

# Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables

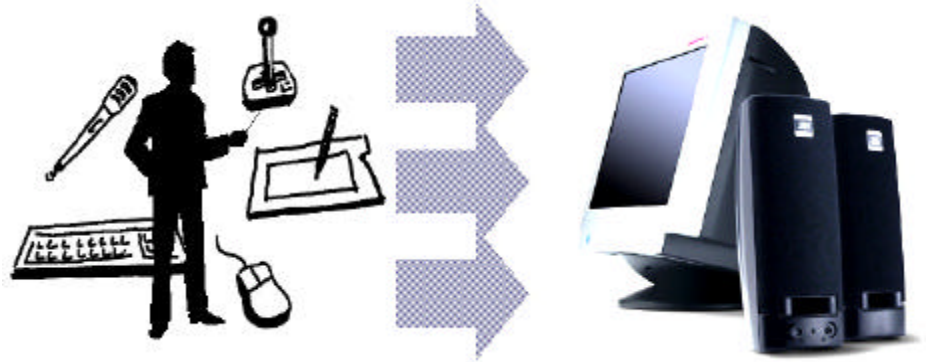
Pierre Dragicevic  
École des Mines de Nantes

9 mars 2004

# Introduction



- Système standard  
(Xerox Star, 1979)



- Système « adaptable en entrée »

# **I. L'interaction en entrée non-standard**

## **II. Modèles et outils existants**

## **III. Le modèle des configurations d'entrée**

## **IV. La boîte à outils ICON**

# L'interaction en entrée

- Entrées (manipulation) / Sorties (présentation)

- Dispositifs d'entrée

- Techniques d'interaction

- Une manière de se servir d'un dispositif d'entrée pour accomplir une tâche sur l'ordinateur [Foley et al. 90]
- Production de sorties (feedback)



*Suppression d'un fichier à la souris*

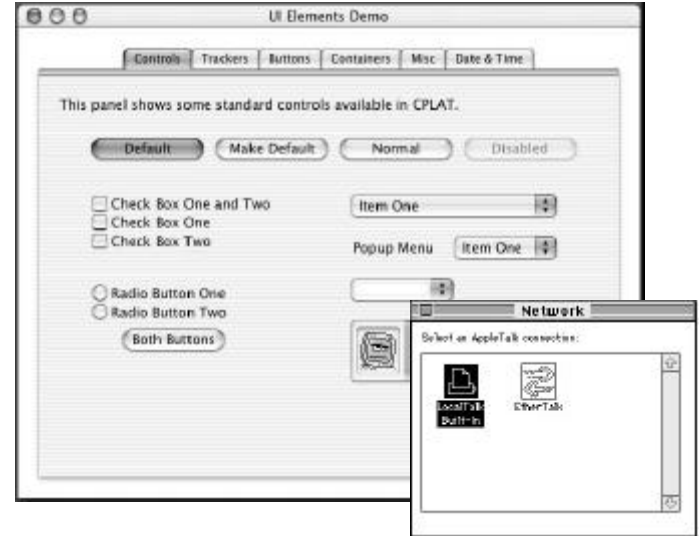
- Paradigme d'interaction

- Ensemble cohérent de t.i. employées ensemble ou qui reposent sur les mêmes principes

# L'interaction en entrée standard



+



- Dispositifs standard:
  - Clavier
  - Souris

- Techniques d'interaction standard:
  - Fenêtres, widgets, icônes
  - Saisie textuelle, clic, double-clic, glisser-déposer

(Interfaces **WIMP** = Windows, Icons, Menus, Pointer)

# Dispositifs d'entrée non standard

Des dispositifs adaptés:



à la **tâche**



à l'**utilisateur**



à l'**environnement**

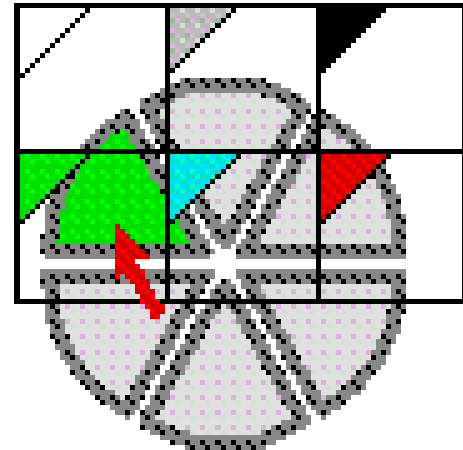
# Nouveaux paradigmes d'interaction

Rendre l'interaction plus directe et plus concise:

- Interaction gestuelle



- Outils semi-transparents



# Nouveaux paradigmes d'interaction

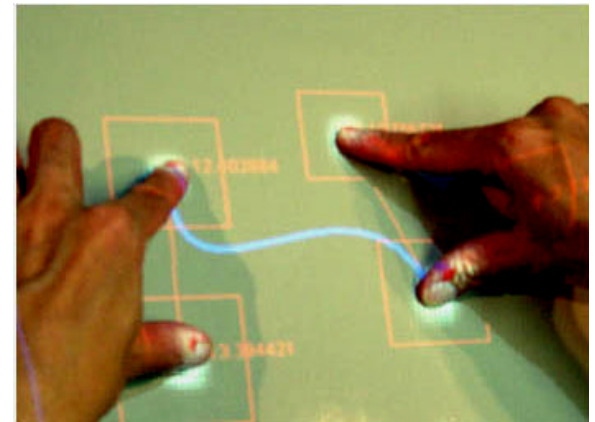
Mieux exploiter nos capacités motrices:

- Interaction bimanuelle



[Hinckley et al 98]

- Interaction « multidigitale »



SmartSkin [Rekimoto 2002]





# Application à l'informatique grand-public

---

- Enseignements tirés de la recherche:
  - Possibilité de « **faire mieux** »
  - Importance du **contexte**:
    - Tâches
    - Utilisateurs (savoirs-faire, handicaps,...)
    - Environnements (mobile, bruyant,...)
- Évolution de l'informatique grand-public:
  - **Diversification des contextes**
  - Prolifération de **nouveaux périphériques**
  - Proportion croissante d'**utilisateurs avancés**

® *Nécessité de systèmes « adaptables en entrée »*



# Application à l'informatique grand-public

---

- Définition de l'adaptabilité en entrée

## **Contrôlabilité:**

Capacité à exploiter efficacement des *entrées enrichies*, ou à utiliser les dispositifs standard de façon *plus efficace*.

## **Accessibilité:**

Capacité à exploiter des *entrées appauvries*.

## **Configurabilité:**

Capacité de choisir librement *la manière dont on veut utiliser les dispositifs* pour contrôler l'application.

# Application à l'informatique grand-public

## ■ Adaptabilité des systèmes interactifs actuels

### ■ Au niveau du **système d'exploitation**:

#### - **Contrôlabilité:**

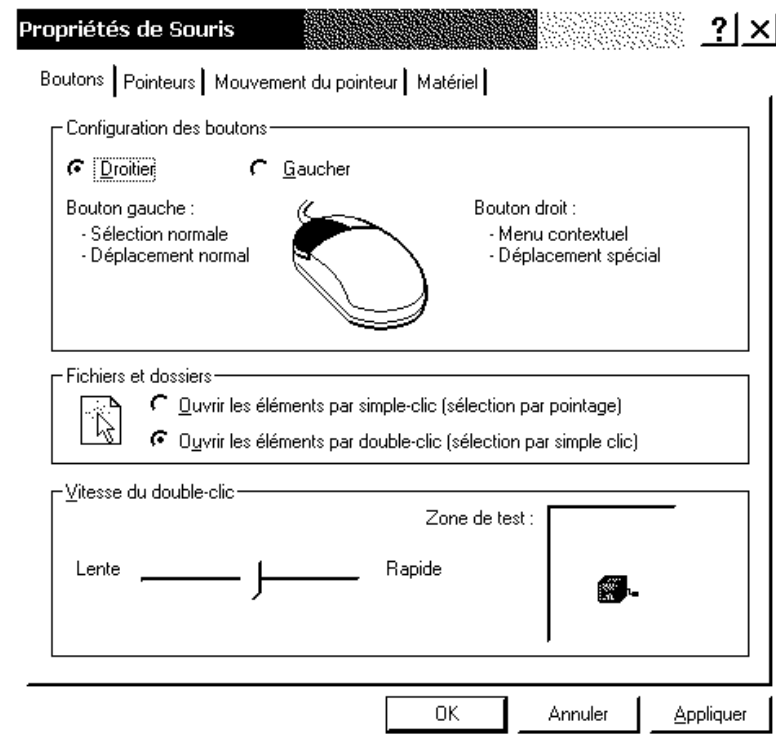
- Fusion des entrées multiples
- Entrées étendues ignorées

#### - **Accessibilité:**

- Entrées standard requises
- Peu d'outils d'accessibilité

#### - **Configurabilité:**

- Limitée à quelques boîtes de dialogue

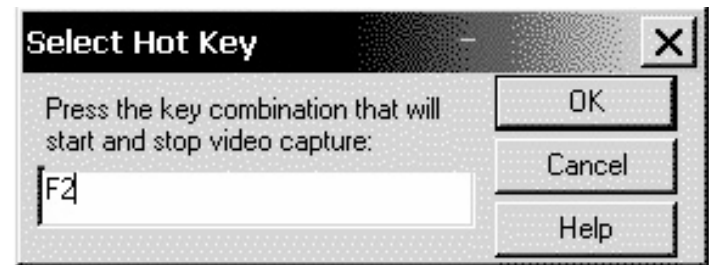


# Application à l'informatique grand-public

- Adaptabilité des systèmes interactifs actuels

- Au niveau des **applications**:

- Applications traditionnelles →



- Applications spécialisées:

- Ex: Photoshop, Cubase,...

- Jeux vidéo →

- Accessibilité spécifique





# Application à l'informatique grand-public

---

- Il est encore extrêmement difficile de développer des applications interactives adaptables en entrée
  - Les **outils** qui servent à les construire sont **câblés** pour l'utilisation exclusive:
    - D'une souris et d'un clavier
    - D'un ensemble limité de techniques d'interaction standard

Ⓒ Nécessité de nouveaux **modèles**

Ⓒ Nécessité de nouveaux **outils**

**I. L'interaction en entrée non-standard**

**II. Modèles et outils existants**

**III. Le modèle des configurations d'entrée**

**IV. La boîte à outils ICON**



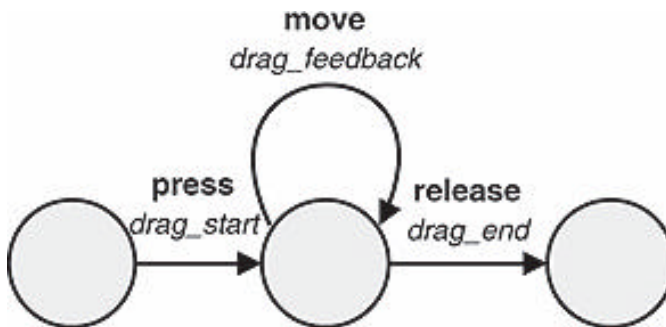
# Les modèles en IHM

---

- Les modèles d'interaction
  - La manipulation directe [Shneiderman 83]
  - L'interaction instrumentale [Beaudouin-Lafon 97]
- Les modèles architecturaux
  - Les approches linguistiques: Seeheim [Pfaff 85], Arch/Slinky [UIMS 92]
  - Les approches à agents: MVC [Schmucker 86], PAC [Coutaz 87]
- Les modèles d'entrées
  - Les approches logiques: GKS/PHIGS [Eckert et al 79], tâches d'interaction [Foley et al 84]
  - Les approches physiques: Taxinomies de Buxton [Buxton 83] et de Card [Card et al 90]

# Les approches formelles

- Les formalismes IHM
  - Interacteurs d'York [Duke & Harisson 93]  
et de CNUCE [Paternò & Faconti 92]
- Les systèmes à états et transitions
  - Automates
  - Réseaux de Petri (ICO [Palanque & Bastide 93])







# Les outils de développement

---

- Les boîtes à outils d'interface
  - Ex: X Toolkit, Swing, ...
  - Abstraction des entrées (**Événements**)
  - Jeux d'objets interactifs (**Widgets**)
  - Mécanismes d'**aiguillage** prédéfinis (pick, focus, ...)
  - **Comportements** internes prédéfinis (feel)

