

# Le futur des boîtes à outils

Atelier IHM - Anglet

Michel Beaudouin-Lafon  
mbl@lri.fr - insitu.lri.fr

## Pourquoi des boîtes à outils ?

Instrument de la recherche en IHM  
Besoin d'outils pour expérimenter

Objet de recherche  
Réification des concepts d'architecture logicielle

Vecteur de transfert  
Faciliter l'usage de nouvelles techniques d'IHM  
dans les produits commerciaux

## La France en pointe

### *France*

Xtv  
Whizz  
Ubit  
ICON  
MaggLite  
Infovis tk  
HsmTk  
IntuiKit  
Metisse  
SwingStates  
...

### *Reste du monde*

Xt  
Garnet  
Amulet  
SubArctic  
GroupKit  
Swing  
Jazz  
Qt  
Prefuse  
d.tools  
...

## Quels types de boîtes à outils ?

Boîtes à outils d'interface graphique

Les plus répandues

Toujours pas matures

Boîtes à outils pour la visualisation

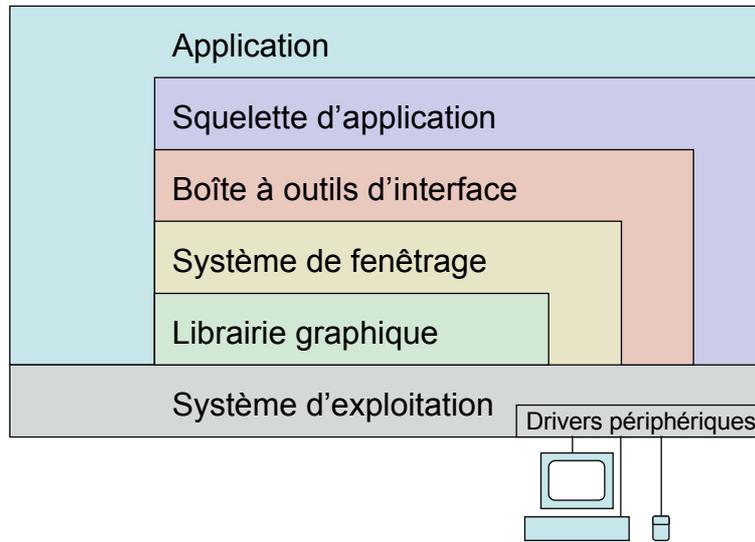
Boîtes à outils pour le collecticiel

Boîtes à outils pour le Web (AJAX...)

...

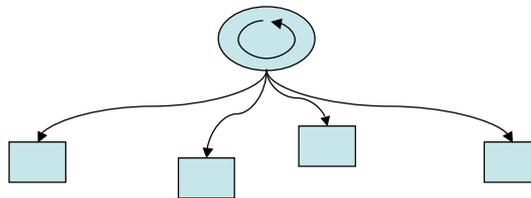
Intergiciels de communication

## Couches logicielles d'une appli. graphique



## Programmation dirigée par les événements

```
tantque non fini faire  
  attendre jusqu'à file non vide // attente passive  
  ev := tête de file // extraire événement  
  cible := chercherCible(ev)  
  si cible ≠ NIL alors cible.traiter(ev)  
fin tantque
```



Style très différent de la programmation algorithmique

## Le modèle prédominant du widget

### Abstraction : le *widget*

Objet interactif, Composant logiciel

Bouton, menu, barre de défilement, boîte de dialogue, ...



### Un widget = trois facettes

Présentation - Comportement - Interface d'application

### Interface = arbre de widgets

Nœuds : conteneurs (barre de menus, boîte de dialogue, ...)

Feuilles : widgets simples (boutons, barres de défilement, ...)

## Les trois facettes d'un widget

### Présentation

Apparence graphique

Paramétrable (« ressources »)

### Comportement

Réaction aux actions de l'utilisateur

Peu ou pas paramétrable

### Interface d'application

Notification des changements d'état

## Placement des widgets

### Règles générales

- Imbrication géométrique d'un widget fils dans son parent
- Contrôle par le parent du placement de ses fils

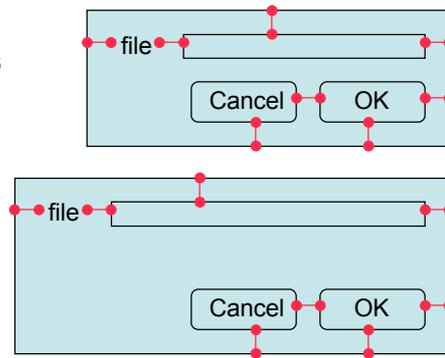
### Algorithme de placement

- Taille naturelle de chaque fils
- Taille et position finales imposées par le parent

Contraintes :

- Grille, formulaire, etc.

