

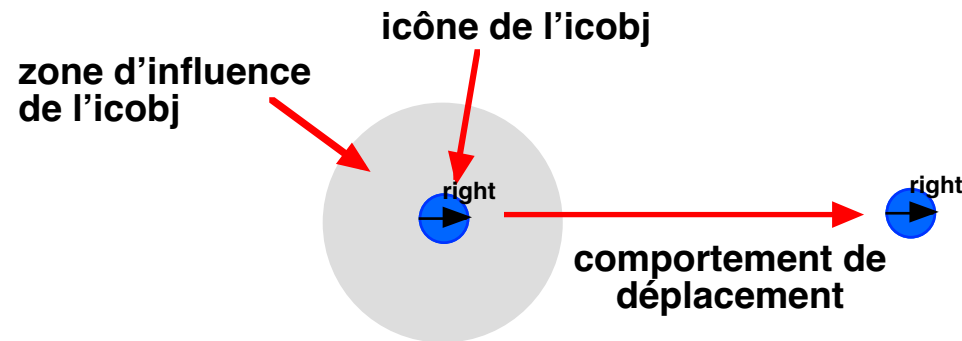
# Programmation réactive graphique

- Programmation par icobjs
- Implémentation
- Démos

# La notion d'icobj

Composants d'un **icobj** (iconic object) :

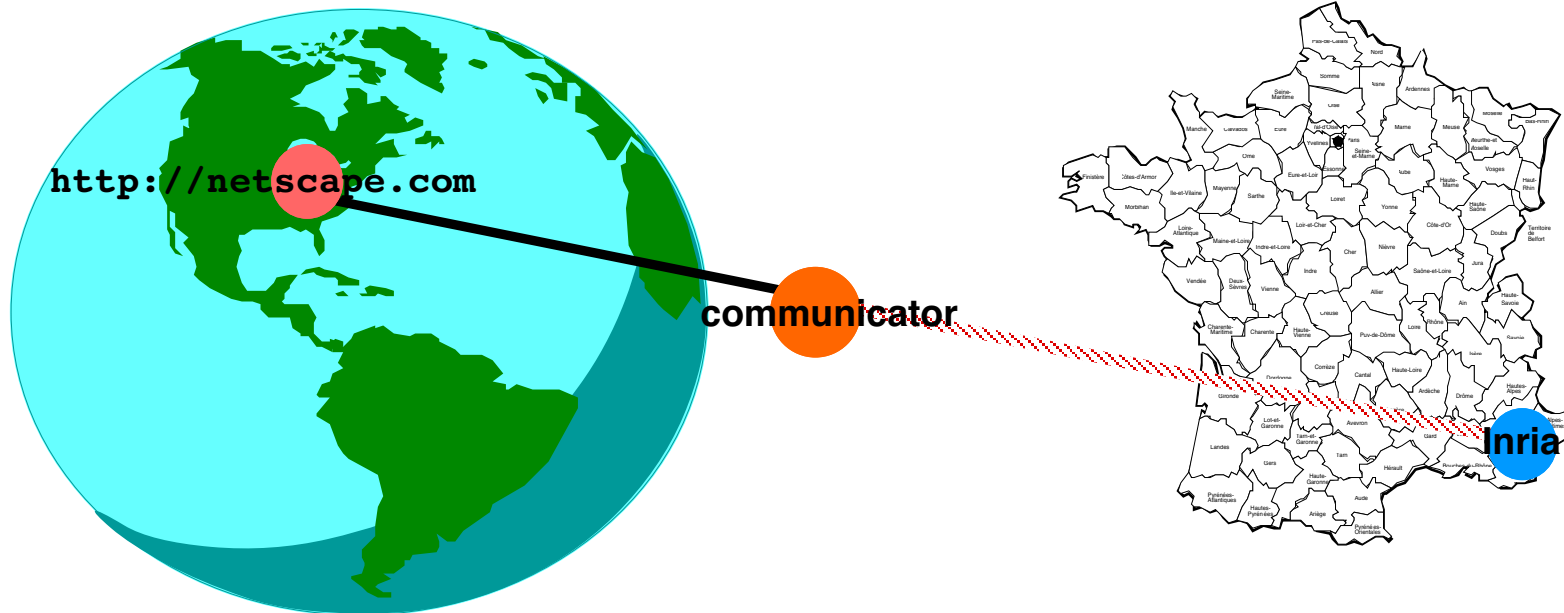
- Un **aspect graphique** : sa représentation à l'écran
- Sa **zone d'influence** : partie graphique dans laquelle l'icobj peut influencer directement d'autres icobjs
- Son **comportement** : s'exécute lorsque l'icobj est activé