® *Très difficiles à étendre*



# Les outils de développement

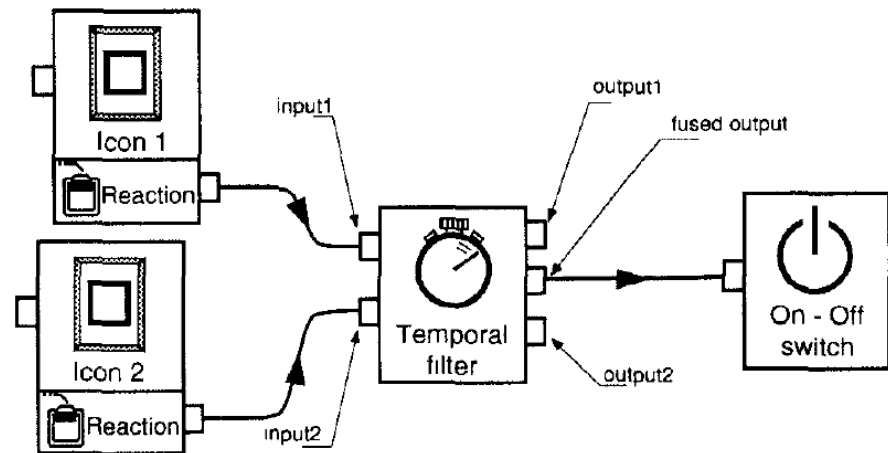
---

- Les boîtes à outils avancées
  - ArtKit/Subarctic [Tyson et al 90, Hudson & Smith 96]
  - Garnet/Amulet [Myers 90, Myers et al 97]
- Les boîtes à outils Post-WIMP
  - Interaction gestuelle et multimodale: Satin [Hong & Landay 2000], « MultiModal » Subarctic [Mankoff et al 2000] , ...
  - Interaction multi-pointeurs, outils semi-transparentes: MMM [Bier & Freeman 91], CPN2000 [Beaudouin-Lafon & Lassen 2000] , ...
  - Interaction multi-dispositifs: Context Toolkit [Salber & al 99], Phidgets [Greenberg & Fitchett 2001], Boîtes à outils 3D

# Les outils visuels

## ■ Les éditeurs d'interaction 2D

- À base de contraintes: ThingLab [Borning 79], Fabrik [Ingalls et al 88]
- À états+transitions: PetShop [Bastide, Navarre & Palanque 02]
- À flots de données:  
Whizz'Ed  
[Chatty 94, Esteban 97]



## ■ Les éditeurs de comportements 3D

- Senseurs/actions: Maya RTA, Blender
- Flots de données: Virtools Dev, VRED [Jacob et al 90]



# Bilan

---

- Peu de travaux sur les entrées
- Aucun outil « Post-WIMP » ne combine:
  - Techniques d'interaction évoluées
  - Dispositifs multiples et variés
- Aucun modèle ou outil ne traite à la fois:
  - La contrôlabilité
  - L'accessibilité
  - La configurabilité

**I. L'interaction en entrée non-standard**

**II. Modèles et outils existants**

**III. Le modèle des configurations d'entrée**

**IV. La boîte à outils ICON**

# Le modèle des configurations d'entrée

- Approche:
  - **Découpler** au maximum les dispositifs physiques des applications
  - Fournir un moyen flexible de **relier** les dispositifs aux applications



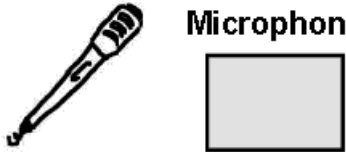
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques

Clavier

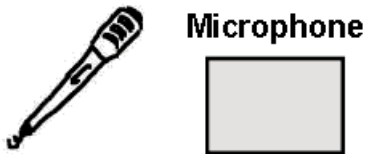


Microphone

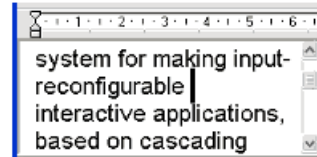
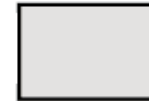


# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application



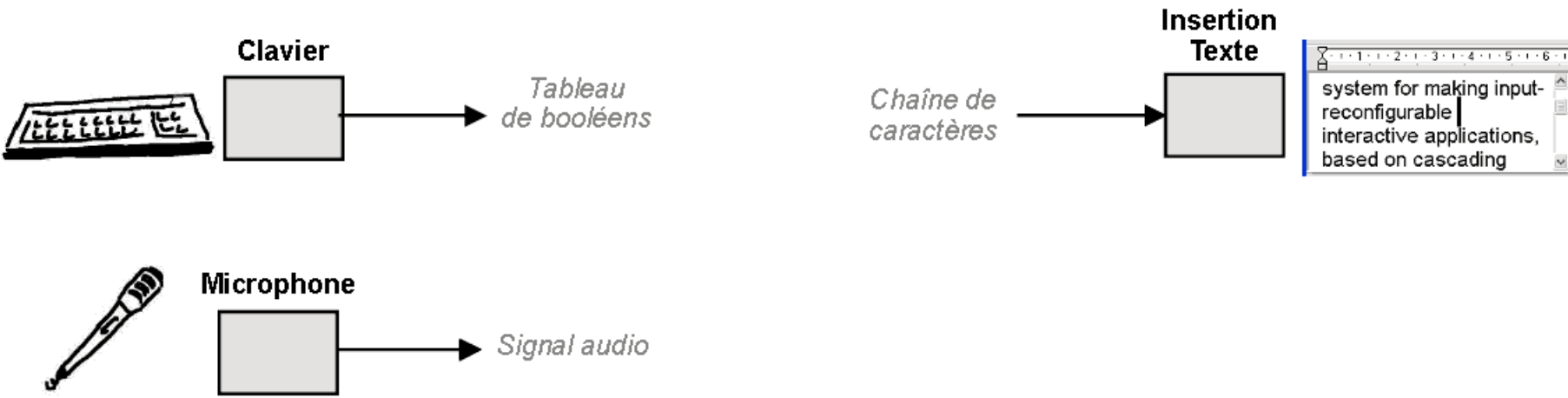
Insertion  
Texte





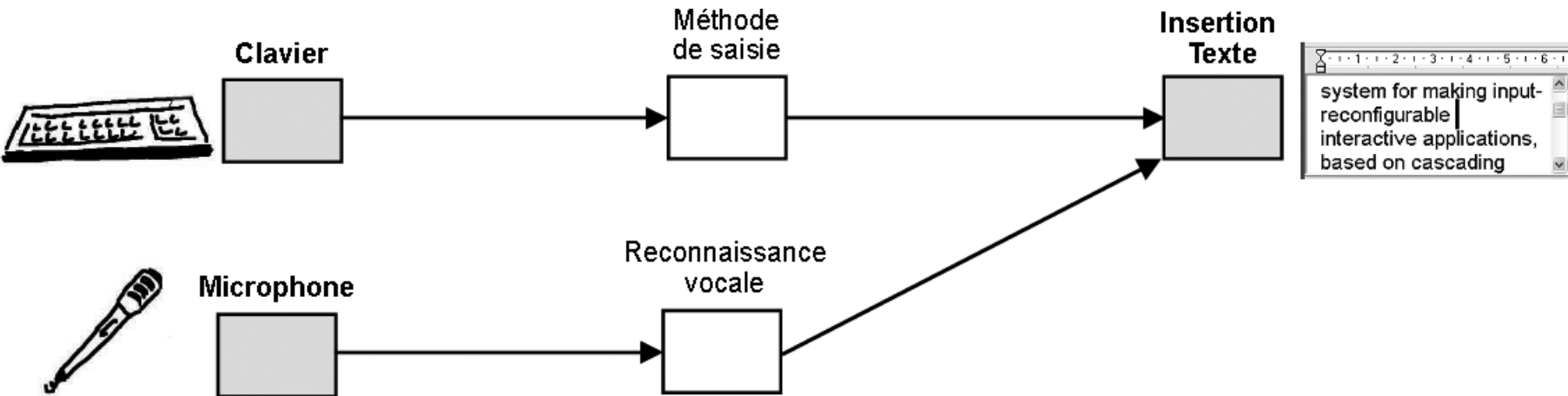
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application



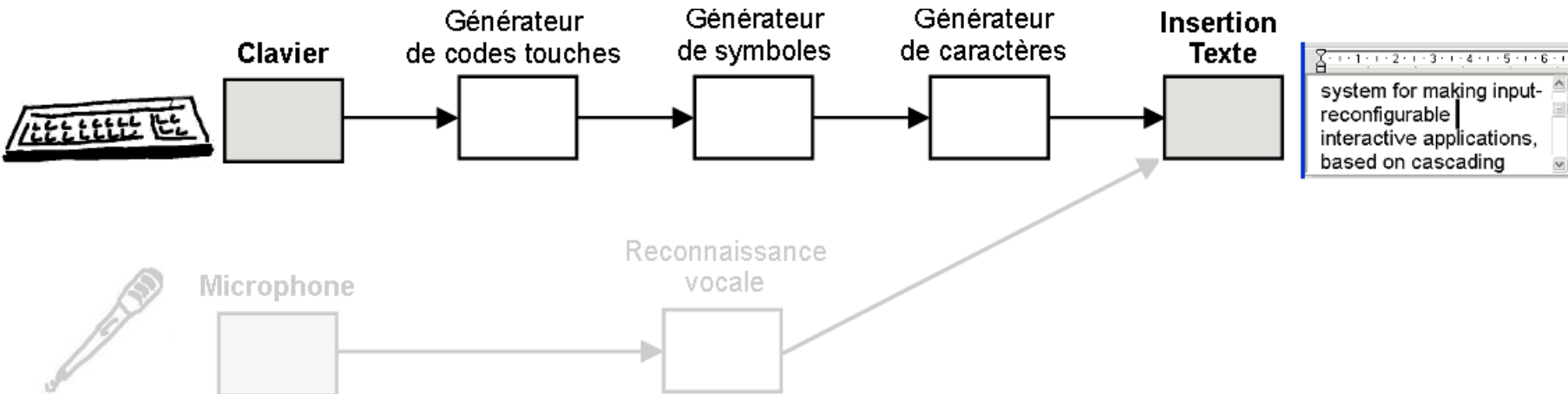
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application
  - **Adaptateurs**



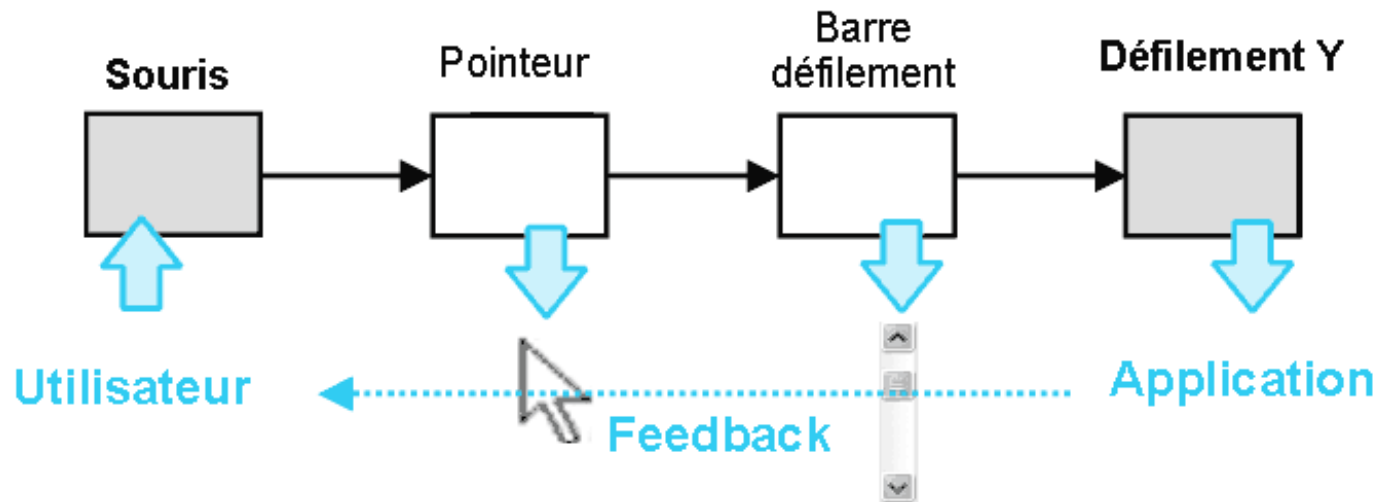
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application
  - **Adaptateurs** cumulables en série



