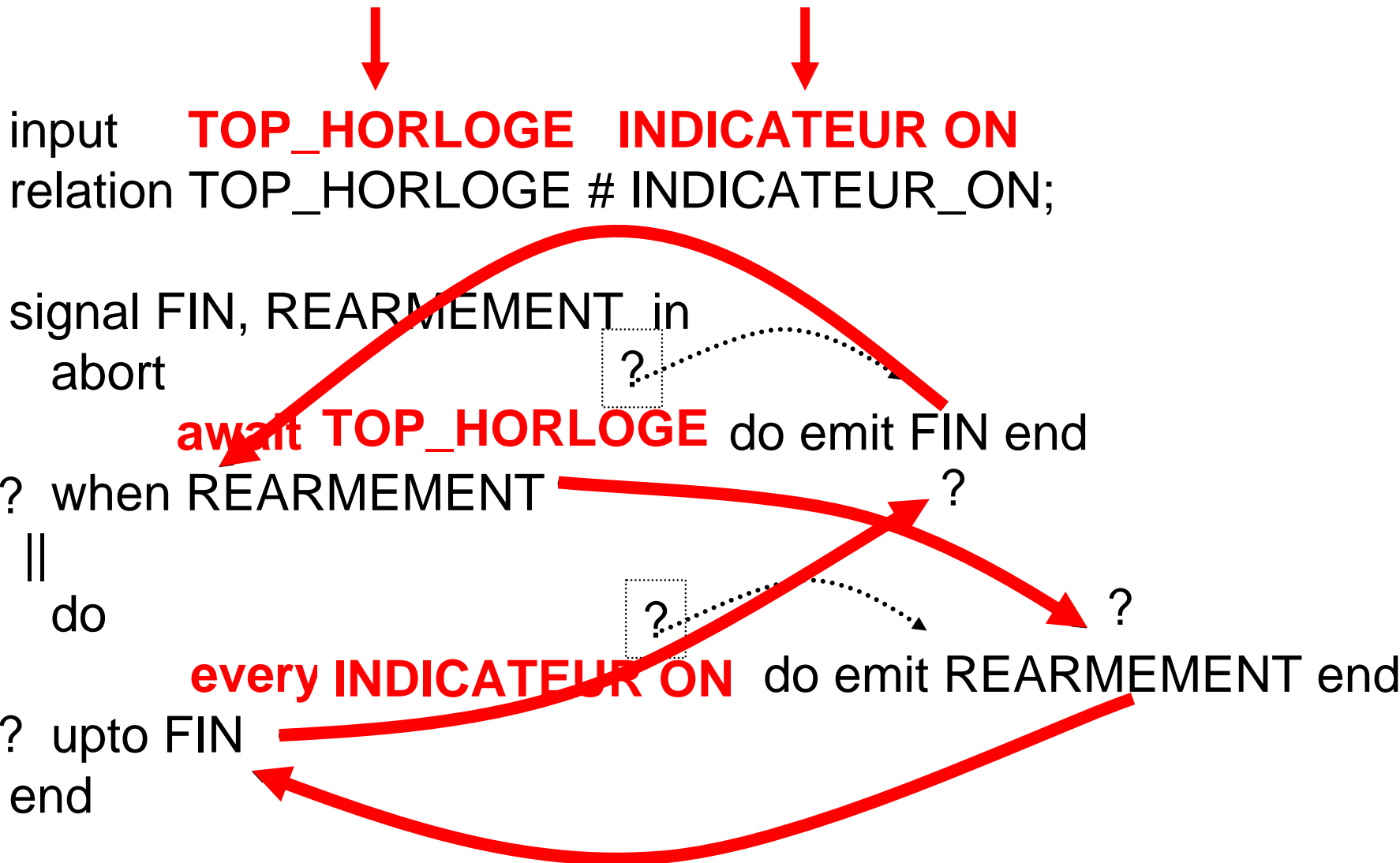
OK!



Absent !