# Utilisation d'icobjs

Exemples :

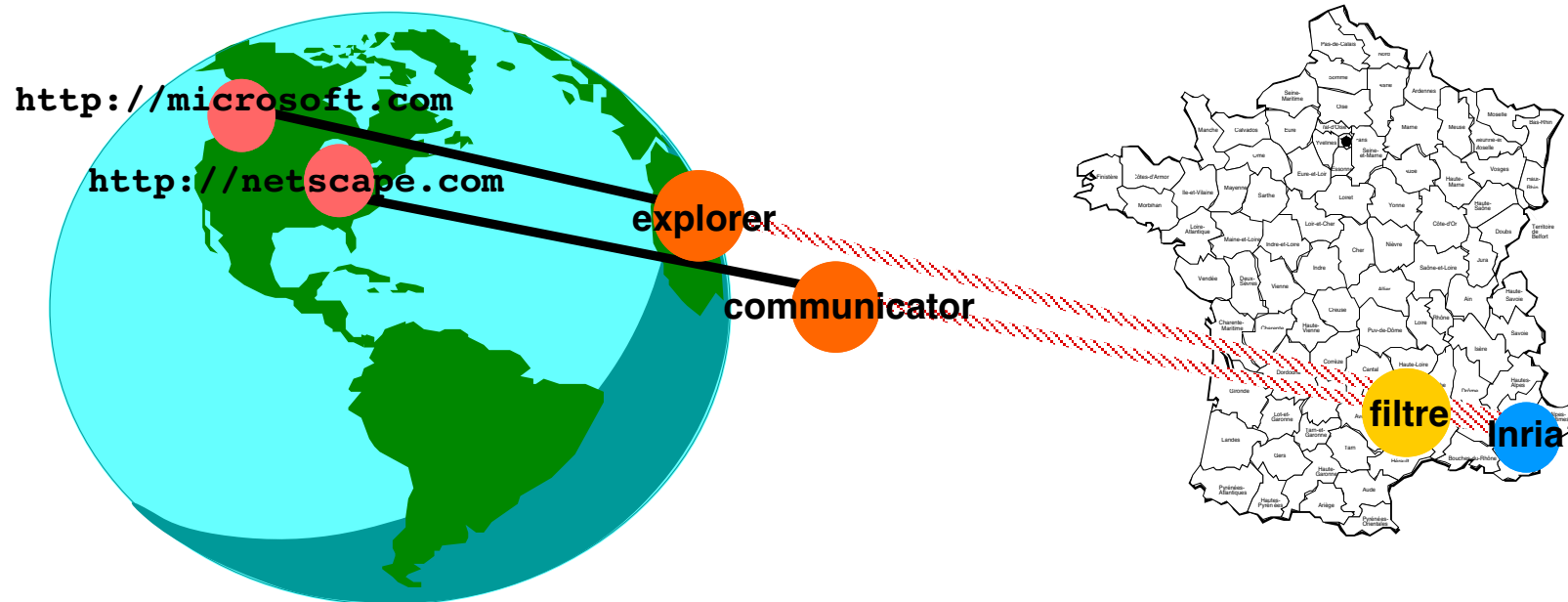
- Download d'un logiciel :



- Visualisation de la migration d'agents

# Interactions entre icobjs

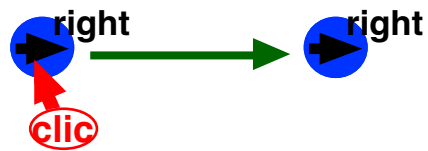
- Interaction par les zones d'influence
- Communication par événements diffusés :