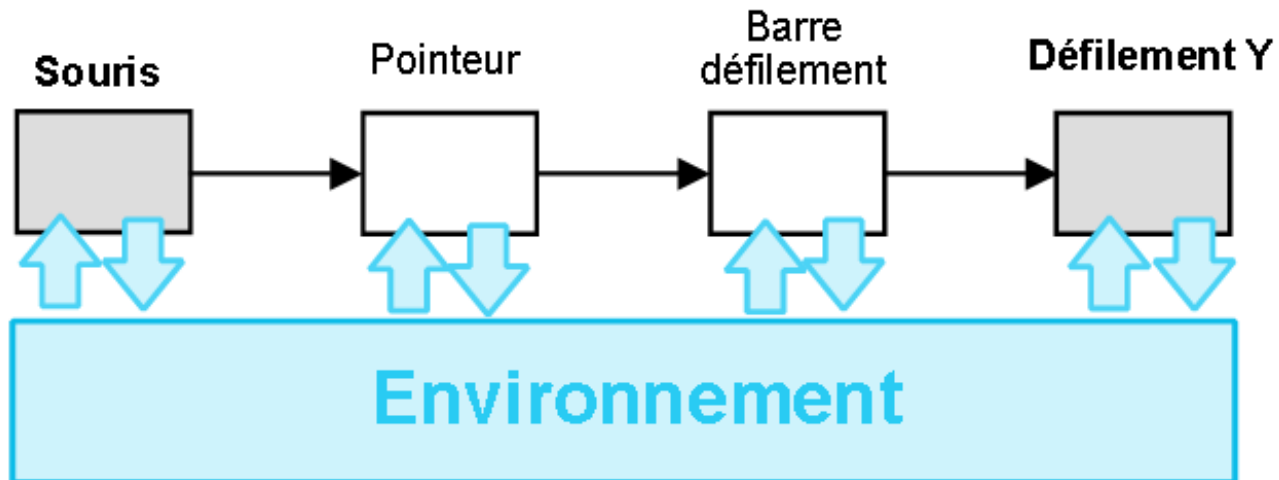
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application
  - **Adaptateurs** cumulables en série
- Feedback:



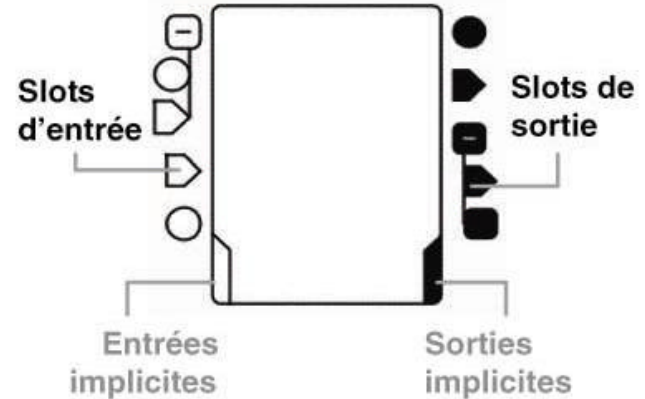
# Le modèle des configurations d'entrée

- Entités:
  - **Dispositifs** d'entrée physiques
  - **Points d'entrée** vers l'application
  - **Adaptateurs** cumulables en série
- Feedback:



# Le modèle des configurations d'entrée

- **Dispositif généralisé:**
  - Boîte noire **paramétrable**
  - **Slots** d'entrée et de sortie
    - Types simples
    - Structurables
  - **Traitement** de données
  - **E/S implicites**

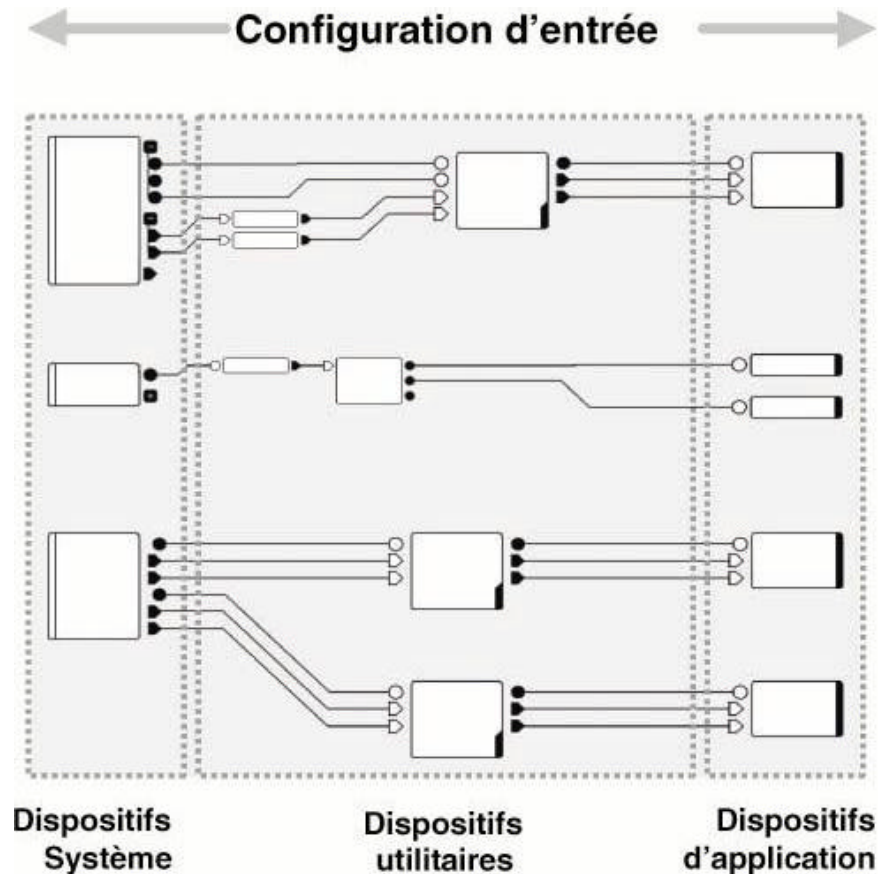


	any	double	int	boolean	string object	null
entrée						
sortie						

- **3 types de dispositifs:**
  - **Système** (ressources matérielles, présence variable)
  - **Utilitaire** (processus logiciels, toujours présents)
  - **D'application** (propres à chaque application)

# Le modèle des configurations d'entrée

- **Connexion:**
  - Arc orienté entre un slot d'entrée et de sortie
- **Configuration d'entrée:**
  - Ensemble de dispositifs **ystème** et **d'application** reliés par des dispositifs **utilitaires**

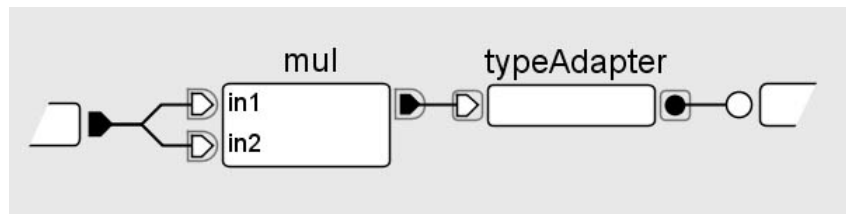


# Le modèle des configurations d'entrée

- **ICoM** (Input Configuration Model)

- Décrit la **structure** et les **mécanismes** des configurations d'entrée
- Système à flot de données **compositionnel**

- **Polymorphisme**



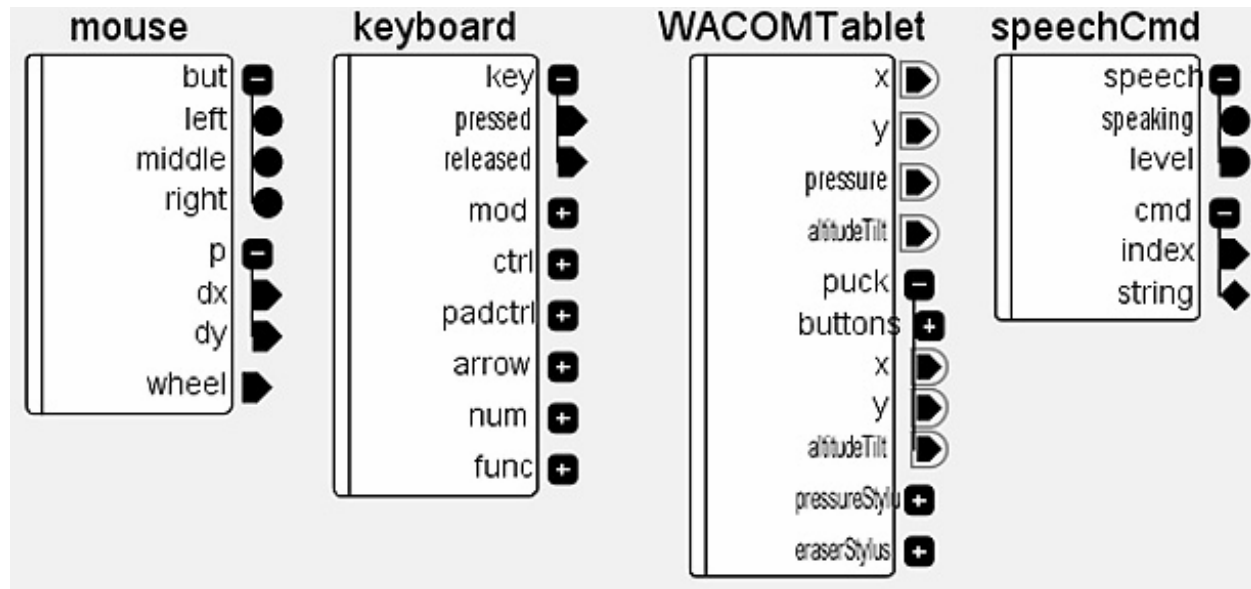
- Modèle d'exécution **réactif** (Lustre [[Halbwachs et al 91](#)], Esterel [[Berry 87, 99](#)])



- I. L'interaction en entrée non-standard
- II. Modèles et outils existants
- III. Le modèle des configurations d'entrée
- IV. La boîte à outils InputConfigurator**

# La boîte à outils ICon

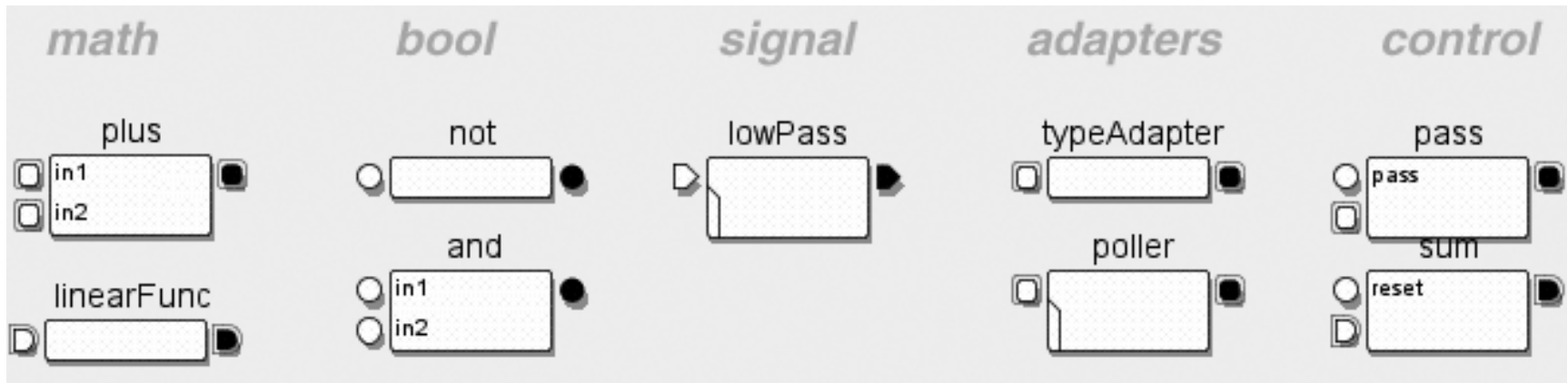
- Les dispositifs système



- Souris, clavier, tablettes graphiques
- Reconnaissance vocale
- Dispositifs de jeu, dispositifs 3D isométriques
- Instruments et contrôleurs MIDI

# La boîte à outils ICon

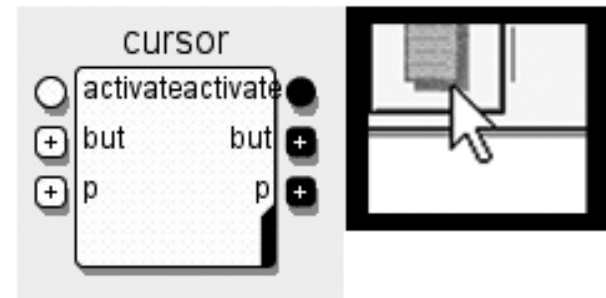
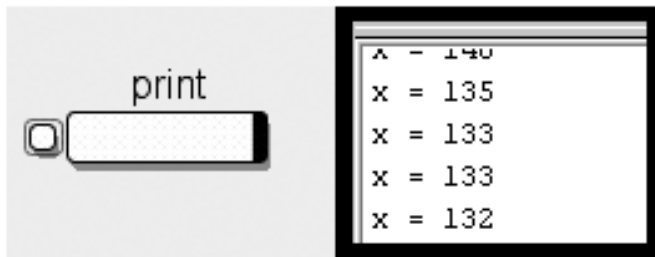
- Les dispositifs utilitaires