Placement dynamique



## Facettes d'un widget

### Présentation

- Apparence graphique
- Paramétrable (« ressources »)

### Comportement

- Réaction aux actions de l'utilisateur
- Peu ou pas paramétrable

### Interface d'application

- Notification des changements d'état

## Interface d'application : fonctions de rappel

1. Enregistrement lors de la création du widget



2. Appel lors de l'activation du widget



Problème : « spaghetti » des call-backs

Partage d'état entre plusieurs call-backs par variables globales

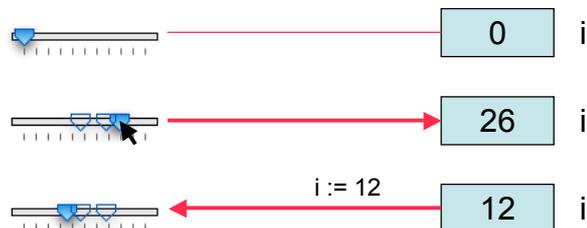


global string filename;  
`DoSetFile () {filename = ...}`

`DoSave () { SaveTo(filename) }`

## Interface d'application : valeurs actives

Lien bi-directionnel entre une variable d'état du widget et une variable de l'application



Problèmes

- Limité aux types simples
- Lien de retour peut être coûteux

Avantages

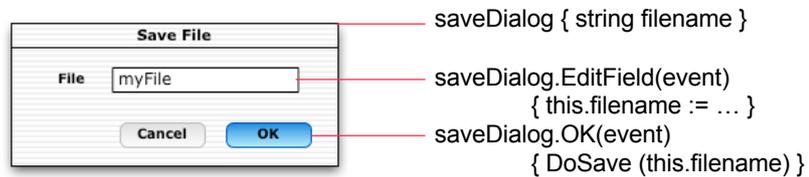
- Vues multiples

## Interface d'application : envoi de message

Association d'un objet à un widget  
et de méthodes de l'objet aux changements d'état du widget

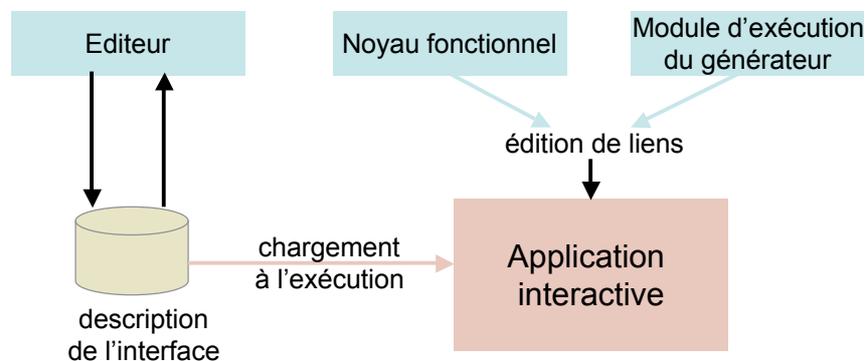


Meilleure encapsulation



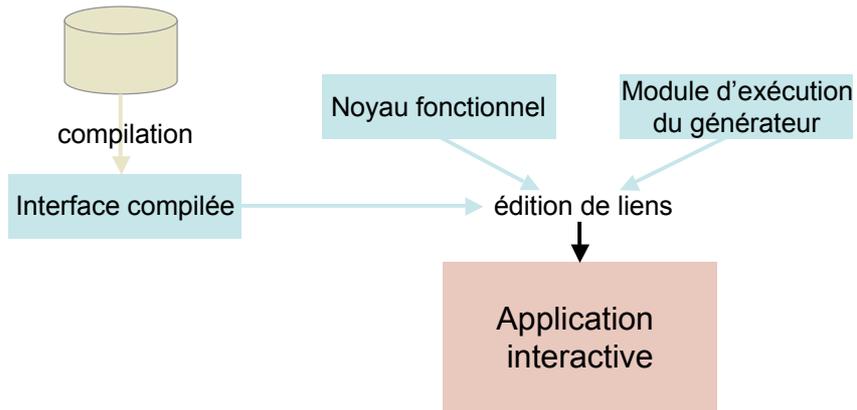
## Générateurs d'interface

Description (textuelle ou graphique) d'une partie de l'interface  
Génération de la partie exécutable