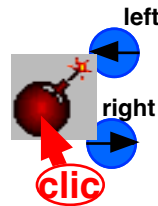
**filtre** détecte le premier icobj qui l'atteint et envoie un événement de destruction aux autres

# Un système d'icobjs

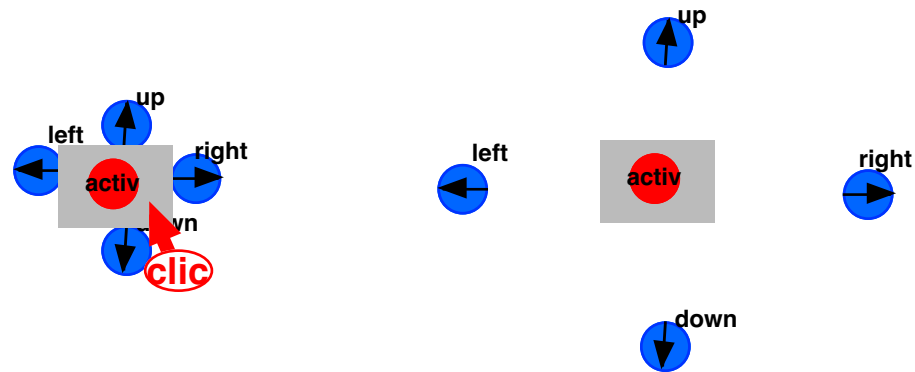
Right, left, up, down :



Delete :



Activ :