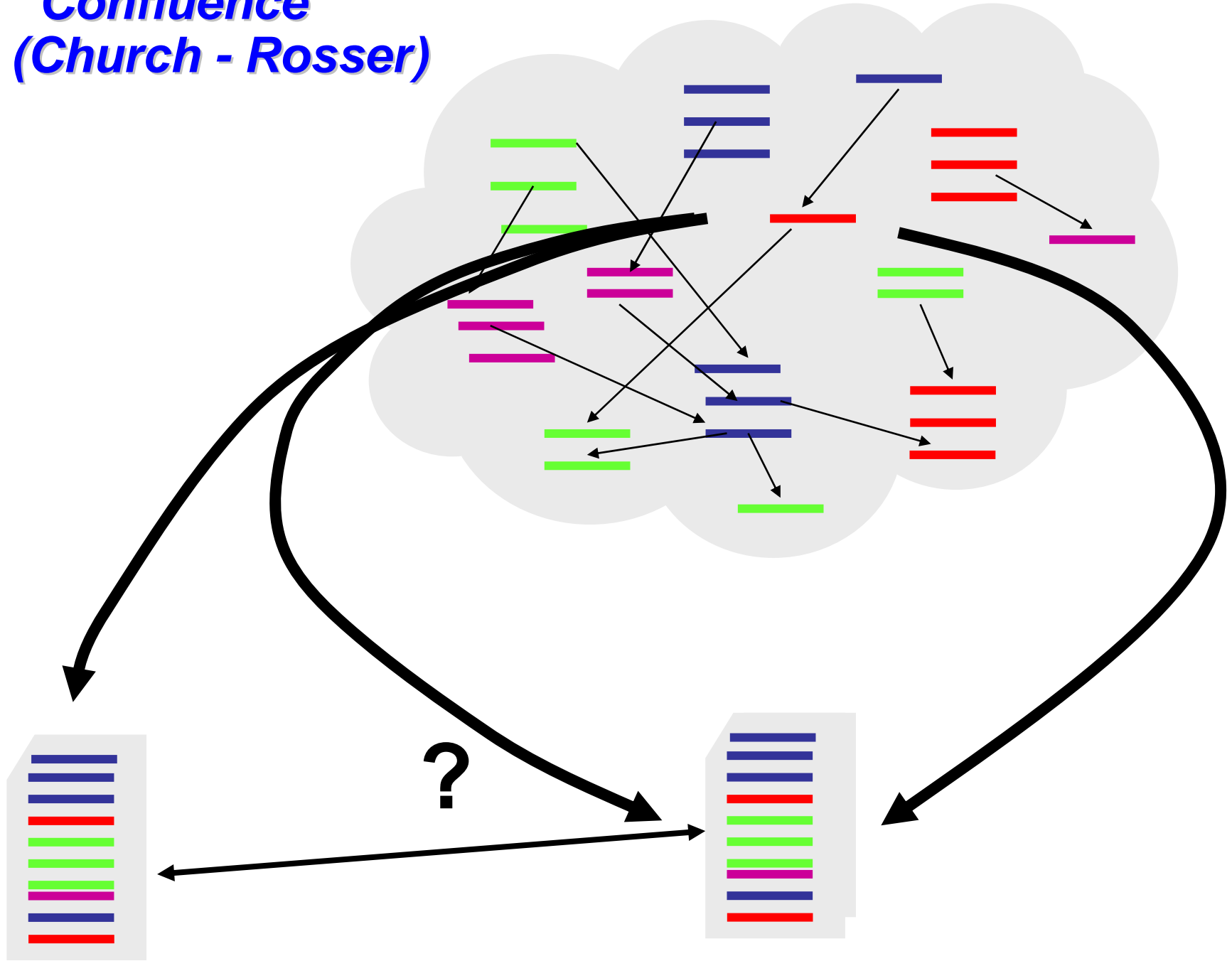


Cycle de causalité



*Une décision à un "instant" d'un instant
ne doit pas être remise en cause dans le reste de l'instant*

Confluence (Church - Rosser)



Enjeu: des comportements qui ne dépendent pas ...

- **de la version du compilateur ...**
- **de positions graphiques relatives sur l'écran ...**
- **de l'ordre lexicographique des identificateurs ...**
- **etc.**