- Dispositifs de **traitement**

# La boîte à outils ICon

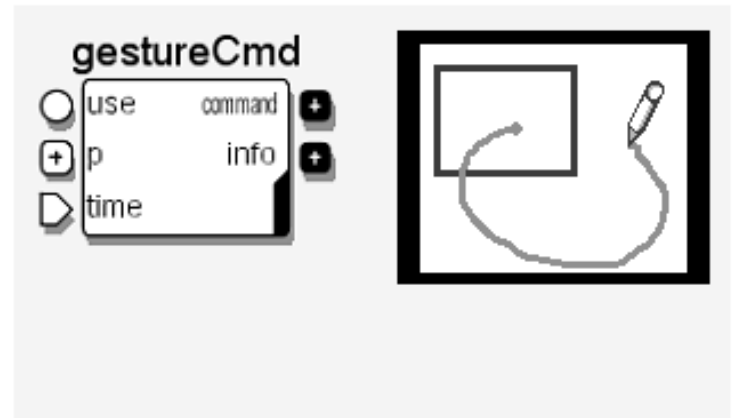
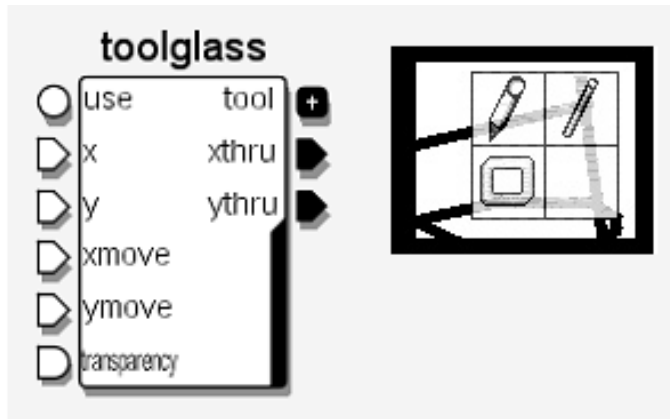
- Les dispositifs utilitaires



- Dispositifs de **retour graphique**

# La boîte à outils ICon

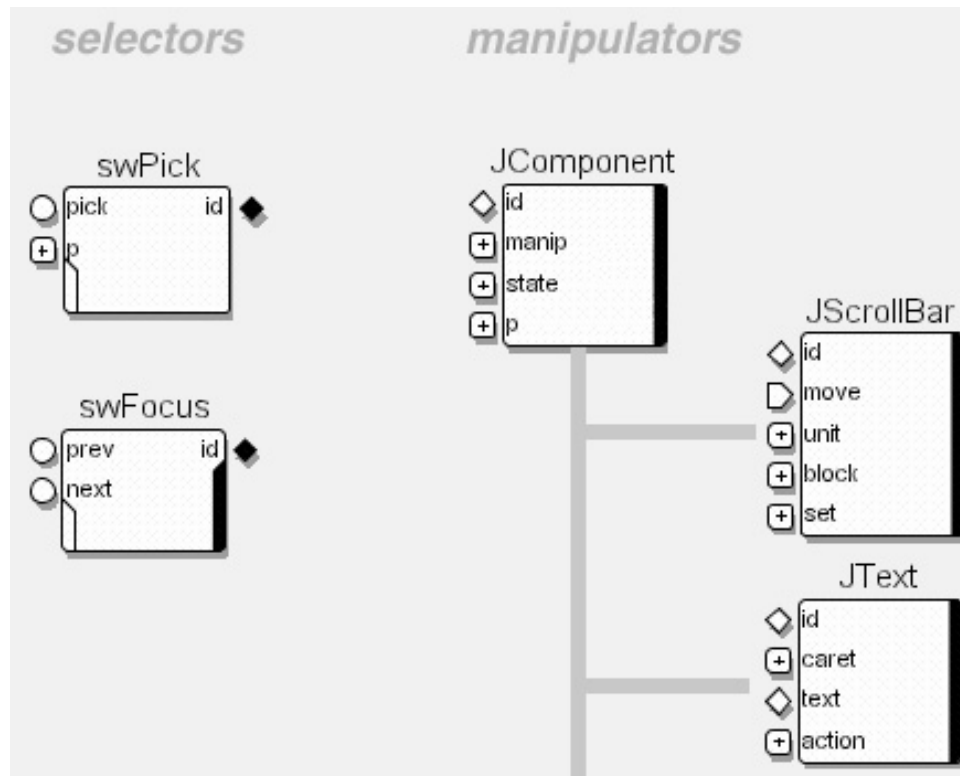
- Les dispositifs utilitaires



- Techniques d'interaction

# La boîte à outils ICon

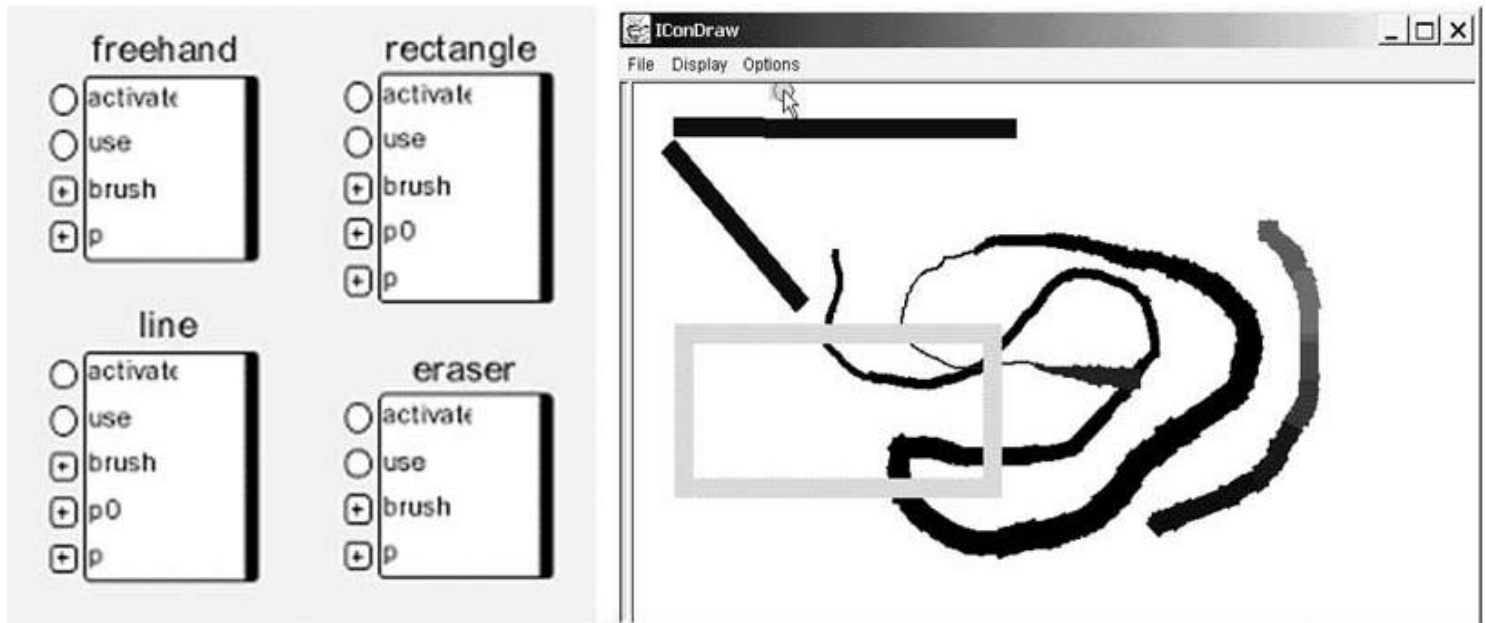
- Les dispositifs Swing



- Interfacent ICon avec Swing
- Permettent un **contrôle générique** des applications Swing

