

Wearable Computer

- Jean-Yves Tigli,
- Florent Grillon, Daniel Cheung
- Laboratoire I3S, MAINLINE
- Nice - Sophia Antipolis
- *tigli@essi.fr*



GDR I3 – GT « Mobilité Ubiquité », Paris Mai 2003
J.-Y. Tigli, F. Grillon, D. Cheung, email: tigli@essi.fr



Exposé en trois parties

- Introduction au « Wearable Computer »
- Wcomp 1.0 une plate-forme de prototypage rapide pour Wearable Computer
- Limites et Perspectives : Vers une architecture orientée comportement
Wcomp 2.0



Première Partie

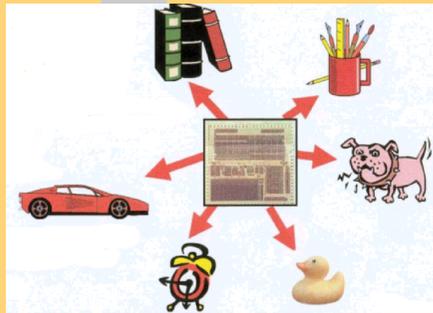
- Introduction au Wearable Computer



Ordinateur



Ordinateur et ubiquité



Ubiquité
(environnement équipé)

Ordinateur et mobilité



Pas
d'environnement



Mobilité
(utilisateur
équipé)

Ordinateur et mobilité



Pas
d'environnement



Simple
Changement de
localisation

Mobilité
(utilisateur
équipé)

Ordinateur et mobilité



Pas
d'environnement



Simple
Changement de
localisation



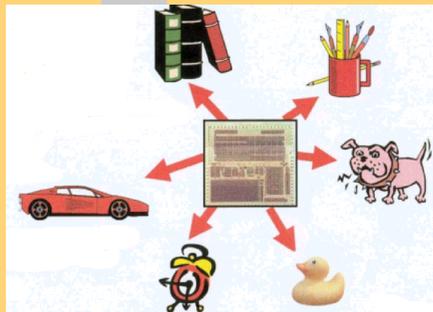
Environnement
Variable et Réel

Mobilité
(utilisateur
équipé)

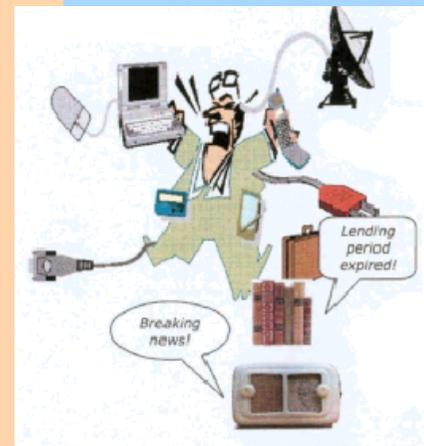
Ordinateur, ubiquité et mobilité



Mobilité
(utilisateur
équipé)



Ubiquité
(environnement équipé)



Wearable Computer

- Le Wearable Computer n 'est pas seulement le vêtement communicant
- Le Wearable computer n 'est pas seulement le téléphone ou le PDA
- Le Wearable Computer n 'est pas seulement le vêtement intelligent
- Car il s 'agit souvent de systèmes Ad-Hoc, non ouverts, non flexibles ET/OU

limités



Exemple : Application militaire

(Cf. D. Donsez)

- USA : projet Land Warrior
 - 600 M de dollars
 - 2003 → commando
 - 2008 → tous les fantassins
- France : projet Felin (Fantassin à équipement et liaisons intégrés)
 - 2005 → première version
 - 2015 → version finale

Application militaire



- Réduire les risque
- Corriger les déficiences du soldat
- Augmenter la connaissance du terrain
- Identification amis/ennemis



Casque



Renseignement

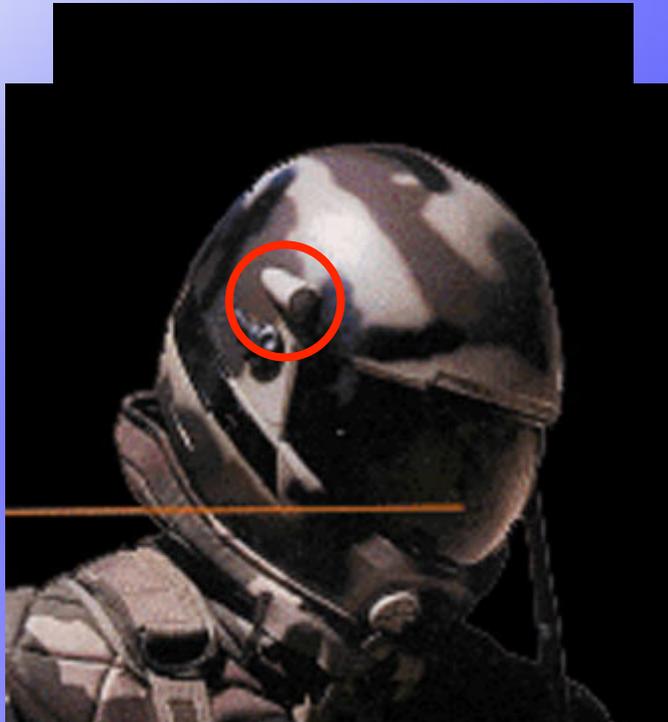
Famas



Combinaison



Le casque

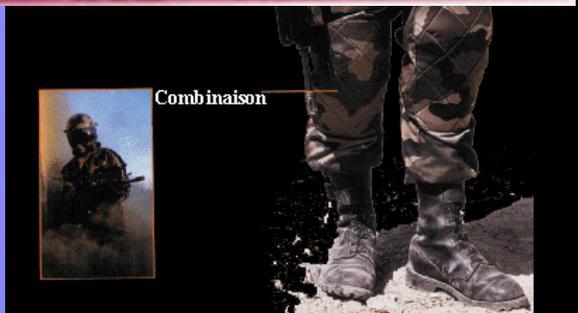


- Vision nocturne
- Évaluation des distances
- Dispositif allier
- Positions ennemis
- Outils de navigation
- État physique

Le renseignement



- Carte
- Repérage GPS
- Envoi de renseignements



Le Famas



- Conduite de tir
- Système de saisie
- Capture d'images
- Laser de visée/
verrouillage
- Laser d'identification

La combinaison



- UC
- Capteurs
- Diagnostic médical
- Climatisée
- NBC
- Furtive

Wearable Computer qu 'est-ce ?

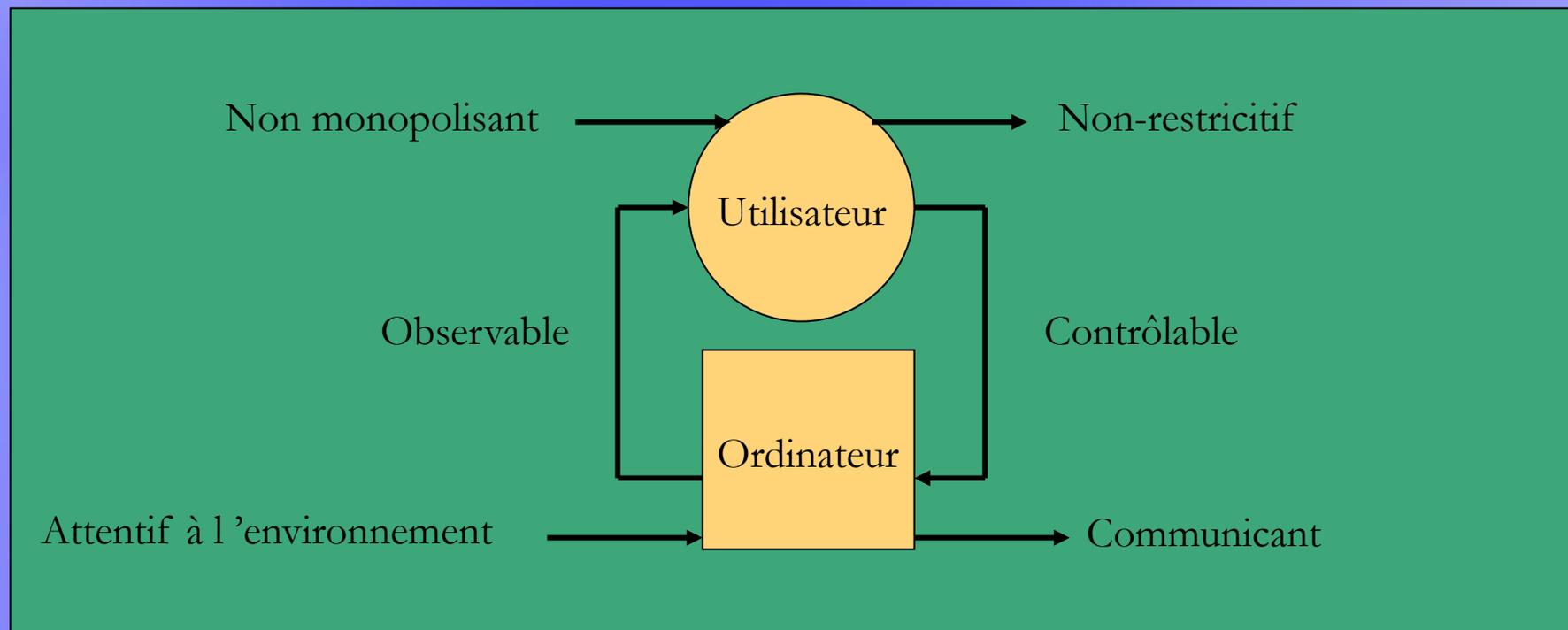
- Objectif du Wearable Computer :
 - Minimiser l 'encombrement, la redondance
 - Améliorer la connectivité, les services
 - Réduire les coûts de développement

En conclusion : un wearable computer, c'est quoi ?