Dessine moi un logiciel



Deux planches retirées

Volume du Logiciel



F1 1980

~ 20 m²

? m³

1 m



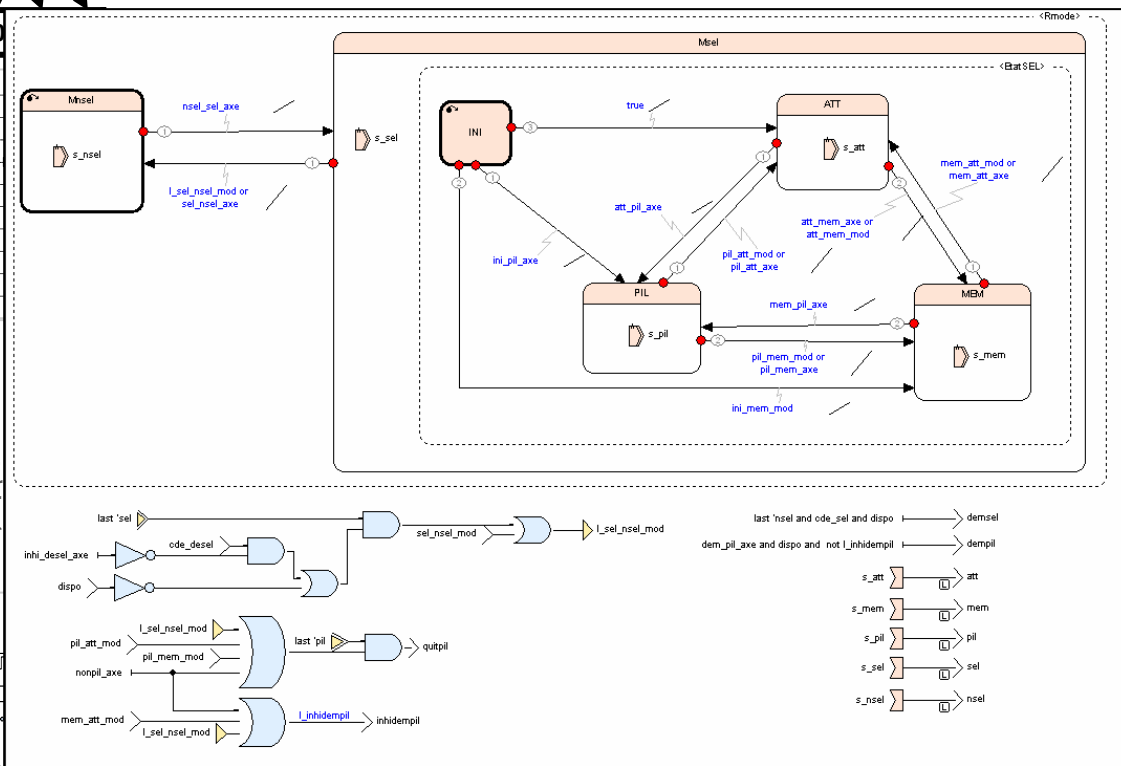
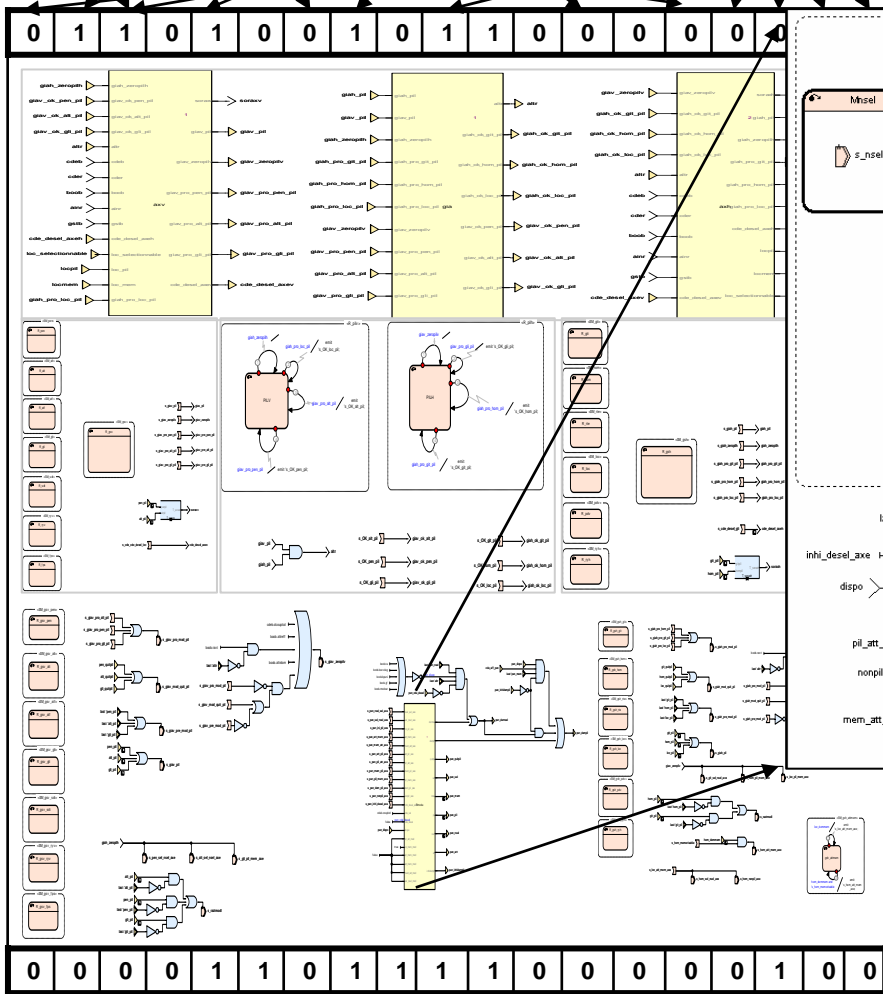
M2000 1987