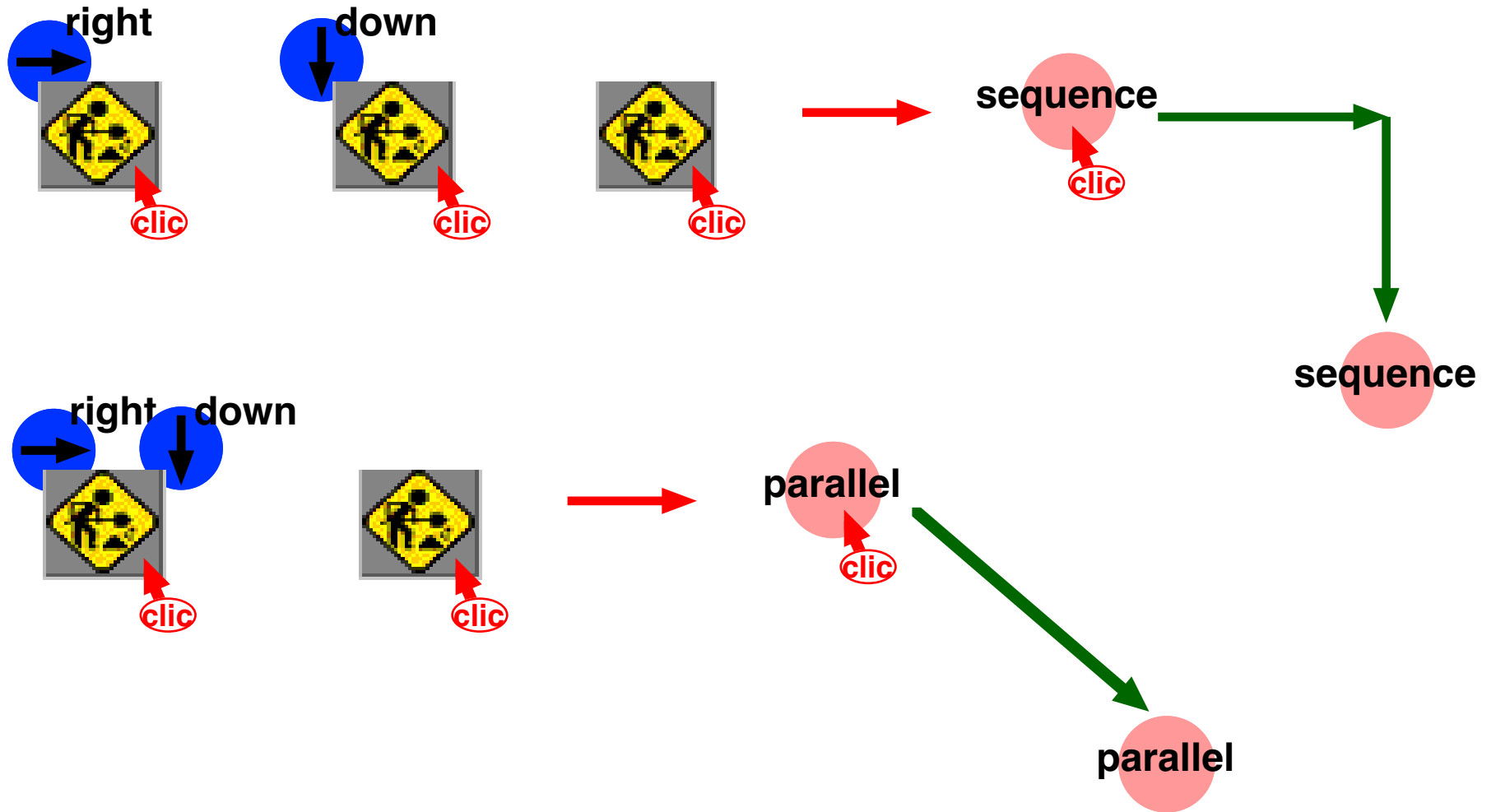
# Le constructeur

Icobj builder utilisé pour :

- **Nommer** un icobj
- Créer un **clone** (copie conforme d'un icobj existant)
- Enchaîner les comportements d'icobjs (les mettre en **séquence**)
- Combiner les comportements d'icobjs (les mettre en **parallèle**)
- Créer des icobjs ayant des comportements cycliques (**boucles**)



# Exemples de constructions

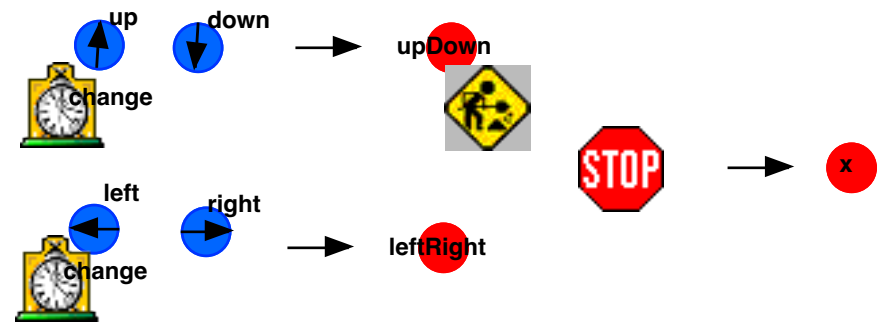


# Les événements

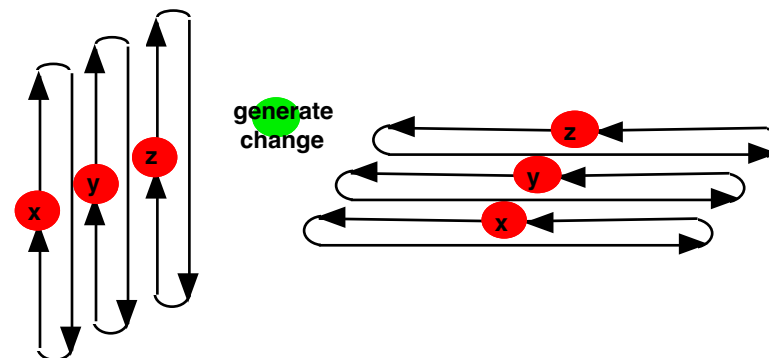
Icobjs generate, await et until :



Haut/bas devient droite/ gauche et inversement, à chaque fois que change est généré :



Diffusion des événements :



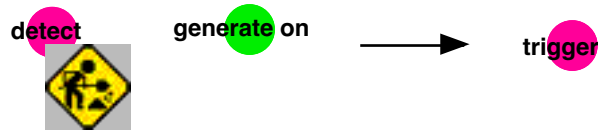


# L'icobj detect

Séquence de construction d'un icobj mine :