- "Un ordinateur devrait être "porté" sur soi, interagir avec l'utilisateur en continu selon le contexte et agir en tant qu'assistant à diverses tâches » (*Thad Starner MIT/GeorgiaTech*)
- ~~Terminologie~~
 - ~~Ordinateur Vestimentaire~~

Comment le décrire alors ?

Les six caractéristiques du Wearable computer pour (Steve Mann)



Comment le décrire alors ? : Les six caractéristiques du Wearable computer pour (Steve Mann)

- La non-monopolisation de l'attention de l'utilisateur
 - Ne coupe pas l'utilisateur du monde extérieur
 - Lui permet d'avoir d'autres activités en utilisant le système
 - Garantit que l'utilisation de l'ordinateur sera une activité secondaire.
 - Fournit des informations perceptuelles supplémentaires
- Exemple : Réalité Augmentée

Les six caractéristiques du Wearable computer pour Steve Mann

- Non-restrictif pour l'utilisateur
 - Lui permet d'avoir d'autres activités en utilisant le système
- Exemple
 - IBM - Voice Systems
 - Dragon Systems – NaturrealySpeaking
 - Philips – Speech processing
 - Jabra - EarSet



Les six caractéristiques du Wearable computer pour Steve Mann

- **Observable par l'Utilisateur**
 - Les sorties du système sont visibles en permanence par l'utilisateur
- **MicroOptical**
 - Exemple : head mounted display
 - S'adapte sur une paire de lunettes neutre
 - Écran à cristaux liquides
 - Résolution : de 320*240 à 640*480
 - Poids : 7g



Les six caractéristiques du Wearable computer pour Steve Mann



- Contrôlable par l'utilisateur
- Exemple Twiddler 2
 - Pointeur: IBM Trackpoint
 - touche: 16
 - Sortie: PS2 souris et signal clavier
 - Poids: 165 g
 - Prix : \$199.00

Les six caractéristiques du Wearable computer pour Steve Mann

- Attentif à l'environnement
 - Ouvert sur l'environnement, multicapteurs.
- Exemple : localisation mais bien d'autres mécanismes de perception de l'environnement : capteurs:

Les six caractéristiques du Wearable computer pour Steve Mann

- Communicant avec les autres
 - Peut-être utilisé comme un medium de communication
- Exemple : Communications (ex. Mail, SMS...), Video Projecteur ..

Deuxième Partie

- Une première plate-forme de développement d'applications « wearable computer »
- WCOMP 1.0

Nos Premiers Objectifs :

- **Augmenter les capacités perceptuelles de l'ordinateur sur l'environnement**
- **Une plate-forme de prototypage rapide logiciel / matériel**
- **Plate-forme à coût réduit**
- **WCOMP 1.0 basée sur la Beanbox de SUN et le bus I2C sur IPAQ**



Augmenter les capacités perceptuelles de l'ordinateur sur l'environnement

- Le Wearable Computer doit pouvoir percevoir
 - l'état du système (ex. Niveau d'énergie)
 - l'état de l'utilisateur (ex. en mouvement, assis)
 - l'état de l'environnement local physique (ex. Niveau sonore)
 - les équipements de l'environnement local (ex. Présence d'un réseau cellulaire) (*)

*concerne l'ubiquité

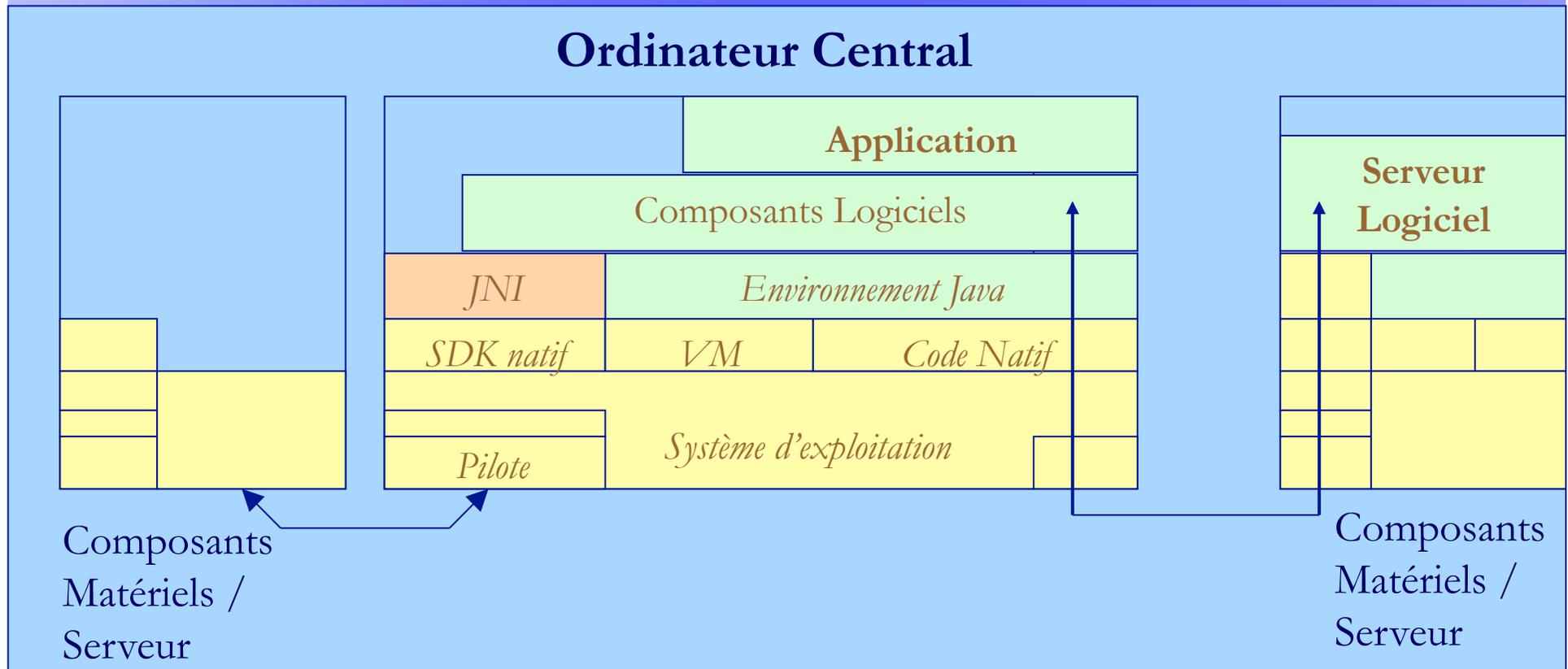
Notre approche :

Objectif : Une plate-forme matérielle/
logicielle ouverte et flexible

- Composants matériels Serveur / Client
BeanBox « Plug and use »
- Mutualisation des développements
- Le prototypage d'applications d'un
Wearable Computer facilité

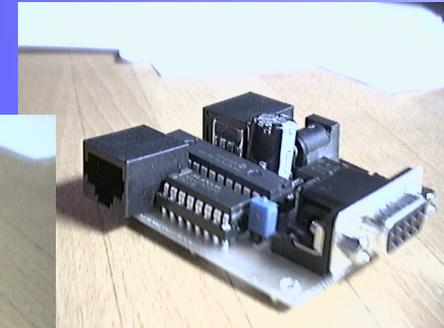
Composant matériel

« plug and use »