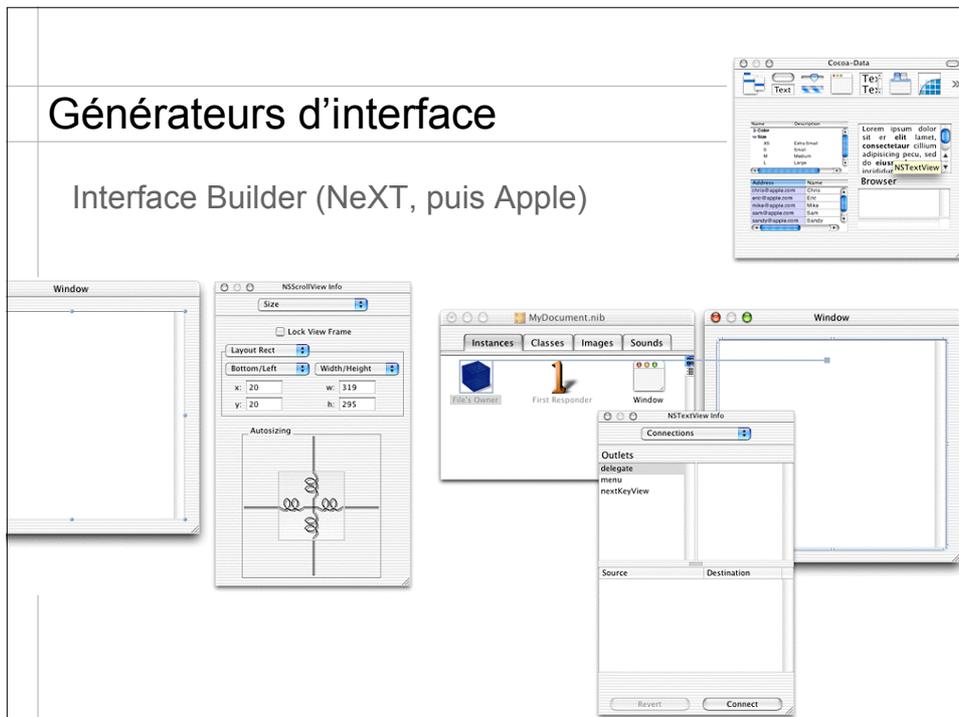
## Générateurs d'interface

Génération de l'application finale



## Générateurs d'interface

Interface Builder (NeXT, puis Apple)



## Limites des boîtes à widgets

### Avantages de ces outils

- Diminuer les coûts de développement et de maintenance
- Faciliter le respect des règles ergonomiques

### Inconvénients de ces outils

- Styles d'interaction stéréotypés (pas de drag-and-drop !)
- Peu extensibles
- Coûteux de programmer des interactions "non standards"

### Problèmes de recherche

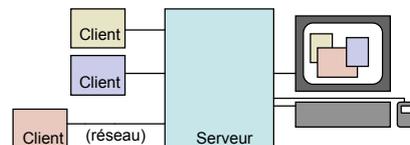
- Sortir du modèle des widgets
- Définir des langages / environnements plus adaptés

## Quels nouveaux défis ?

Personnalisation / adaptation "profonde" des interfaces  
modèle conceptuel externalisable

### Interfaces réparties

- client-serveur
- Web
- collecticiel



### Ubicomp / Pervasive / Ambient computing

- Interaction multi-surfaces
- Interfaces tangibles

## Nouveaux concepts, nouvelles idées ?

Graphes de scènes / Graphes d'interaction  
Jazz, MaggLite, IntuiKit, ...

“Séparation of concerns”  
skin vs. interaction (IntuiKit, INDIGO, ...)

Réifier l'interaction non localisée (non-widget)  
DPI (document-presentation-interaction)

Interaction vs. programmation  
scripting, end-user programming, etc.

## Questions ouvertes

Evaluation des boîtes à outils  
“simple things are simple, complex things are possible”  
valider par l'usage  
benchmarks

Mutualisation des efforts  
distribution des logiciels  
interopérabilité  
plate-forme