Séquence de construction de l'icobj de déclenchement :

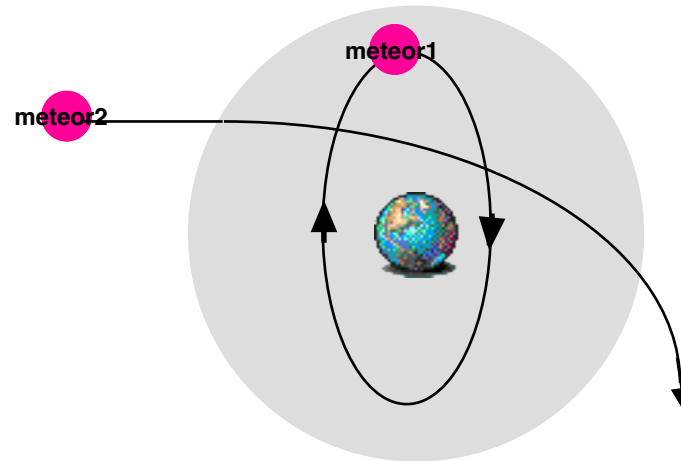


Destruction de right :



# Attracteurs

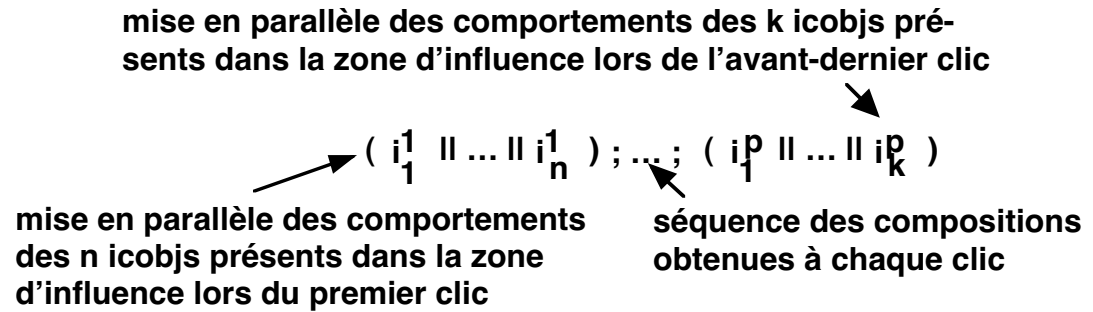
**Objets planet et meteor :**



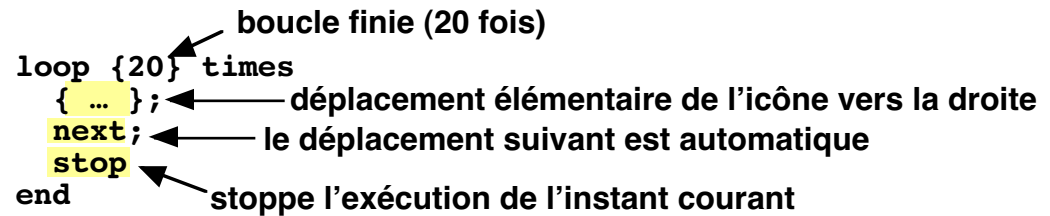
**Planet + meteor ?**

# Sémantique des icobjs: scripts réactifs

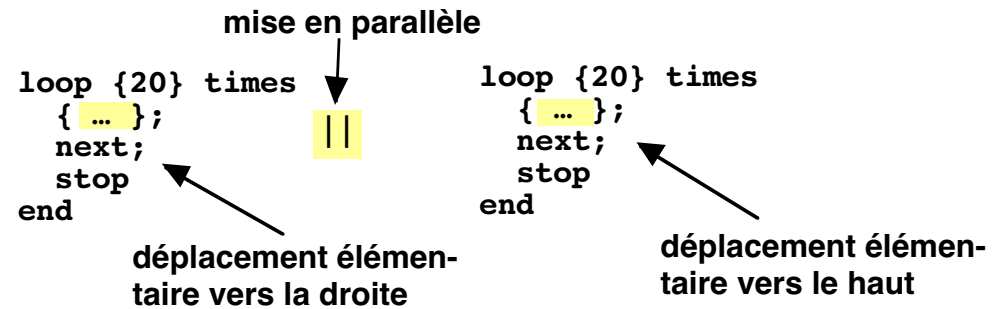
## Builder :