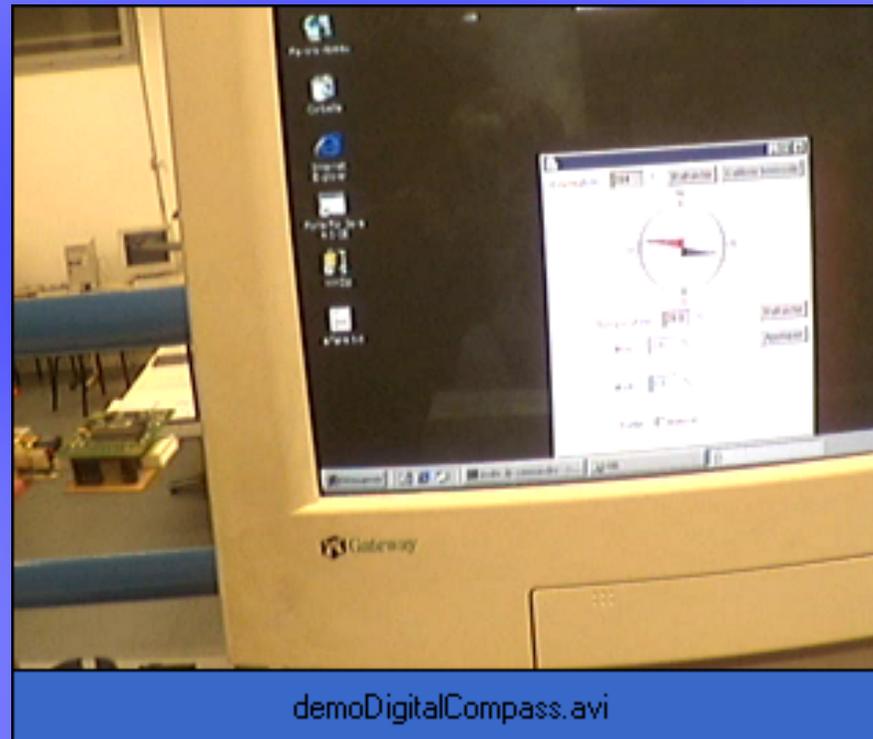
Exemple : Composants matériels I2C

- I2C générique « plug and use »
 - Télémètres Ultrasons
 - Boussole Numérique
 - Capteur de Lumière



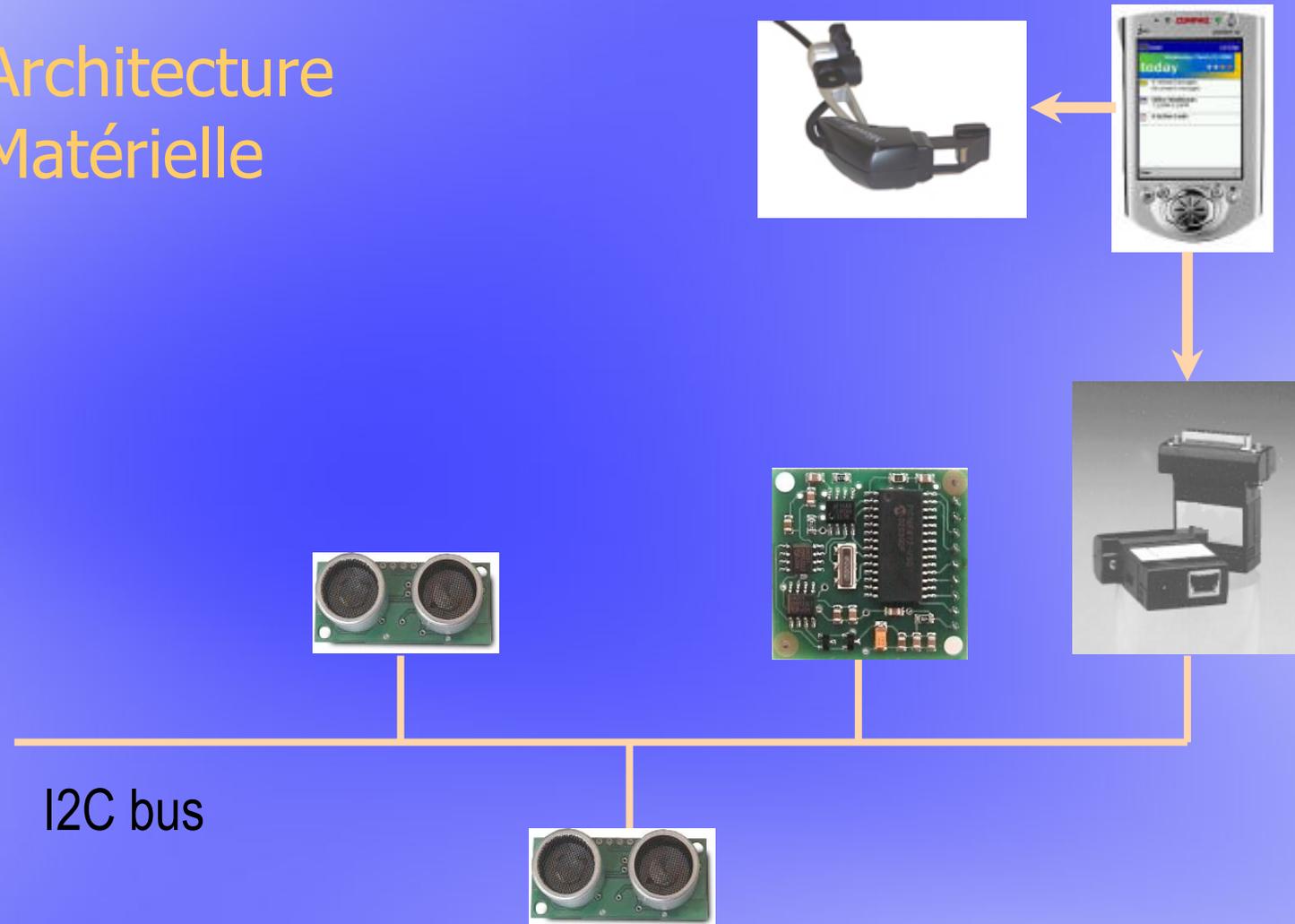
Exemple

- un capteur boussole numérique



Composants I2C sur IPAQ

- Architecture Matérielle

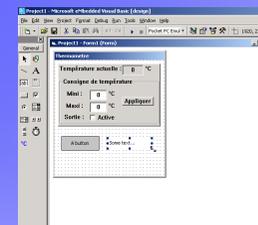
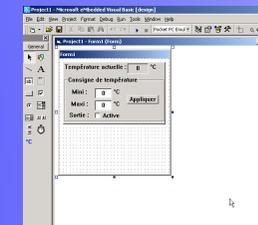
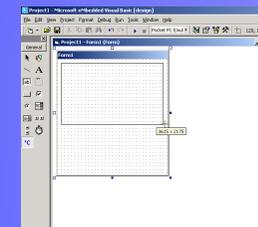
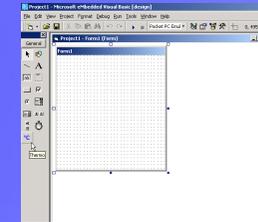