5 m



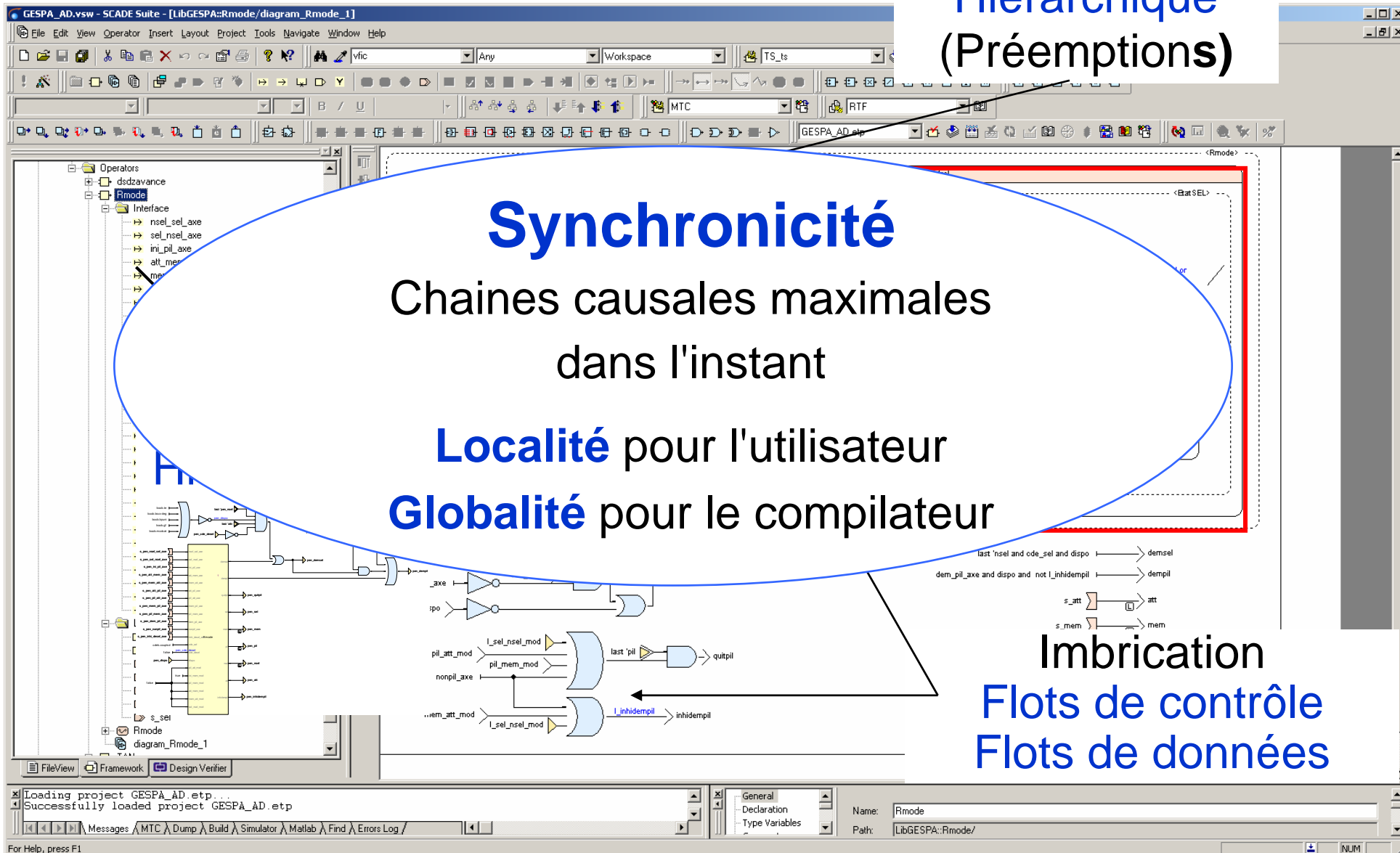
Rafale 2000

100 m



Dessine moi un logiciel: Expressivité

Etats
Hiérarchique
(Préemptions)



1989-2013 D'Estereel V3 à SCADE 6



2003
1 m



2007
1 m



2012
20 m

Conclusion

- **Systemes intégrés: quand on ajoute, on multiplie** (les états d'activation)
- **Factoriser pour maîtriser: orthogonalité + interactions** (abstraction du séquençement)
- **Causalité:** conditions d'existence d'un séquençement *statique*
- **Confluence:** invariance du comportement de l'ordre partiel à *un* ordre total
- **SCADE 6:** aboutissement de 20 ans de recherche et 10 ans d'industrialisation

Un outil essentiel pour la maîtrise des logiciels critiques

Merci

En particulier à: Joël Boudin
Philippe Bourdais
Christine La Porte
Henri Michel
Bruno Ramirez