# La boîte à outils ICon

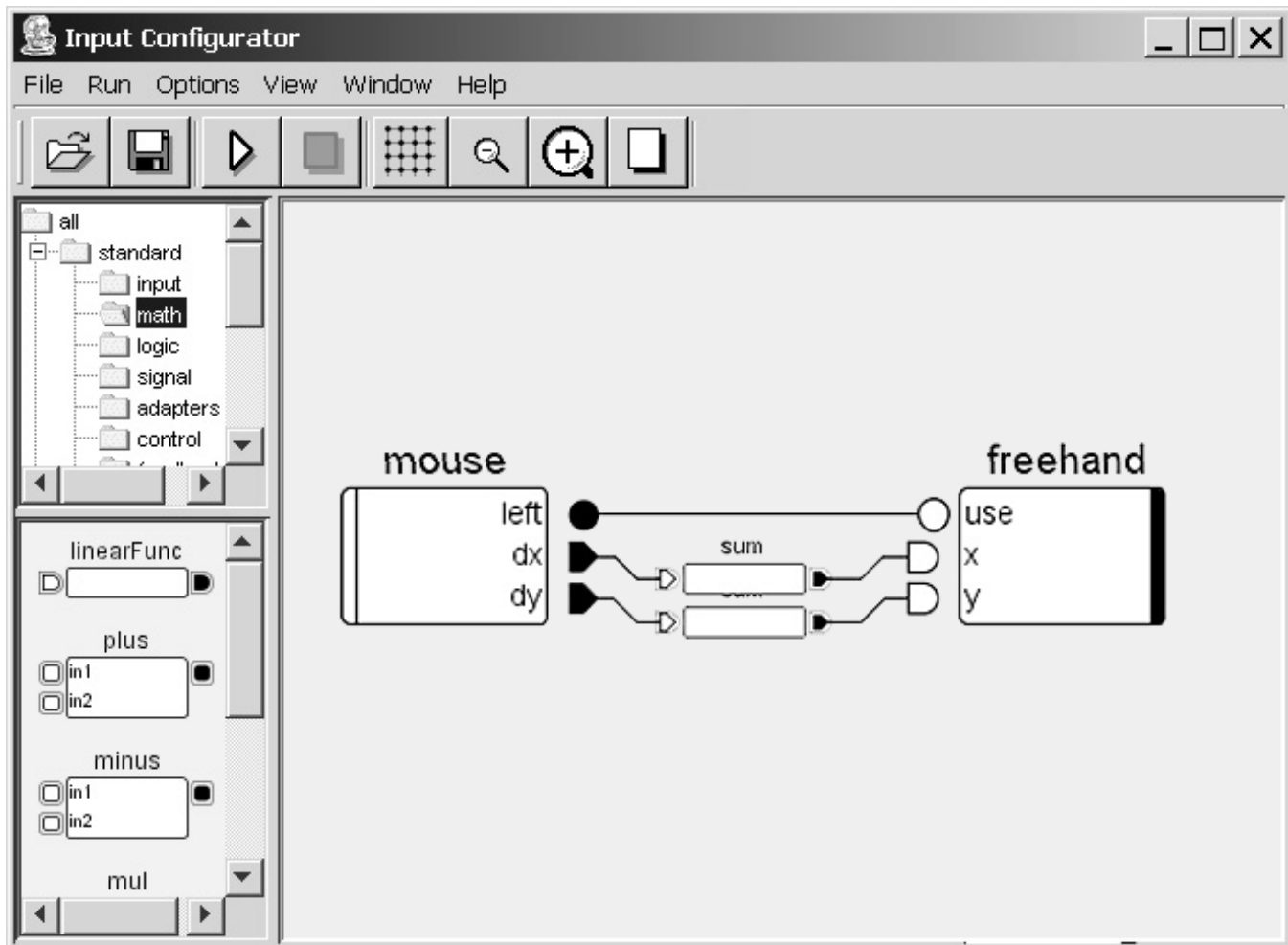
- Les dispositifs d'application



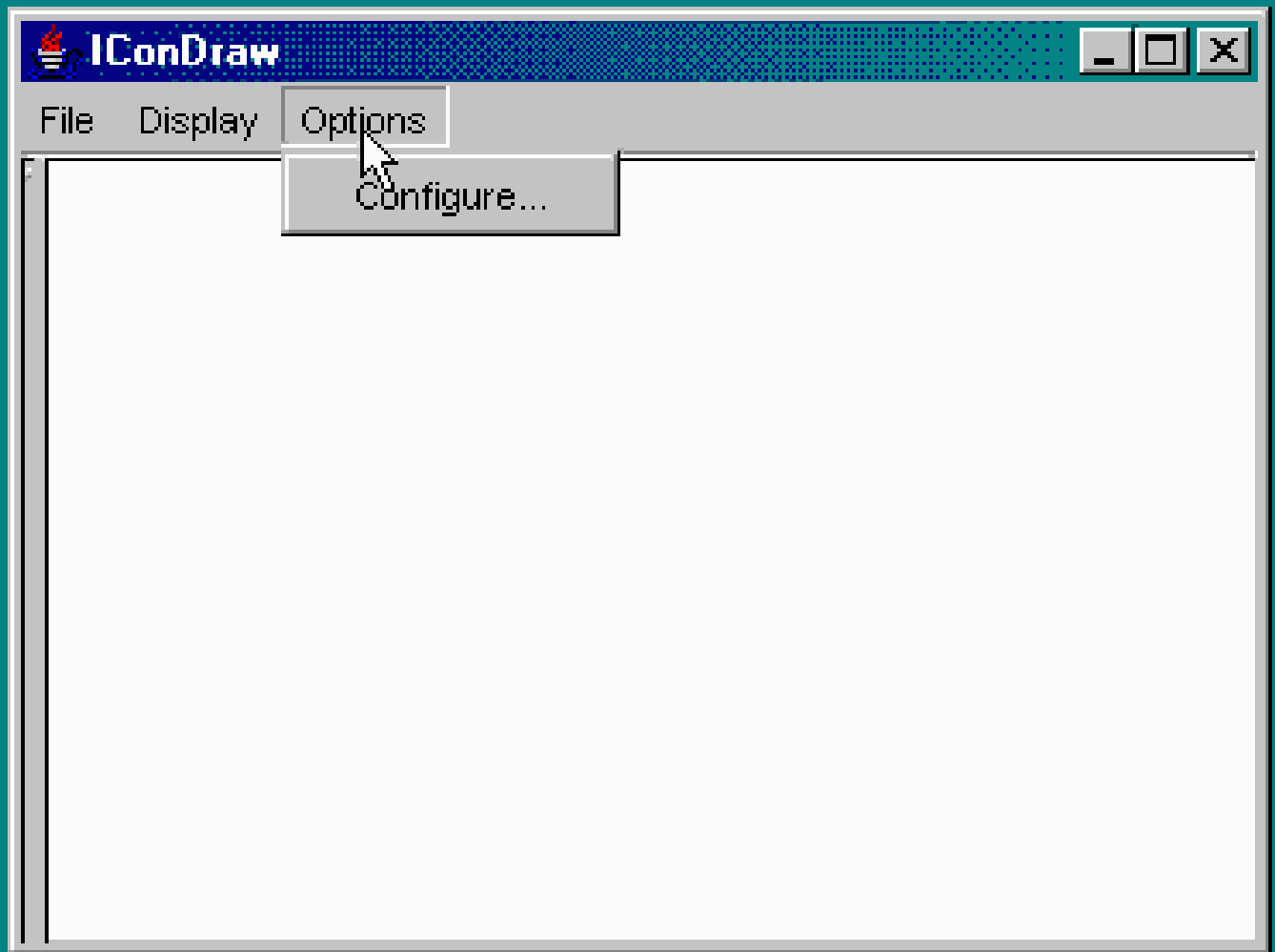
- Déclarés par chaque application
- Exemple: IConDraw

# La boîte à outils ICon

- L'éditeur de configurations

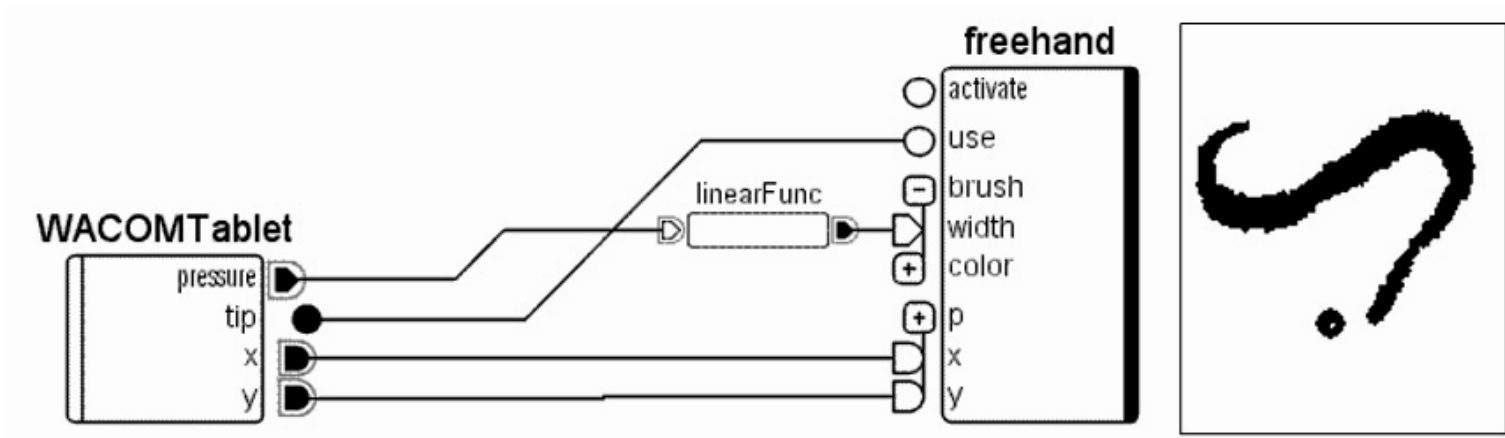






# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité
  - Dessin sensible à la pression dans IConDraw



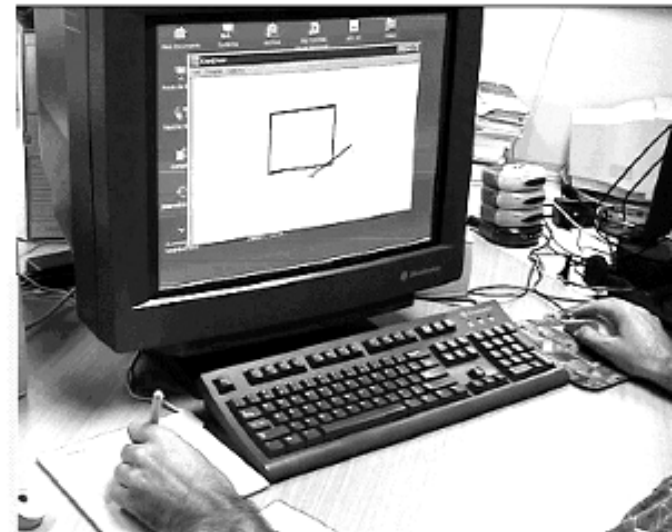
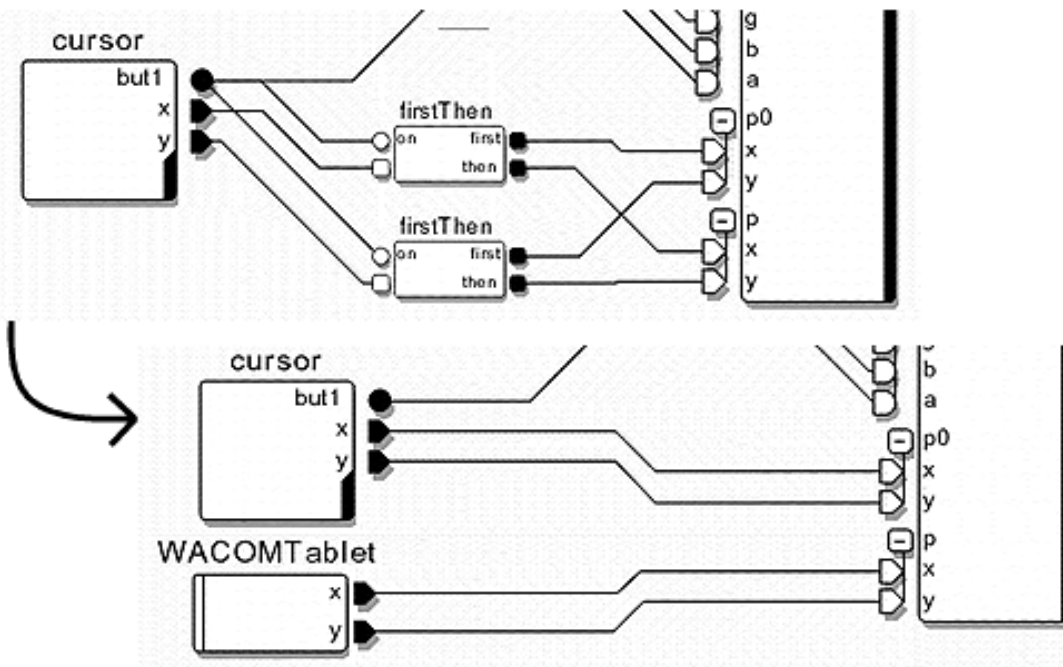
# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité



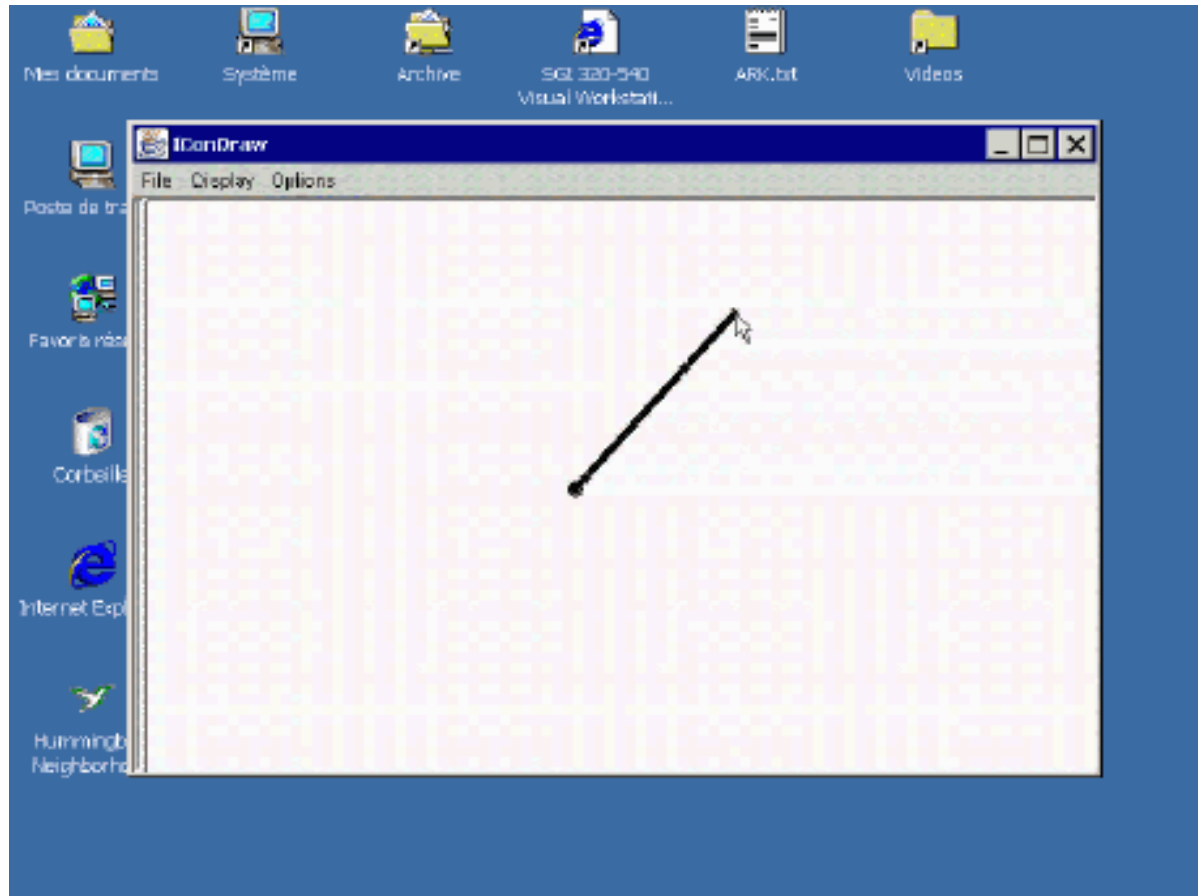
# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité
  - Tracé bimanuel de lignes dans IConDraw



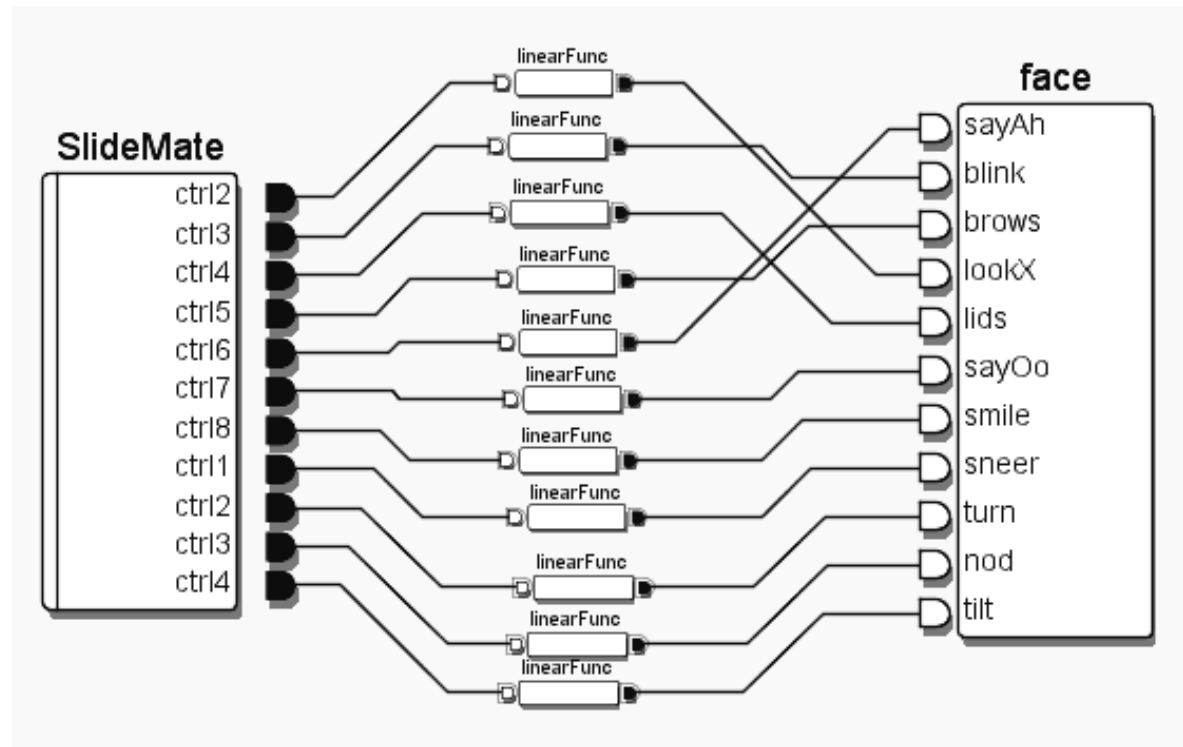
# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité



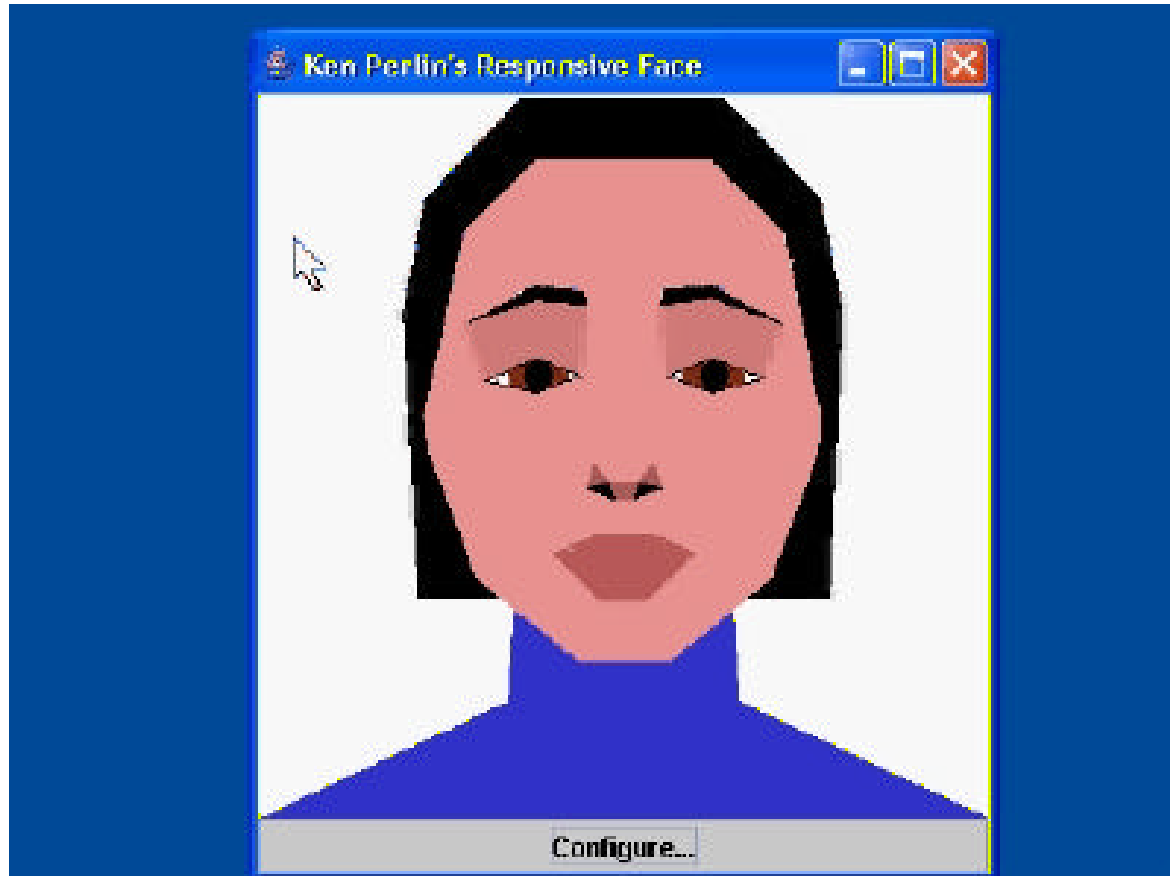
# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité
  - Contrôle intégral du Responsive Face [Perlin 97]



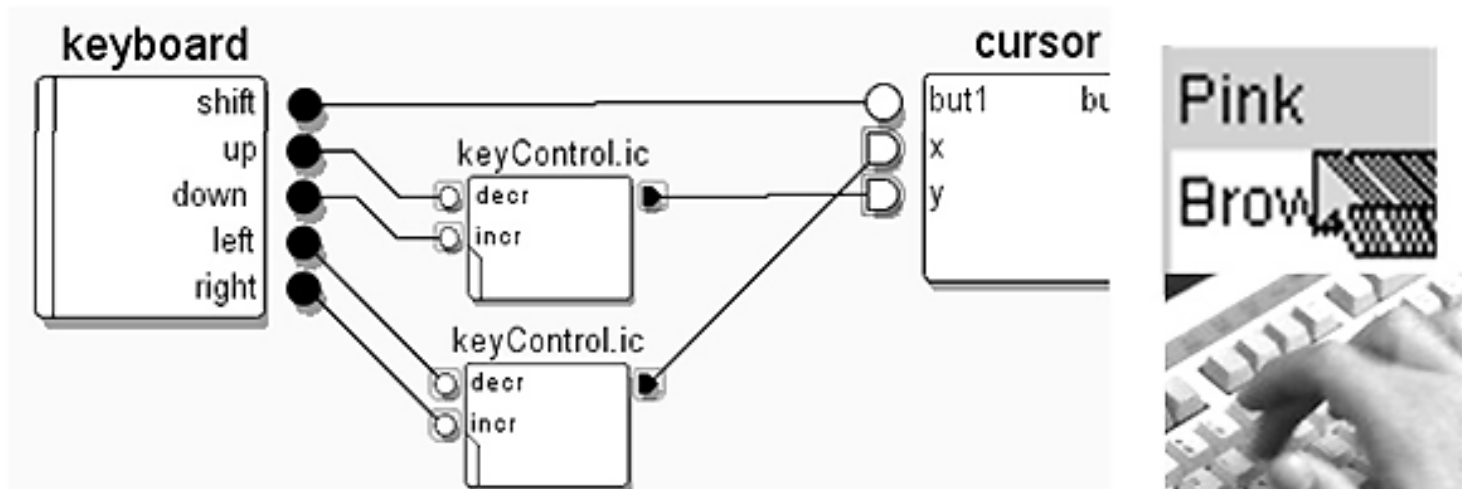
# La boîte à outils ICon - Exemples

- Améliorer la contrôlabilité



# La boîte à outils ICon - Exemples

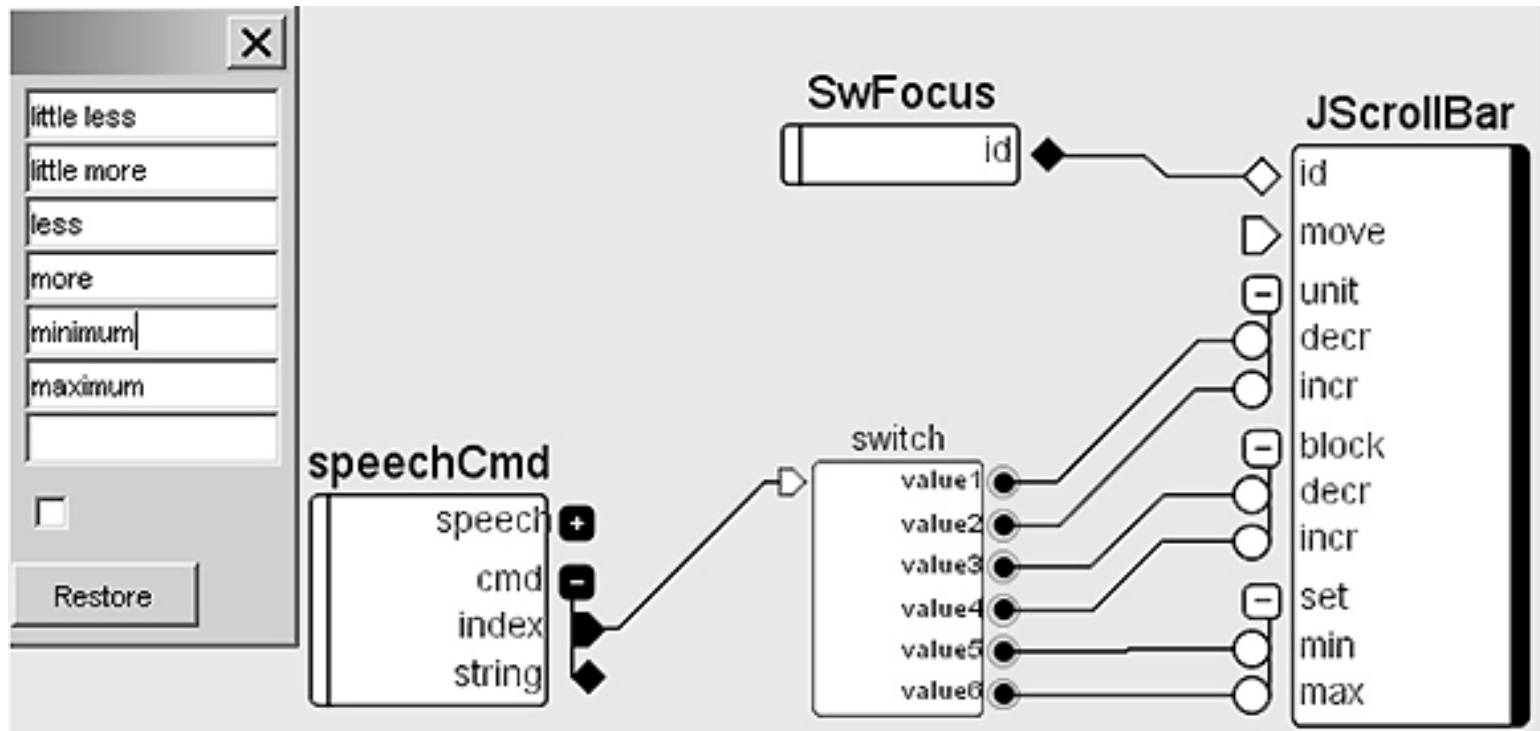
- Améliorer l'accessibilité
  - Pointage au clavier





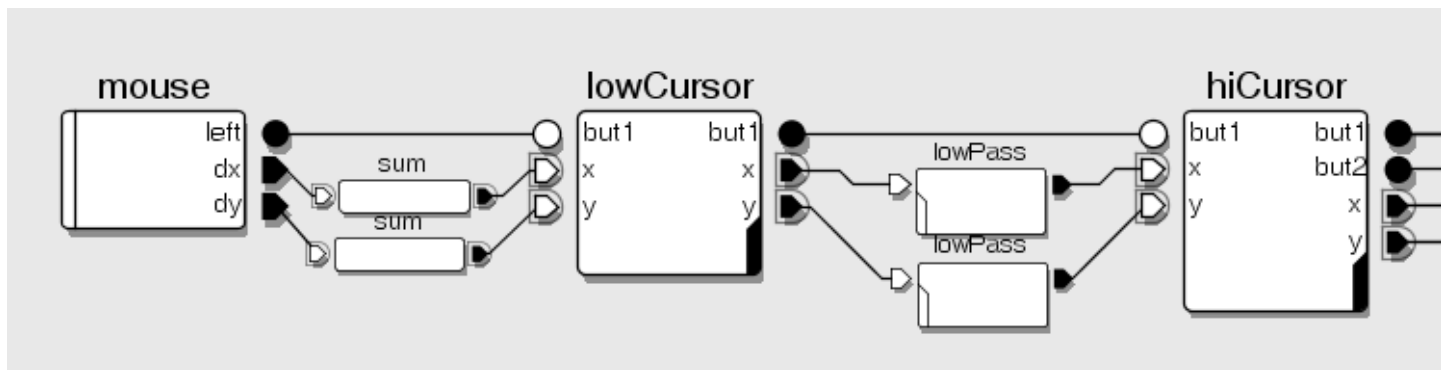
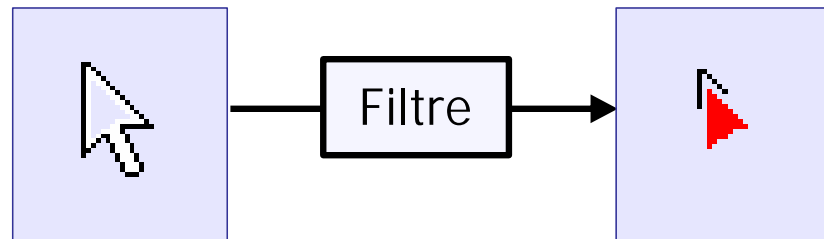
# La boîte à outils ICon - Exemples