et donc Wearable Computer Wcomp

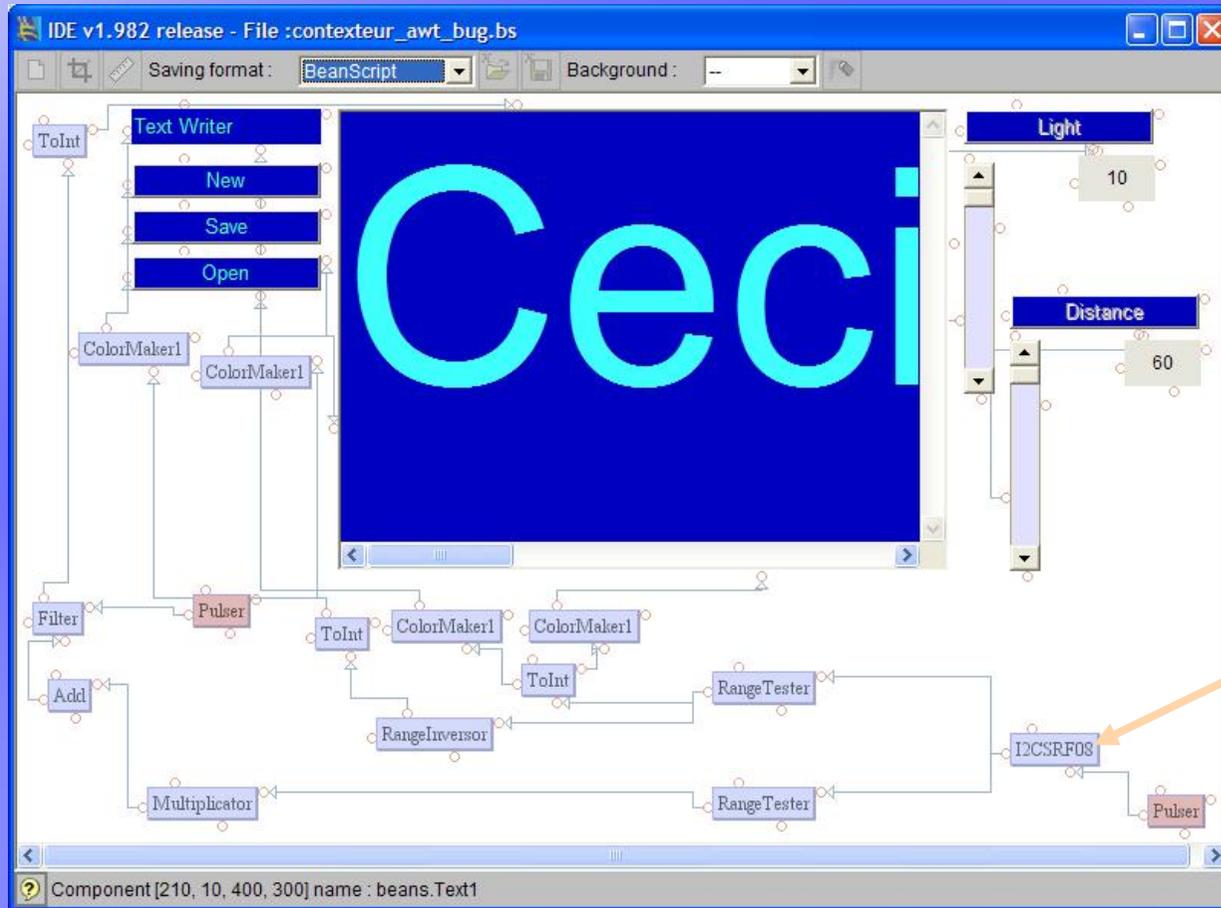


Une plate-forme de prototypage rapide logiciel / matériel

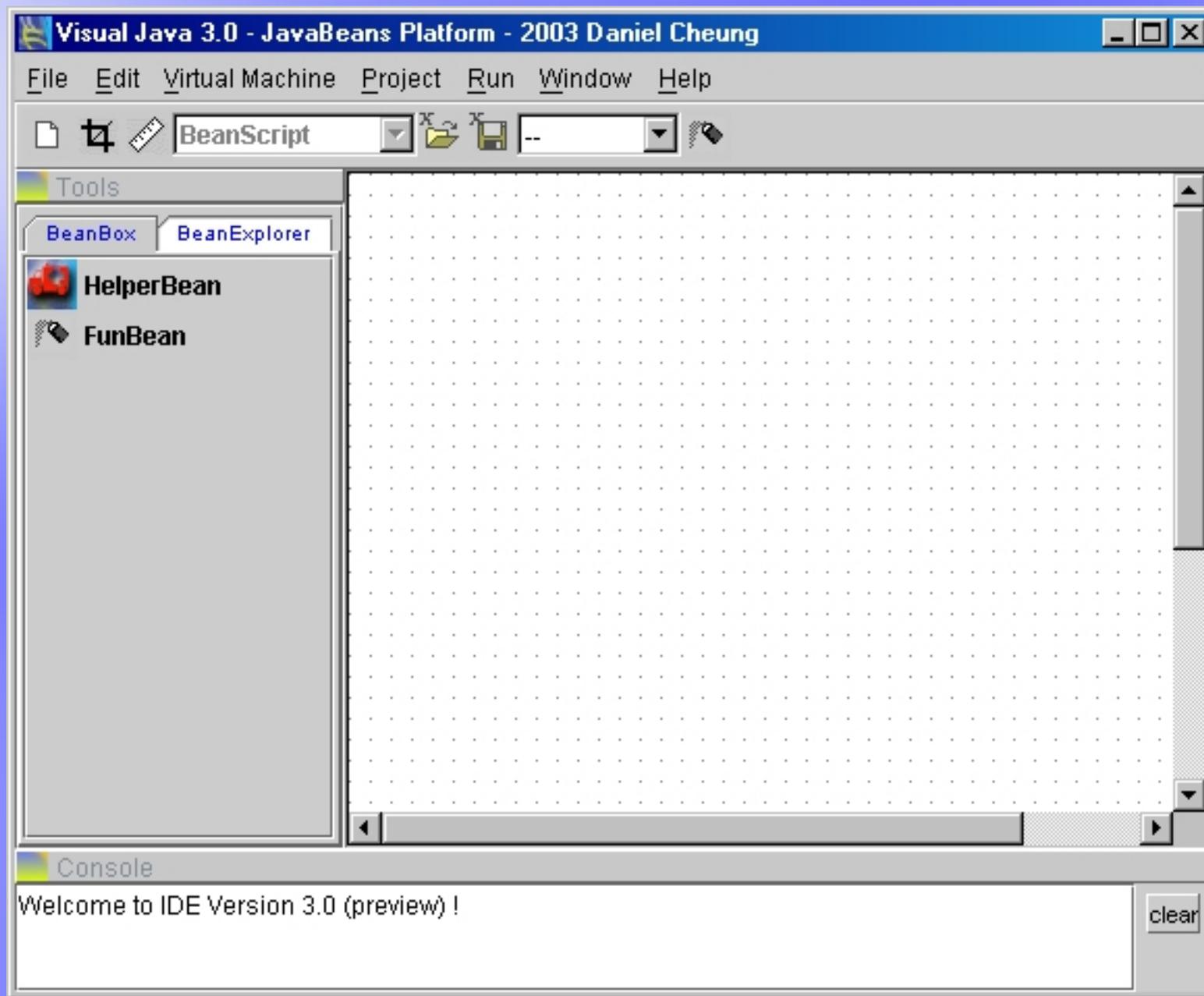
1. Assemblage de composants matériels
2. Conception de composants logiciels applicatifs
3. Assemblage et configuration des composants logiciels
4. Réalisation de code minimal de mise en œuvre de l'application



Environnement de développement Beanbox



« Tout est Bean »



Visual Java 3.0 - JavaBeans Platform - 2003 Daniel Cheung

File Edit Virtual Machine Project Run Window Help

BeanScript

Tools

BeanBox BeanExplorer

HelperBean
FunBean

R not altered
Q not altered
R1 not altered
Q1 not altered
R2 not altered
Q2 not altered