## lcobj right :

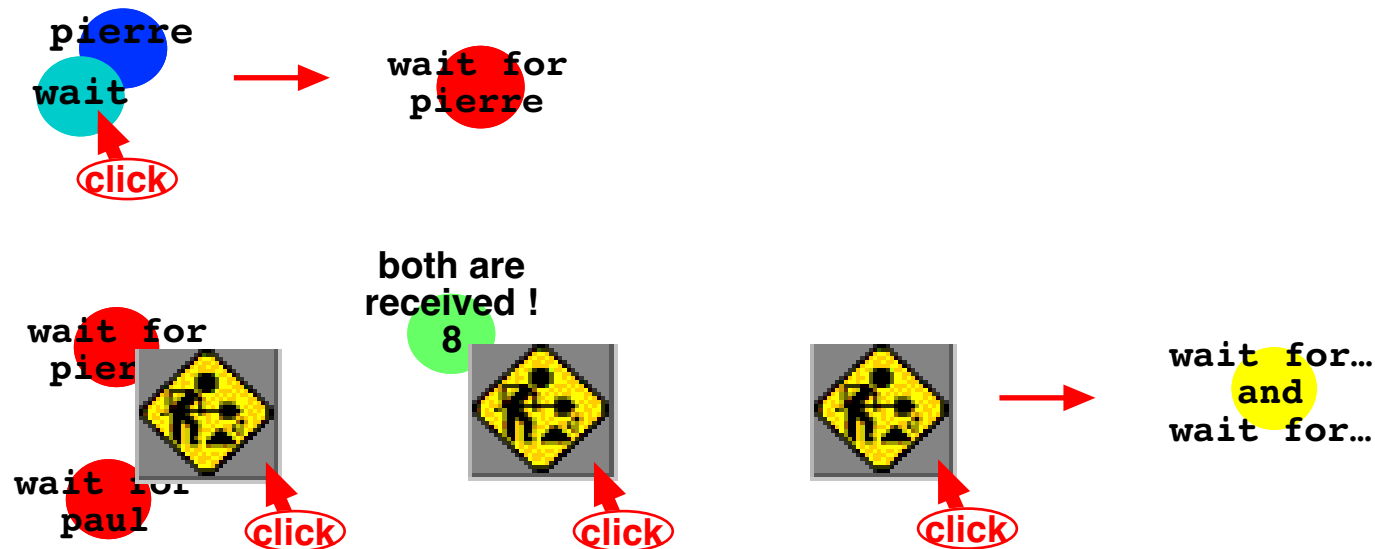


## Combinaison de right et up :



# Utilisation possible pour les télécoms

Construction de services par l'utilisateur final :



```
(await pierre_com || await paul_com); run speak(message_8)
```

Activation du nouveau service :

`wait for...`  
`and`  
`wait for...`  
`click` (circled in red)

# Conclusion

---

Icobjs avec les caractéristiques suivantes:

- pas de syntaxe : **tout graphique**
- interprétation : rien à compiler
- un seul icobj pour séquence, **parallèle**, boucle
- diffusion d'événements
- **sémantique claire**
- opérateur de préemption
- réutilisabilité

# Conclusion (suite)

---

## Domaines d'application

- IHM graphiques
- Programmation du Web
- **Construction de services par l'utilisateur final** dans un contexte télécom
- Enseignement (Physique)

## Trois implémentations

Reactive-C + Tcl/Tk :

<http://www.inria.fr/meije/rc/Softwares>

SugarCubes :

<http://www.inria.fr/meije/rc/WebIcobj>

**Junior** :

<http://www.inria.fr/mimoso/rp/Icobjjs>