- Améliorer l'accessibilité
  - Contrôle vocal des barres de défilement



# La boîte à outils ICon - Exemples

- Construire des techniques d'interaction
  - Le pointage augmenté

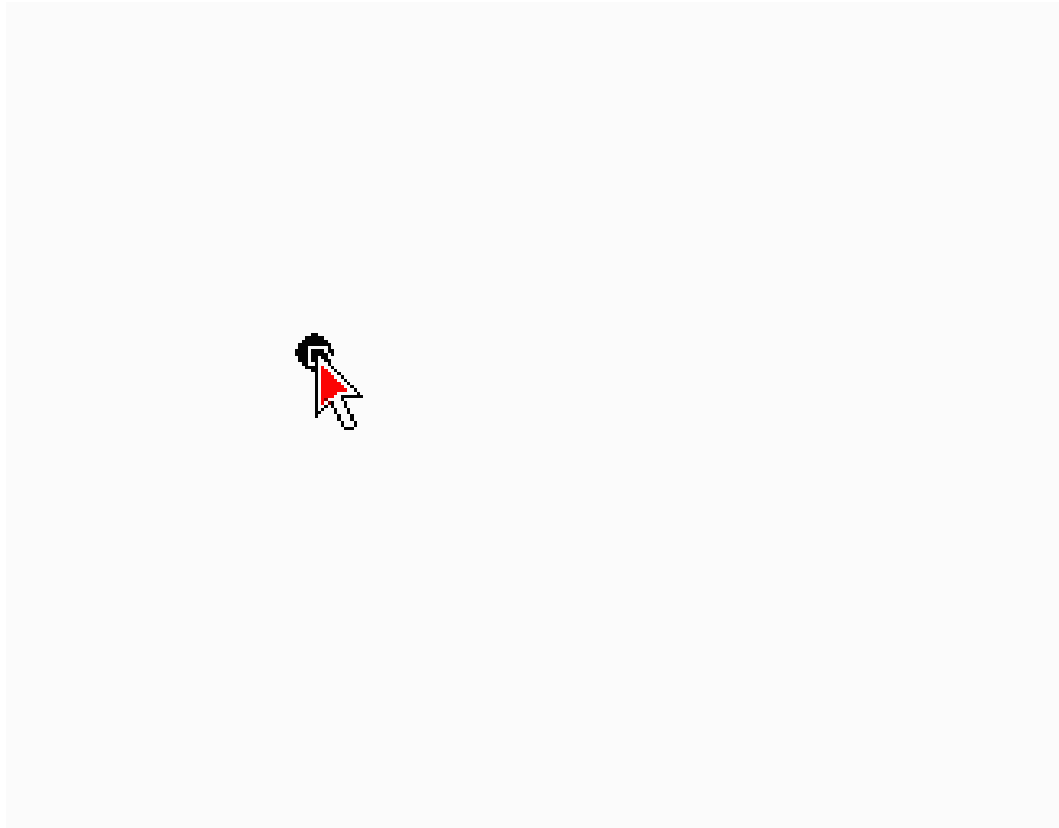




# La boîte à outils ICon - Exemples

---

- Construire des techniques d'interaction





# La boîte à outils ICon - Exemples

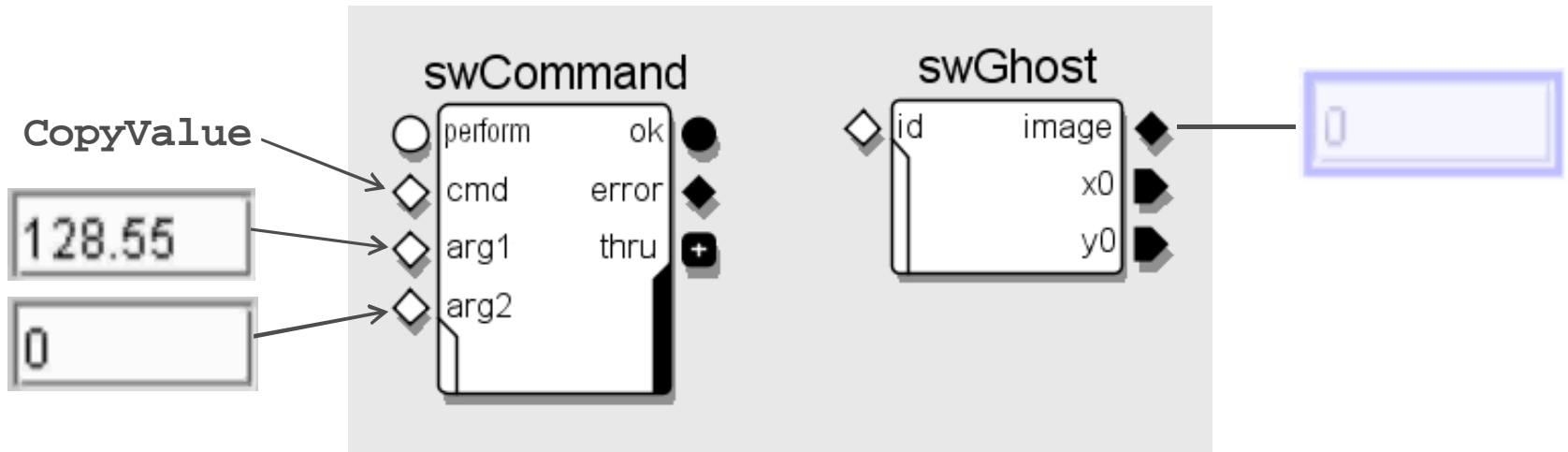
---

- Construire des techniques d'interaction



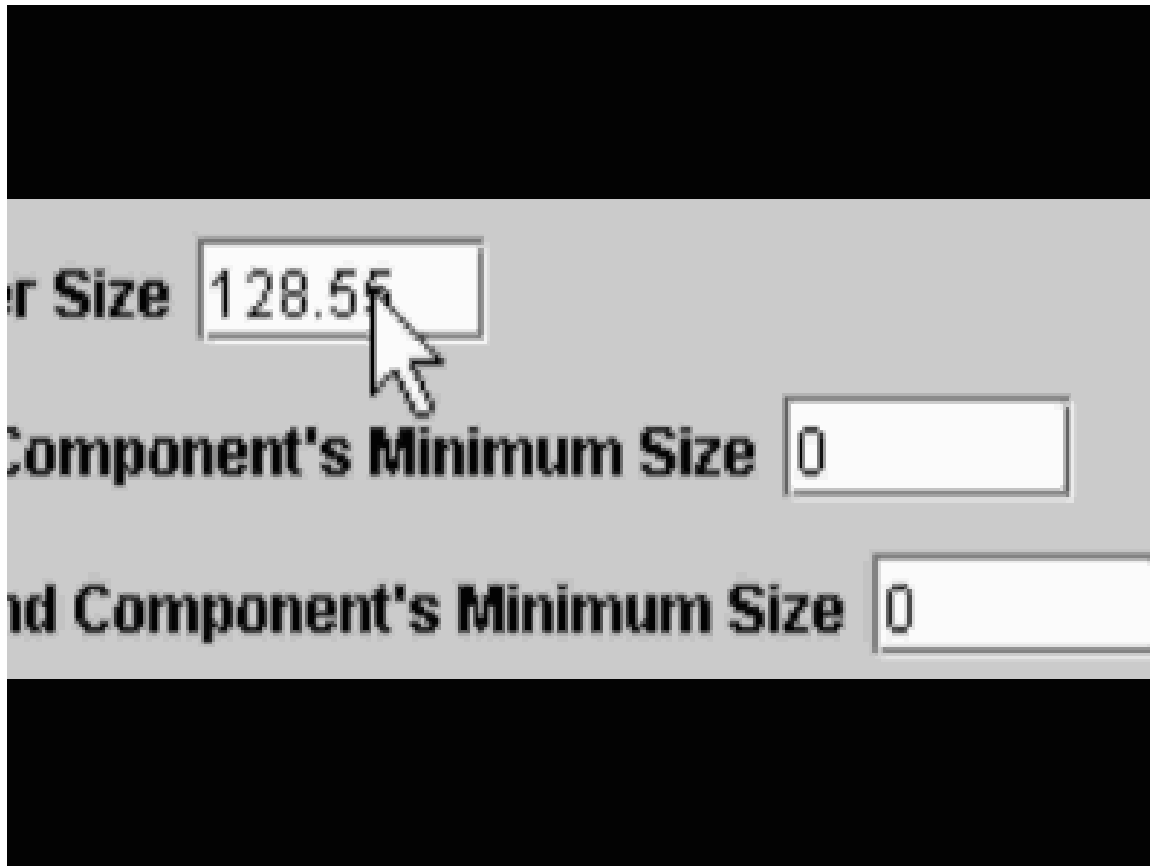
# La boîte à outils ICon - Exemples

- Construire des techniques d'interaction
  - Le glisser-déposer sur des widgets



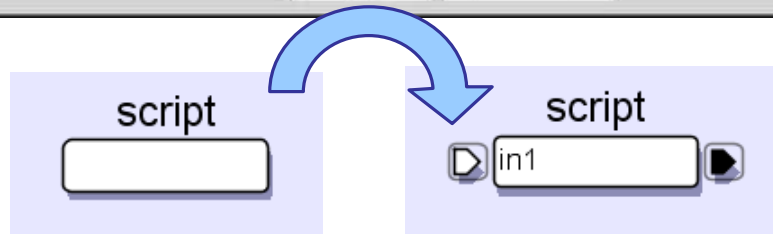
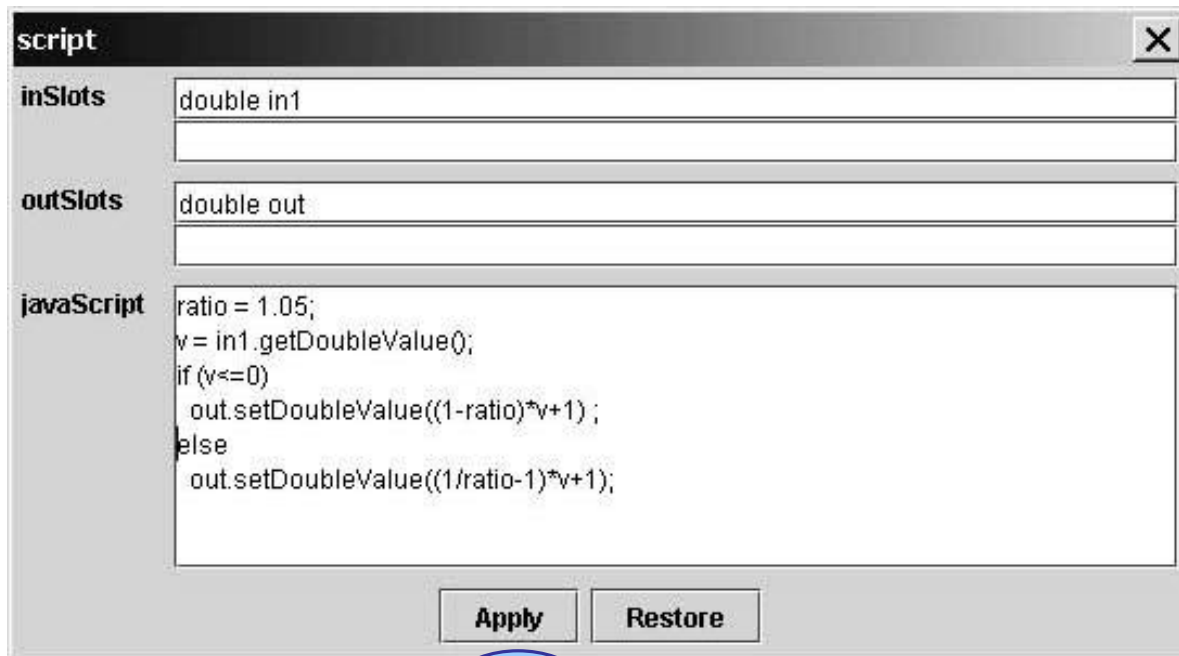
# La boîte à outils ICon - Exemples