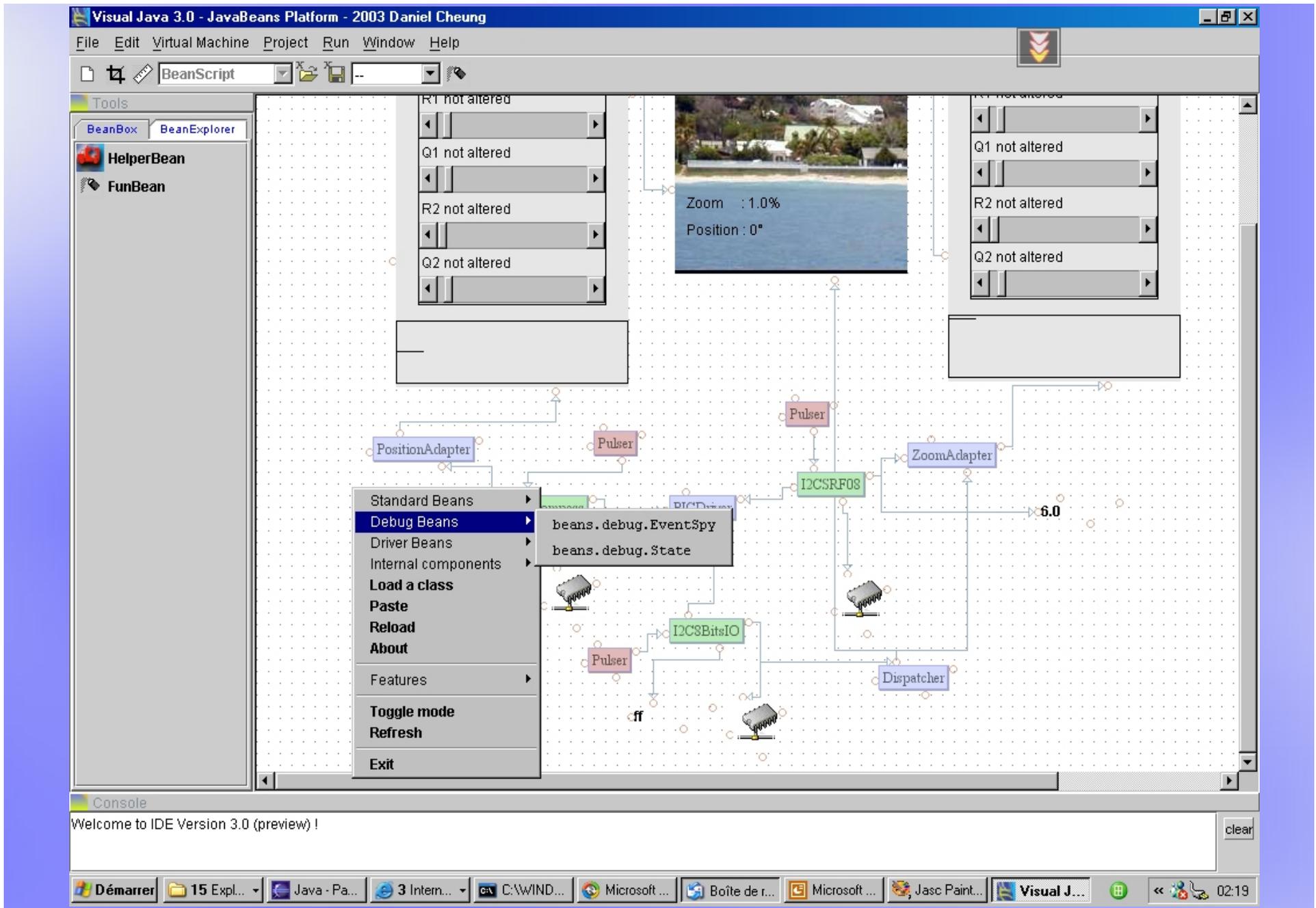
Zoom : 1.0%
Position : -47°

PositionAdapter
I2CCompass
PICDriver
I2CSRF03
ZoomAdapter
Pulser
6.0
1379

Console

Welcome to IDE Version 3.0 (preview)!

clear



Visual Java 3.0 - JavaBeans Platform - 2003 Daniel Cheung

File Edit Virtual Machine Project Run Window Help

BeanScript

Tools

BeanBox BeanExplorer

HelperBean
FunBean

1379

6.0

ff

Console

Welcome to IDE Version 3.0 (preview)!

clear

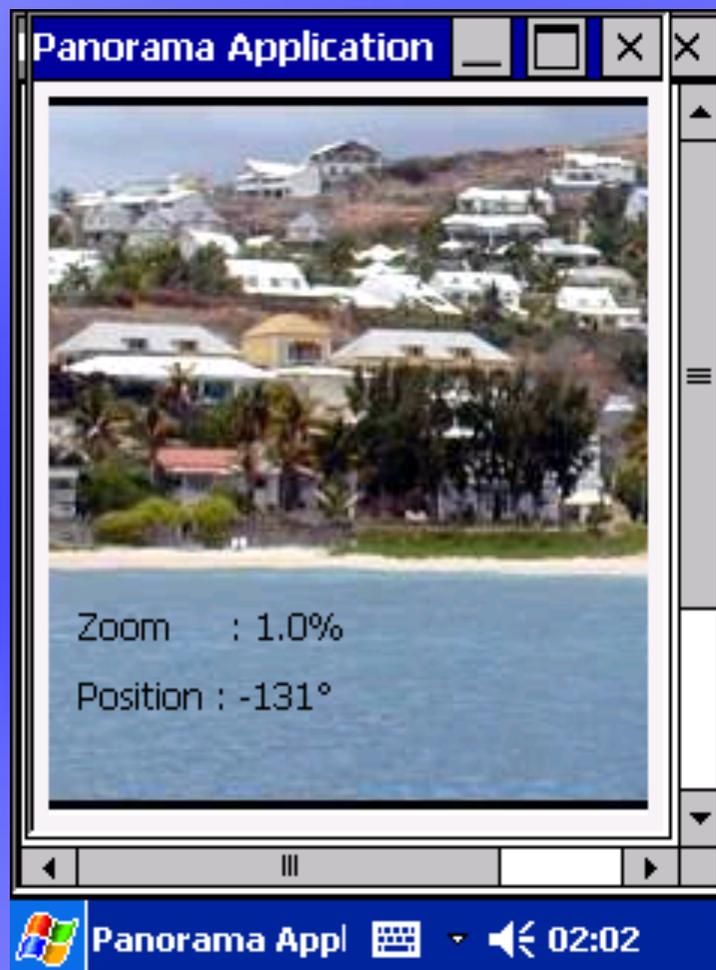
DEMO 1

- Deux threads pour l'affichage des valeurs de la boussole numérique et du proximètre

DEMO 2

- Un projet complet développé en moins de 4h chrono avec développements de Beans supplémentaires (ex. pour la visualisation d 'image panoramique)

Finalement sur PDA



Conclusion et Perspectives



Méthodologie satisfaisante pour le développement d'une chaîne de composants



Enfin une plate-forme ouverte diffusable pour prototypage rapide d'applications du « Wearable Computer »

Pour la faire vivre :

- **tester**
- **rajouter des beans logiciels**
- **rajouter des beans matériels/logiciels**

Limites

- **Travailler sur plusieurs chaînes de composants et sur l'Adaptation du Système est très coûteux**
- **Certaines familles d'applications même simples sont très sensibles aux modifications logicielles**

Architecture orientée comportement

- Objectif : Répondre à ces limites
- Une nouvelle proposition d'architecture logicielle pour l'ordinateur vestimentaire

Introduction : Style d'architecture

- Architectures dédiés à un domaine
- Partage du vocabulaire des composants et de ses connecteurs
- Ensemble de contraintes qui définissent des compositions architecturales.

- Shaw M., Garlan D. - Software Architecture: Perspectives on an Emerging Discipline - Upper Saddle River - NJ - Prentice Hall - 1996



Introduction : Composant logiciel

- Unité de composition
 - Les fonctionnalités basiques doivent permettre de les assembler entre eux afin de réaliser diverses fonctionnalités.
- Interfaces définies
 - Celles-ci peuvent être de plusieurs types : évènementielles, procédurales...
- Clemens Szyperski - Component Software : Beyond Object-Oriented Programming - ACM Press and Addison-Wesley, New York, 1998

Introduction : Composant logiciel spécialisé

- Besoins récurrents
- Solution spécifique à certains besoins
- Généralement associé à un style d'architecture
- Exemples : EJB, Composant Parallèle, Composant IHM
- J. Coutaz, L. Nigay - Chapitre 7 dans « Kolski C. - Analyse et conception de l'IHM - 2001 »
- Sun Microsystems, EJB Specification 2.1, 2002
- OMG, CORBA Components : Joint Revised Submission, 1999



Idée : Définir un composant logiciel spécialisé pour nos contraintes

- Interactivité avec l'utilisateur
- Adaptation au contexte

Mais aussi :

+ Réactivité à l'environnement

- Kortuem G. - Software Architecture and Wearable Computing - 1996



Vers un « composant comportemental » pour l'application

une entité applicative autonome
donc connecté aux entrées et aux sorties

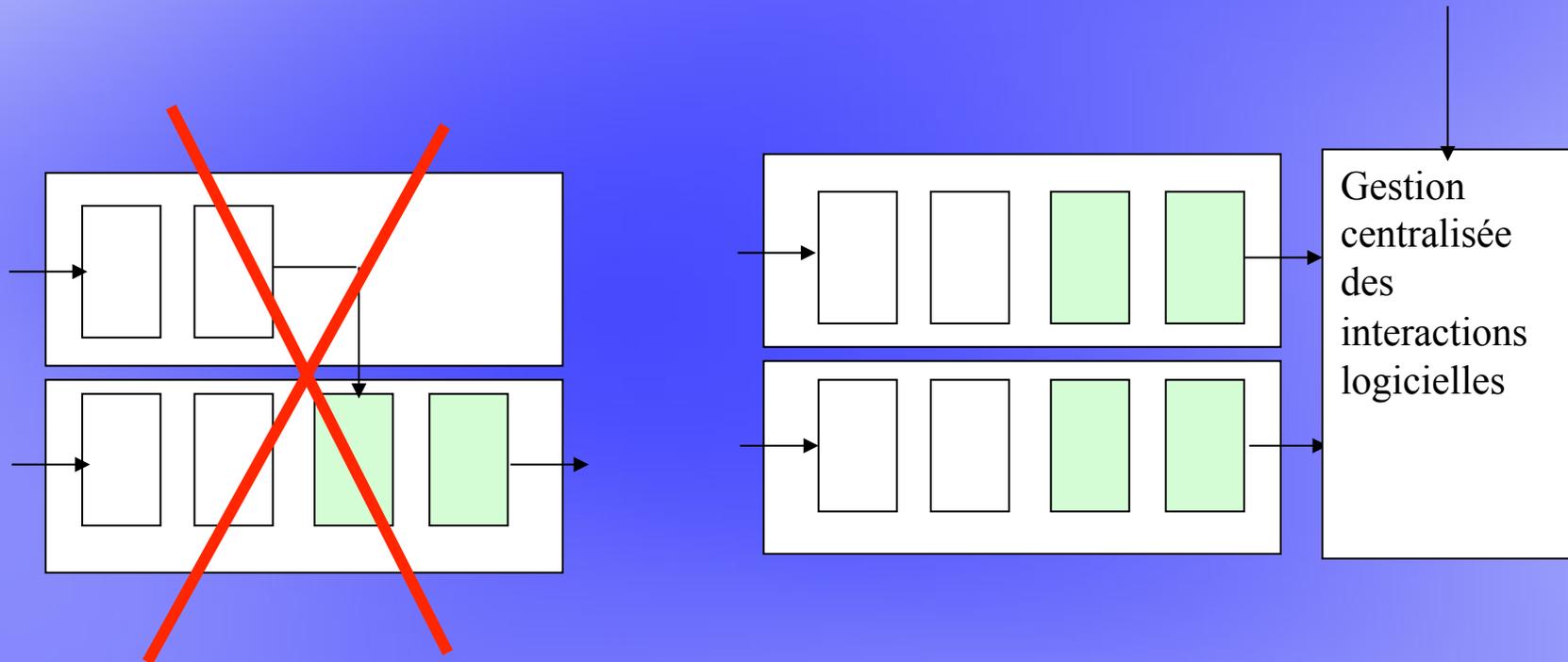


- adapté pour différer et centraliser les interactions entre assemblages de composants logiciels
- Unicité de la représentation
- Méthodes de validation

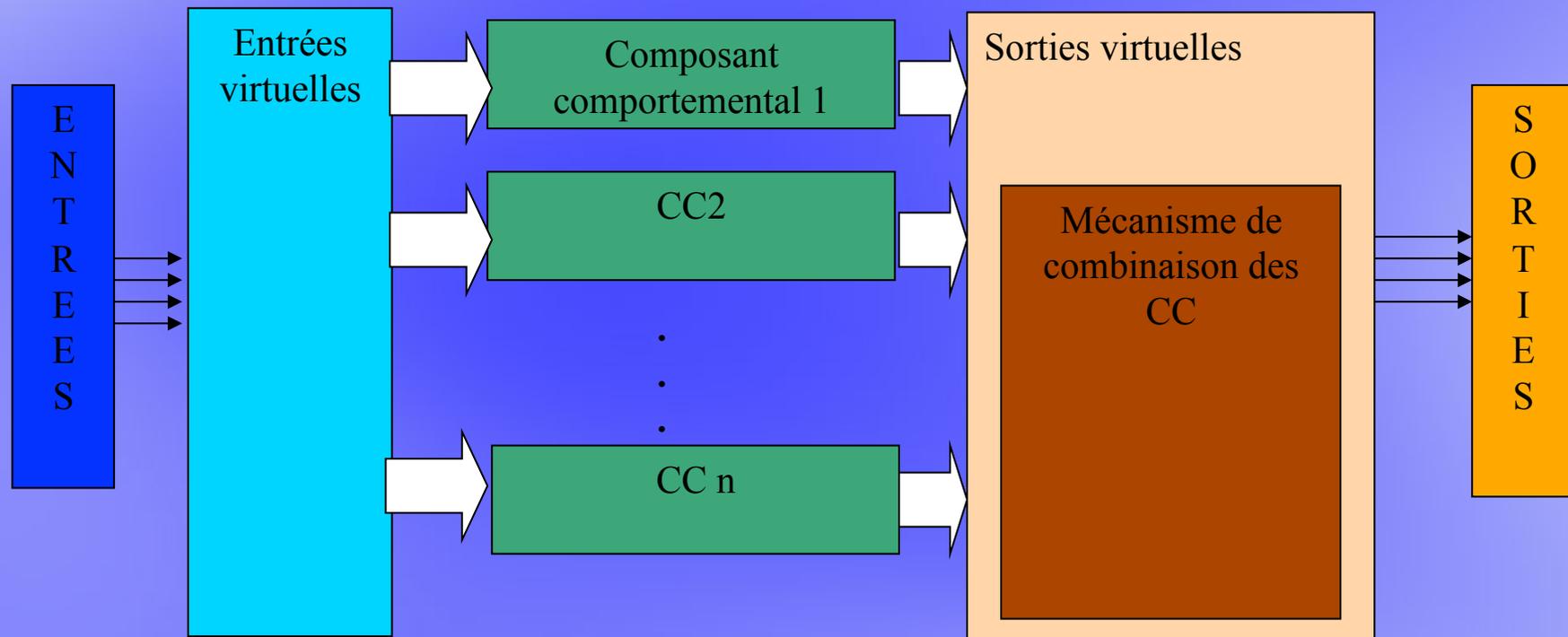
Comment gérer plusieurs composants comportementaux ?

- Duplication des entrées du système :
 - Fournit les entrées physiques aux CC
- Gérer les conflits sur les sorties :
 - Utilise un mécanisme de combinaison
- Mécanisme de combinaison des CCs
- Vers Wcomp 2.0

Ainsi ...



En résumé : Architecture orientée comportement



Différents mécanismes de combinaisons :

- Arbitrage à base de priorité
- Automate à états finis
- Fusion par pondération
- Fusion par algorithme de vote
- ...

Pirjanian P. - Behavior Coordination Mechanisms, State of the art - 1999

Arbitration à base de priorité :

- Résulte dans un ordonnancement temporel des comportements selon les priorités qui leur sont donnés. Typiquement, une tâche de forte priorité aura plus de slots de temps pour diffuser ses sorties qu'une autre de priorité plus faible.

Cette méthode est comparable à un ordonnanceur à temps partagé.

R. Brooks :
mécanisme de subsomption
Brooks, R.A., NJ, 1991

Automate à états finis :

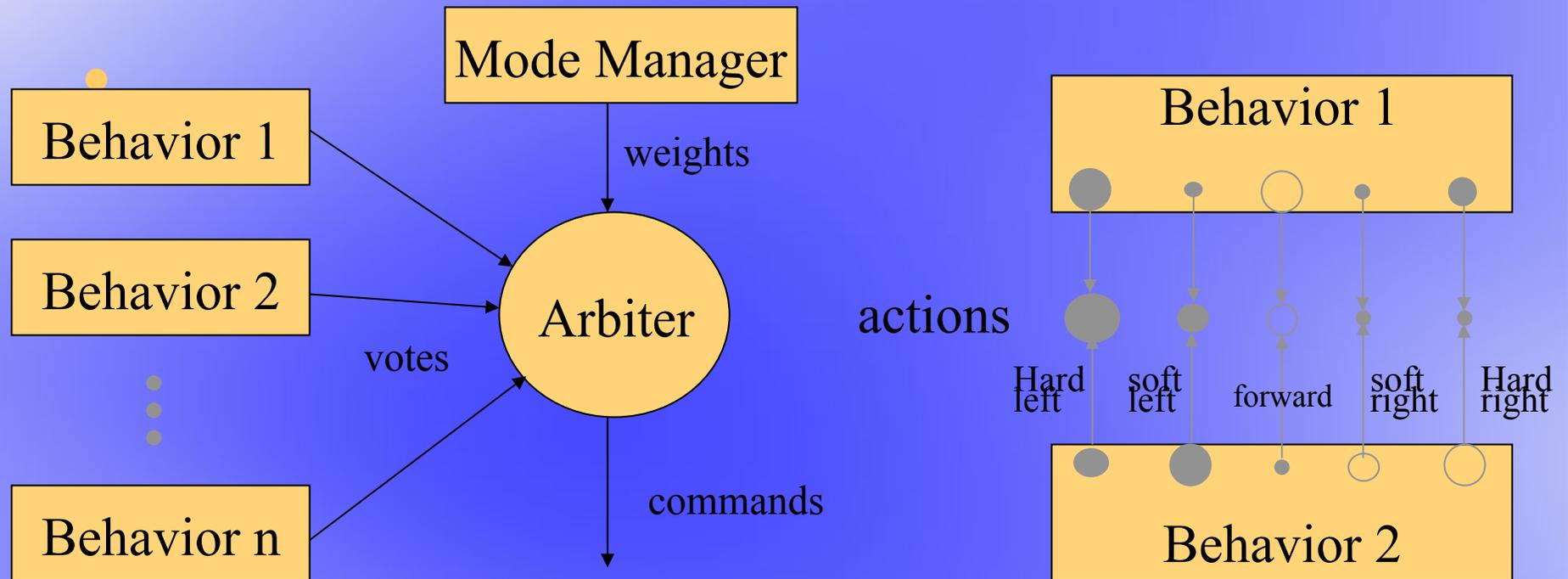
- Le principe des automates à états finis consiste à donner à un comportement la gestion complète des sorties jusqu'à ce qu'un changement d'état s'effectue. Les changements d'états peuvent être déclenchés soit via des évènements soit via une horloge.
- Temporel : Arkin, R.C., 1987
- Évènementiel : Kaelbling, L. P. and Rosenschein (1990).