- Construire des techniques d'interaction



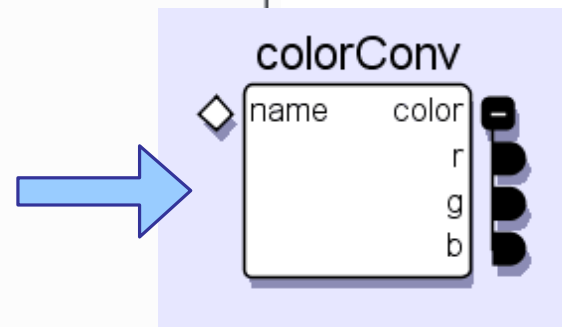
# La boîte à outils ICon

- Décrire de nouveaux dispositifs
  - Par un langage de script



## ■ En Java

```
public class DColorConv extends AbstractDevice {  
  
    // déclaration des slots  
  
    public final In name = addIn("name", SlotType.STRING);  
    public final Out r = addOut("color.r", SlotType.DOUBLE);  
    public final Out g = addOut("color.g", SlotType.DOUBLE);  
    public final Out b = addOut("color.b", SlotType.DOUBLE);  
  
    (...)  
  
    // mise à jour des slots de sortie  
  
    public void update() {  
        if (name.hasSignal()) {  
            String s = name.getStringValue();  
            boolean found = false;  
            for (int i=0; i<COLOR_NAMES.length && !found; i++)  
                found = (COLOR_NAMES[i].equals(s));  
            if (found) {  
                r.setDoubleValue(COLOR_R[i-1]);  
                g.setDoubleValue(COLOR_G[i-1]);  
                b.setDoubleValue(COLOR_B[i-1]);  
            }  
        }  
    }  
}
```







# Utilisateurs d'ICON

---

## ■ Tâches:

- Sélection de configurations
- Personnalisation superficielle
- Modifications importantes
- Prototypage
- Extension de la bibliothèque d'ICON
- Développement d'applications compatibles ICON

## ■ Utilisateurs potentiels:

- Tout utilisateur
- Utilisateurs avancés
- Informaticiens
- Concepteurs d'IHM
- Programmeurs
- Programmeurs

® **Utilisabilité du configurateur ?**

® **Utilisabilité des configurations d'entrée ?**

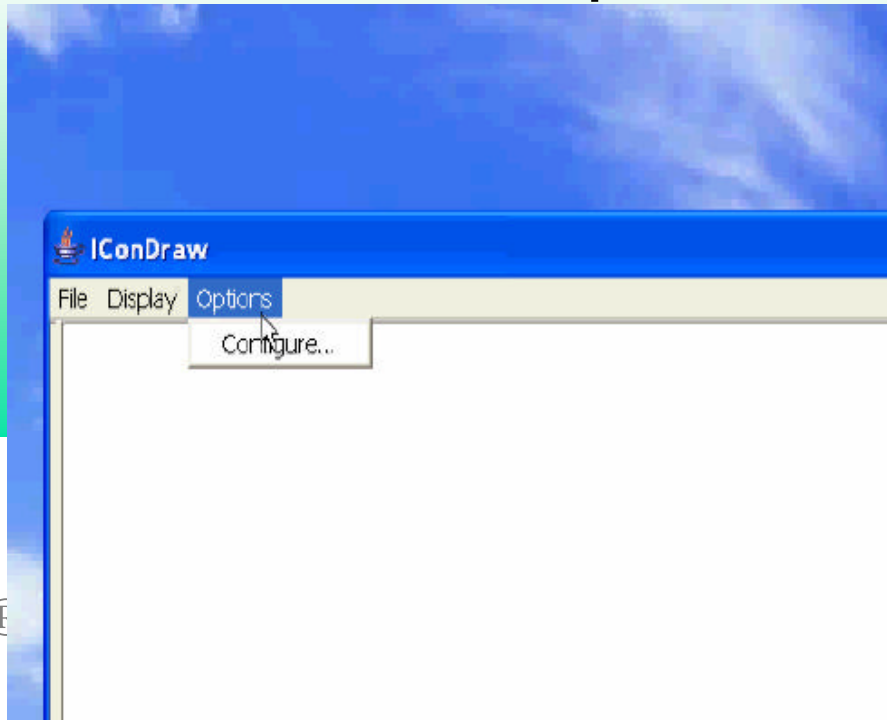
# Utilisateurs d'Icon

- Tâches:

- Sélection de configurations
- **Personnalisation superficielle**

- Utilisateurs potentiels:

- Tout utilisateur
- **Utilisateurs avancés**
- Informaticiens
- Concepteurs d'IHM
- Programmeurs
- Programmeurs



entrée ?



# Utilisateurs d'ICON

---

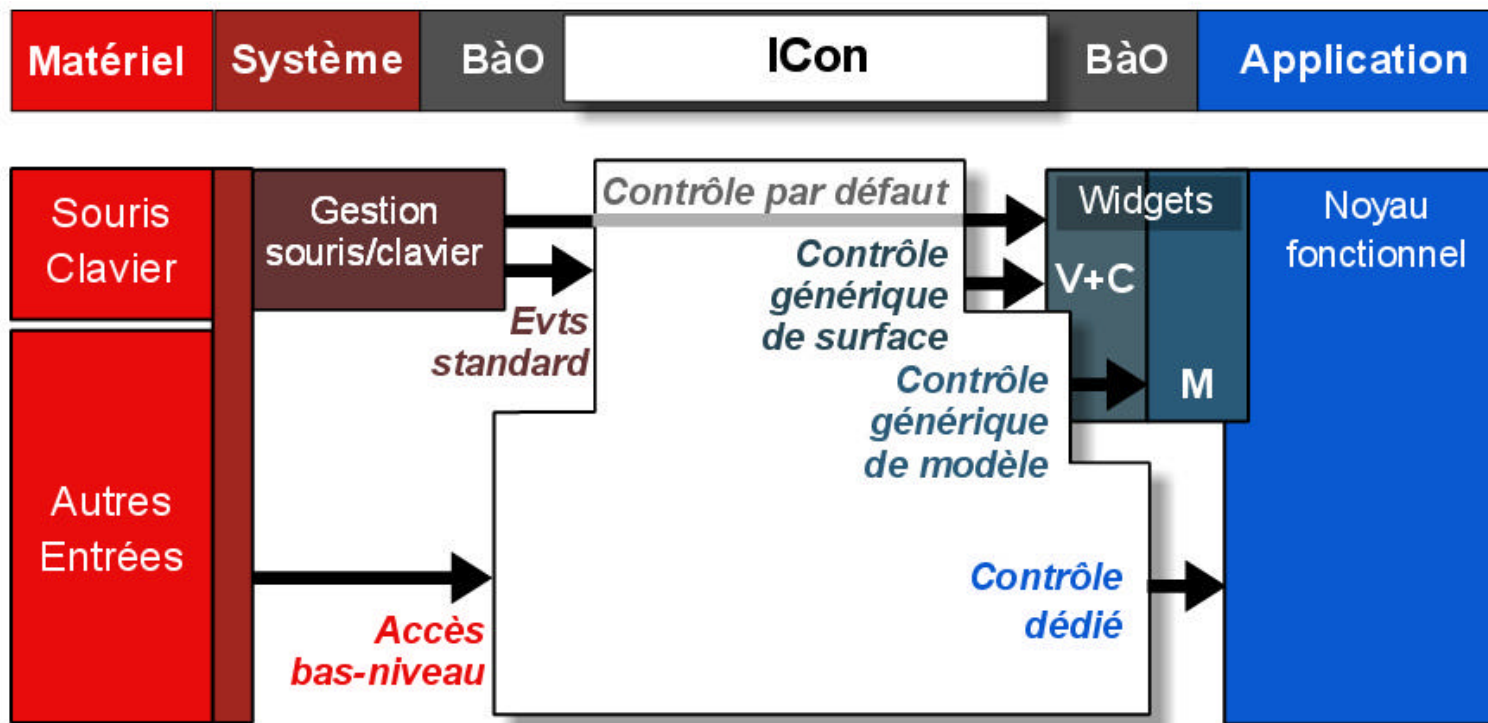
- Tâches:
  - Sélection de configurations
  - Personnalisation superficielle
  - Modifications importantes
  - Prototypage
  - Extension de la bibliothèque d'ICON
  - Développement d'applications compatibles ICON
- Utilisateurs potentiels:
  - Tout utilisateur
  - Utilisateurs avancés
  - Informaticiens
  - Concepteurs d'IHM
  - Programmeurs
  - Programmeurs

® **Utilisabilité du configurateur ?**

® **Utilisabilité des configurations d'entrée ?**

# Apports

- Extension d'un système interactif avec ICon





# Apports

---

- Intérêt d'un modèle réactif
  - Systèmes **conversationnels**:
    - *L'ordinateur est maître de l'interaction*, le client attend d'être servi
    - Ex: bases de données
  - Systèmes **réactifs**:
    - *L'environnement est maître de l'interaction*, l'ordinateur réagit de façon continue à ses changements [Berry 99]
    - Ex: systèmes embarqués
  - Systèmes **conversationnels + réactifs**:
    - Avion: communication avec le sol + pilotage
    - Système interactif: gestion événementielle + pilotes de dispositifs

→ Décrire la plus grande partie de l'interaction en entrée de façon réactive



# Apports

---

- L'adaptabilité en entrée avec ICoN

## **Contrôlabilité:**

- Dispositifs multiples
- Interaction fortement concurrente
- Techniques Post-WIMP prédéfinies
- + *Encourage l'expérimentation de techniques inédites*

## **Accessibilité:**

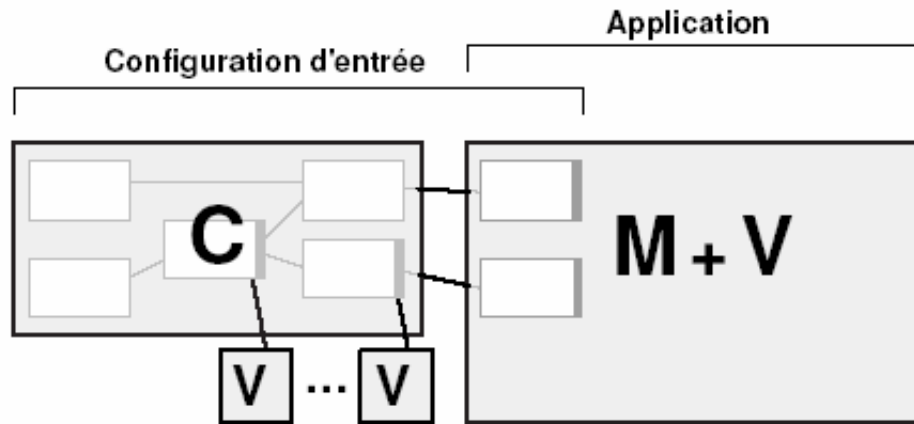
- Adaptateurs d'accessibilité prédéfinis
- + *Facilite la prise en charge d'un grand nombre de situations d'entrées appauvries*

## **Configurabilité:**

- Configurations personnalisables à différents niveaux

# Positionnement de notre approche

- Modèles d'interface: détaille la partie « entrées »



- Approches à événements: complémentaire
- Approches orientées contrôle: complémentaire
- Travaux connexes: Éditeurs visuels à flots de données  
Outils Post-WIMP 2D + outils 3D

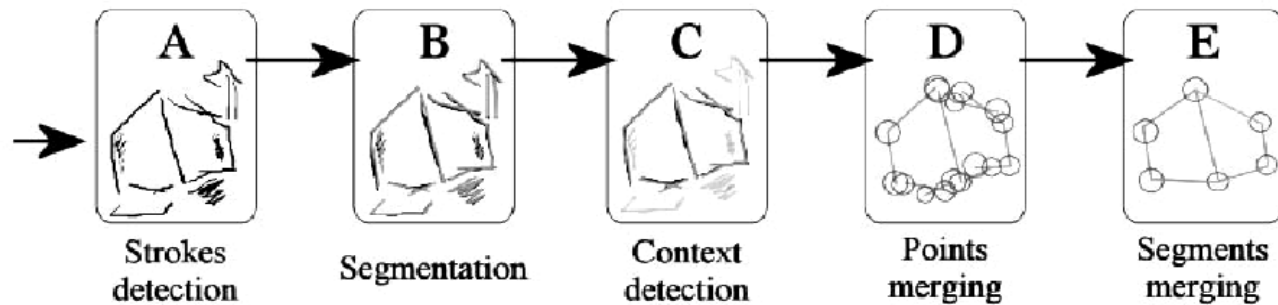
# Projets utilisant ICon

- **Marina II**

Reconstruction de scènes 3D à partir de photographies

- **Svalabard** [Huot et al 2003]

Interface Post-WIMP de modélisation 3D par croquis



- Évaluation de techniques Post-WIMP [Appert et al 2003]

- Boîtes à outils graphiques:

- **PiccoloICon**: intégration d'ICon à Piccolo [Bederson 2003]

- **MagLite**: boîte à outils Post-WIMP reposant sur ICon





# Conclusion et perspectives

---

## ■ Contributions

- Introduction de l'**adaptabilité en entrée**
- Proposition d'une **approche** originale
- Preuve de faisabilité avec **ICON**

## ■ Perspectives

- Validation exhaustive:
  - Tester sur plus d'utilisateurs
  - Tester sur plus d'applications
- Intégration à d'autres modèles et outils
- IConLite
- Explorer d'autres techniques Post-WIMP