Fusion par pondération :

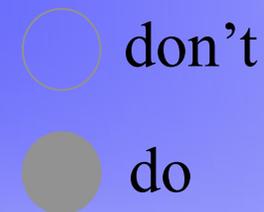
- Les sorties de chaque comportement sont fusionnées. En fait, les vecteurs de sorties sont sommés (avec des pondérations) puis la sortie est normalisée.

R. C. Arkin - 1987

Fusion par algorithme de vote:



- Rosenblatt J. K. - DAMN, 1991



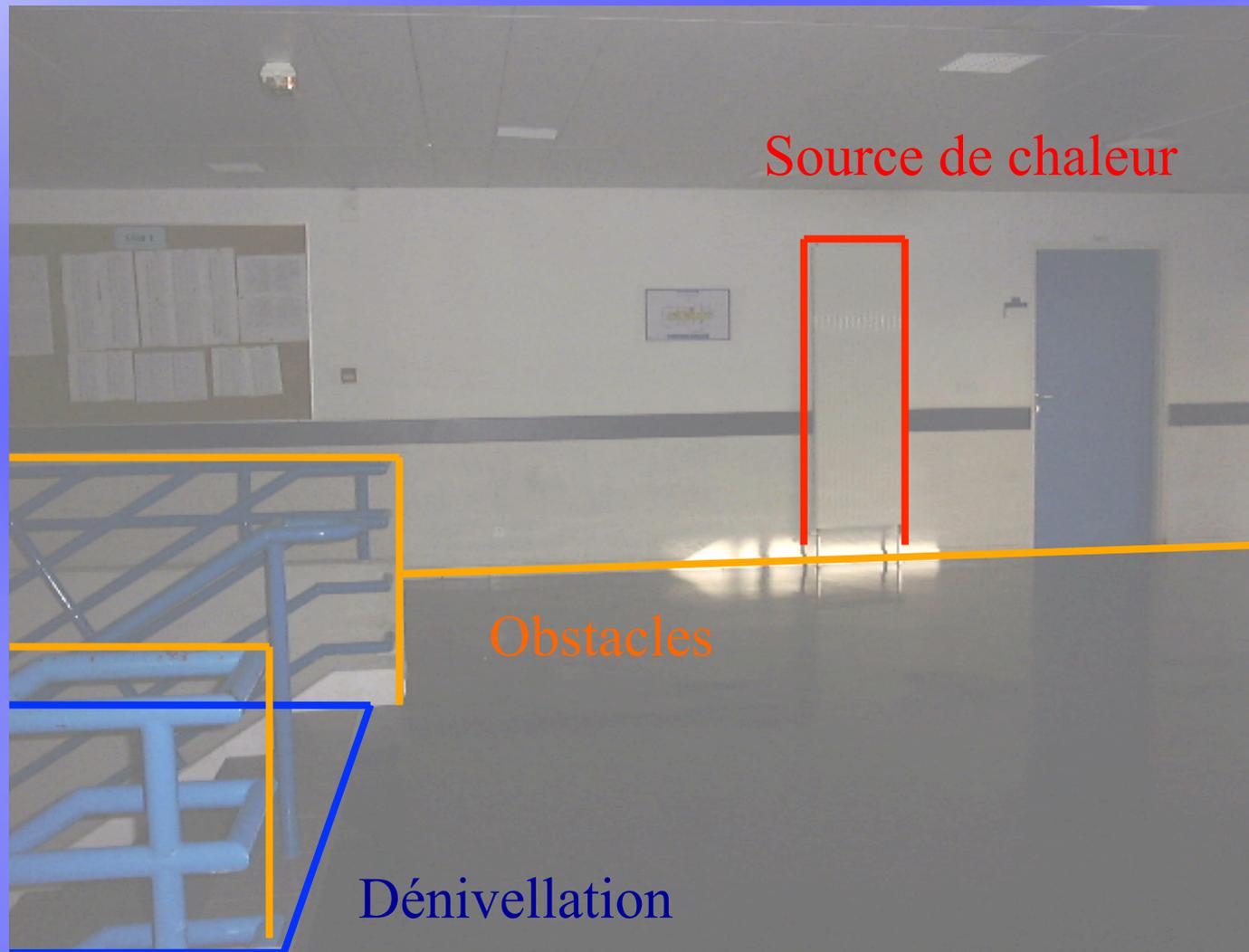


Exemple d'application

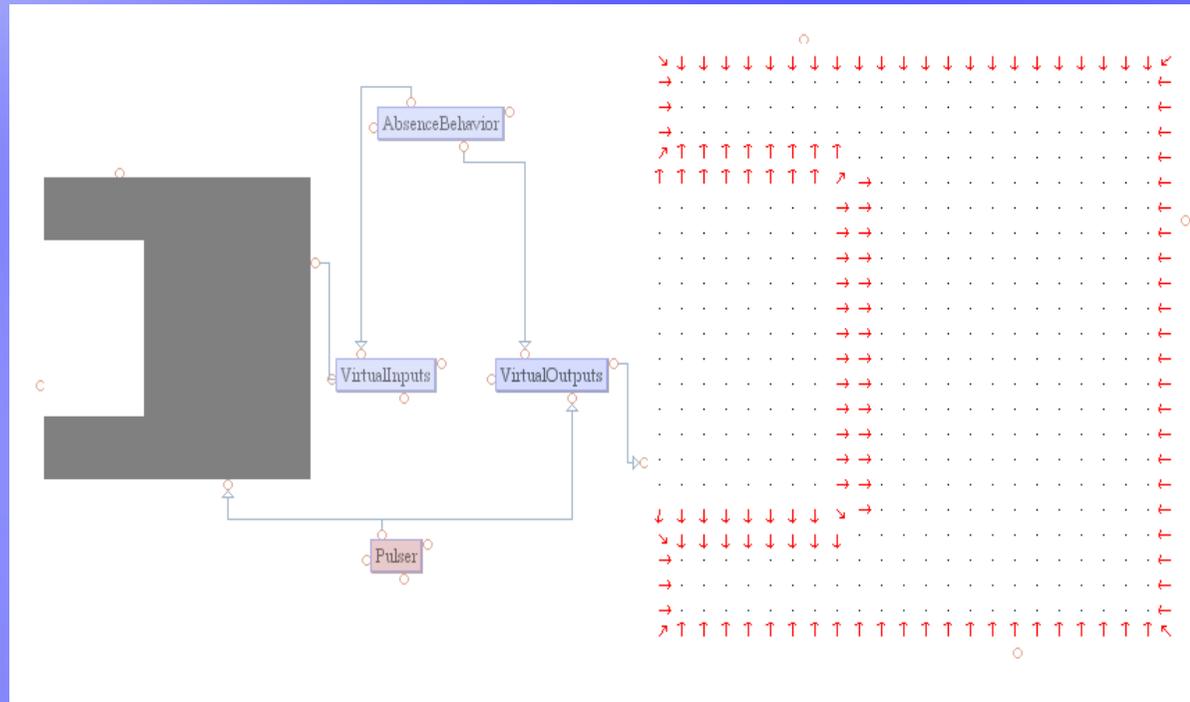
Wearable computer pour
assistance à la navigation d'un
déficient visuel



Exemple d'environnement



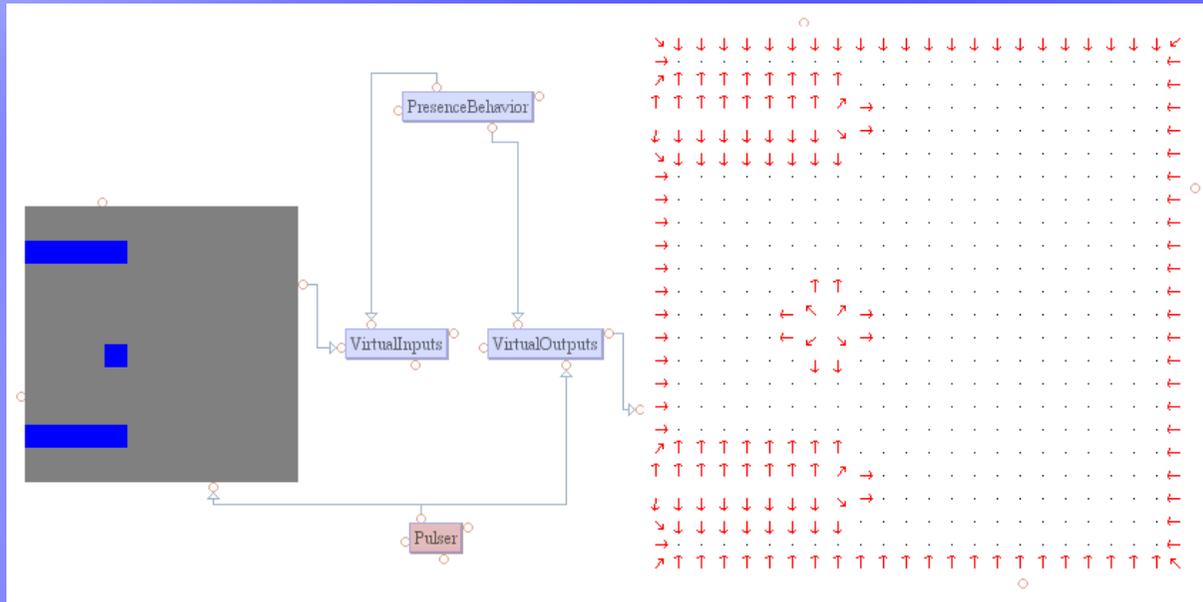
comportement d'évitement d'obstacle



- (4 capteurs : devant, derrière, gauche, droite) : d'où 9 positions réflexes

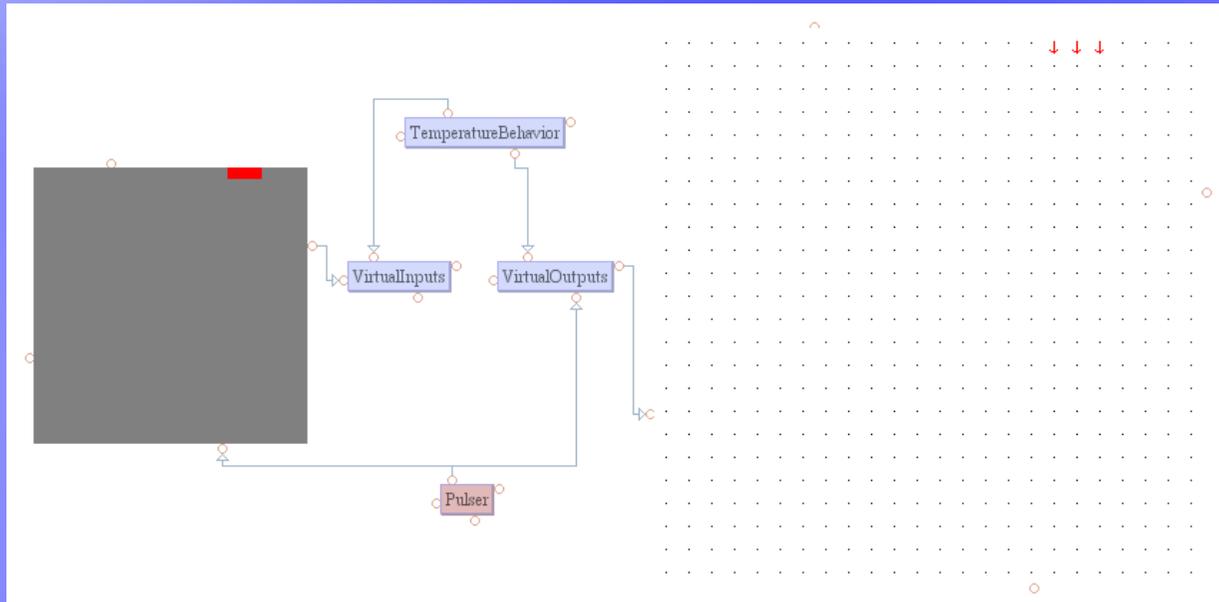
comportement d'évitement de dénivellation du sol

- (4 capteurs : devant, derrière, gauche, droite) d'où 9 positions réflexes

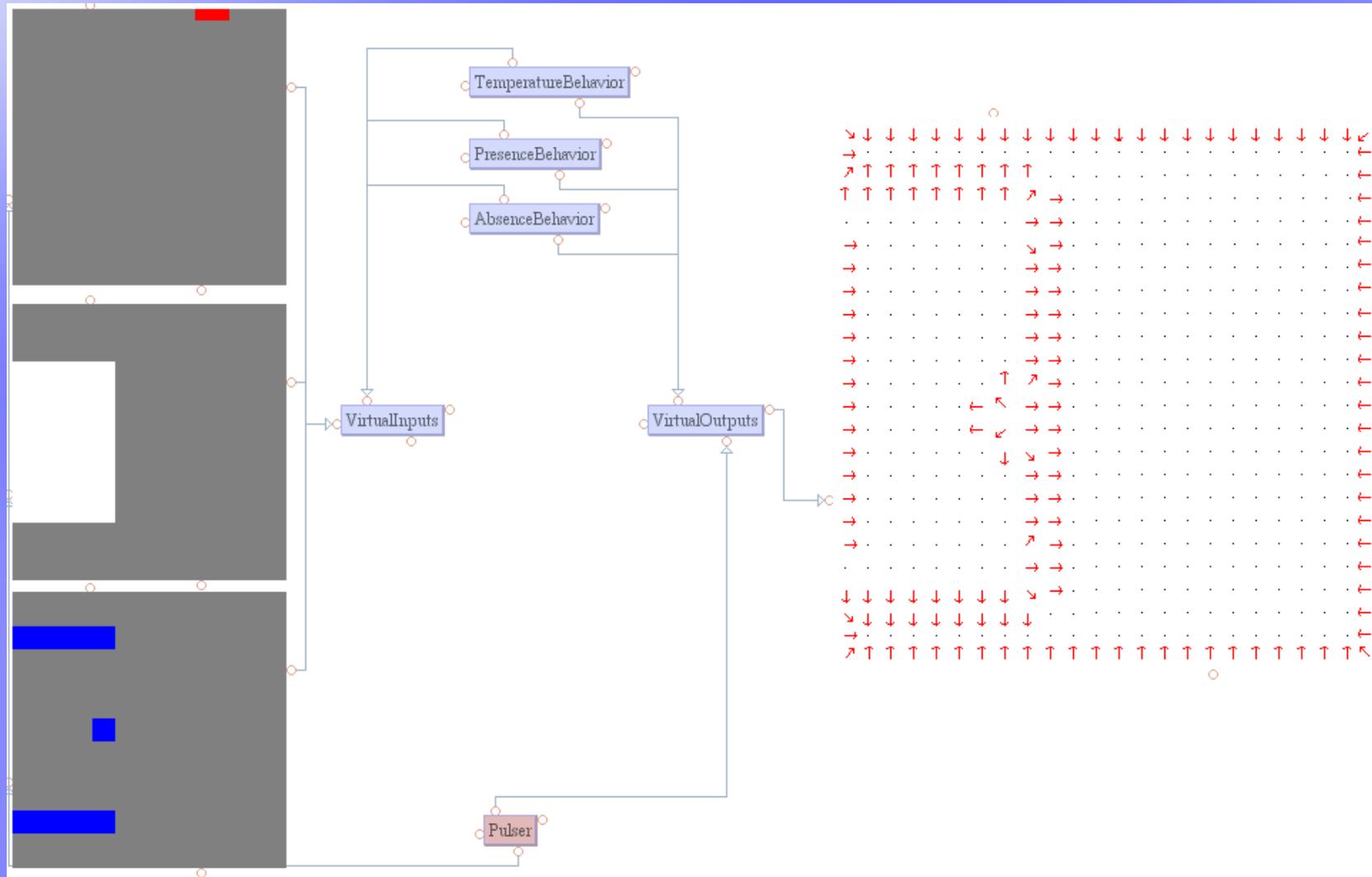


comportement d'évitement de chaleur

- (1 capteur : endroit où l'on se trouve) d'où une seule réaction : recul



Mécanisme de Combinaison : Fusion



Conclusion et Perspectives :

Mécanisme d'Arbitrage prouvable pour applications critiques

Collaboration avec le CMA de l'Ecole des Mines de Paris (V. Roy) et l'INRIA Sophia Antipolis (A. Ressouche) : Architecture Synchrone Asynchrone (SSA)

– Idées :

- Modèle Abstrait de l'environnement
- Équivalence avec un Automate à état fini
- Description avec un langage synchrone orienté contrôle
- Vérification formelle et Compositionnalité

→ Plus au niveau système



Conclusion et Perspectives :

Mécanismes de combinaisons mélangeant arbitrage et fusion, pour des applications non-critiques (DEA F. Grillon en cours)

- Définition de nouveaux composants
 - CC dans des Beans
 - MC dans des Beans
- Vers la plate-forme logicielle Wcomp 2.0

Journée « Déficients visuels et NTIC : présent et avenir »

- 5 juin 2003 – Sophia Antipolis

Organisation : ESSI, CNRS, INRIA,
AbsoluteVox, l'APEDV, la FNDV

<http://www.essi.fr/WAI>



GDR I3 – GT « Mobilité Ubiquité », Paris Mai 2003
J.-Y. Tigli, F. Grillon, D. Cheung, email: tigli@essi.fr

