



pointeurs multiples: étude et implémentation

eric lecolinet
get/enst + cnrs/lcpi





Motivations

- **Interaction multi-pointeurs**
 - variété de cas de figures
- **1e partie: classification**
 - notation UDP/C
 - mettre en évidence les ressemblances / dissemblances
 - guide pour l'implémentation
- **2e partie: implémentation**
 - 2 technologies complémentaires



Notation UDP/C

■ Quantités

- **U** = # Utilisateurs
- **D** = # Dispositifs d'affichage (*Displays*)
 - peut comprendre plusieurs écrans
- **P** = # Pointeurs
- **C** = # unités Centrales (*Computers*)

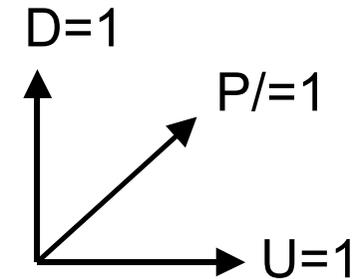
■ 3+1 Dimensions

- U
- D
- **P/** = P/D = # Pointeurs par Display
- C (optionnellement)

Cas standard

■ Situation habituelle

- 1 Utilisateur
- 1 Display (composé de 1 ou N écrans)
- 1 Pointeur ; 1P par D
- **UDP/ = 1-1-1**



■ Variantes

- application exécutée à distance
 - *X Window, VNC, RDP*
- plusieurs unité Centrales
- **UDP/C = 1-1-1-***

Interaction bi-manuelle

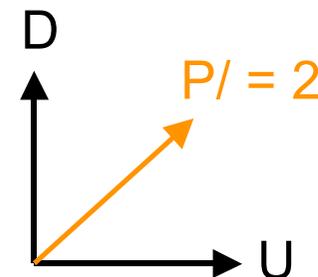
■ Deux pointeurs indépendants

- exemple: outils transparents
 - main « fourchette » non dominante : positionner l'outil
 - main « couteau » dominante : sélection/action

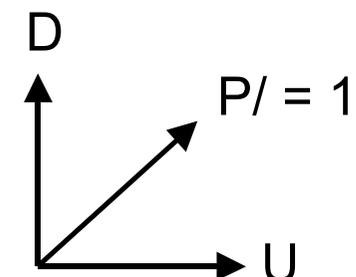


■ Notation UDP/

- 1 Utilisateur
- 1 Display
- 2 Pointeurs par Display
- **UDP/ = 1-1-2**



bi-manuel



standard

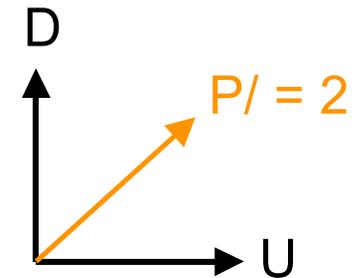
Interaction bi-manuelle (2)

■ Implémentation

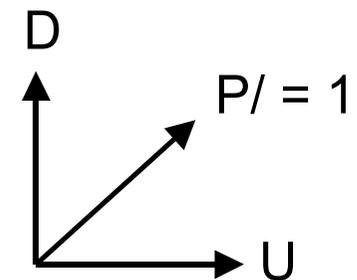
- séparation des flux événementiels
- pseudo-pointeur

■ Deux solutions

- niveau Application
 - pseudo-pointeur propre à l'appli
 - exple: *Icon*, *Jazz MID*
- niveau Display
 - pseudo-pointeur générique
 - « serveur multi-pointeurs »



bi-manuel



standard

Single display groupware

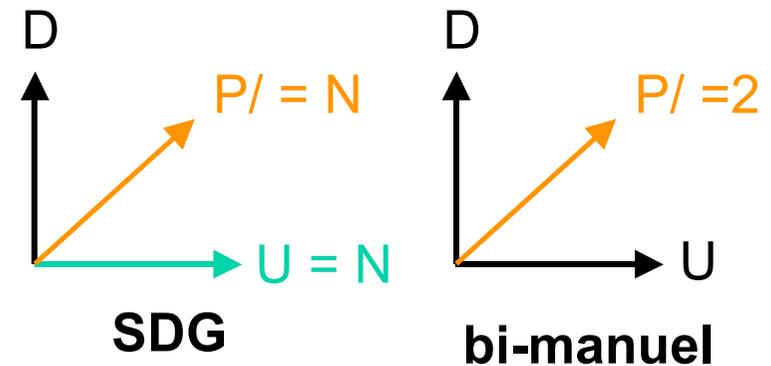
■ Partage d'un même dispositif par plusieurs utilisateurs

- N Utilisateurs (*en un même lieu*)
- 1 Display
- N Pointeurs par Display (1 par U)
- $UDP/ = N-1-N$



■ Comparaison

- rappel bi-manuel : $UDP/ = 1-1-2$
- $P/$: généralisation
- U : problème des modes, actions conflictuelles, etc.

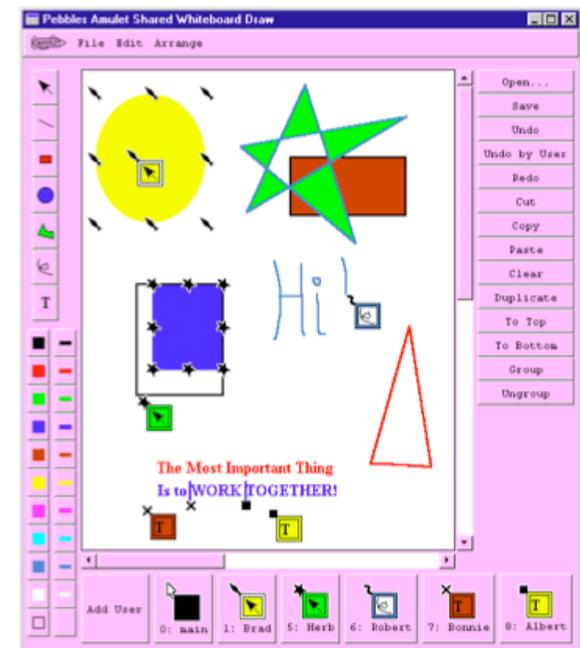
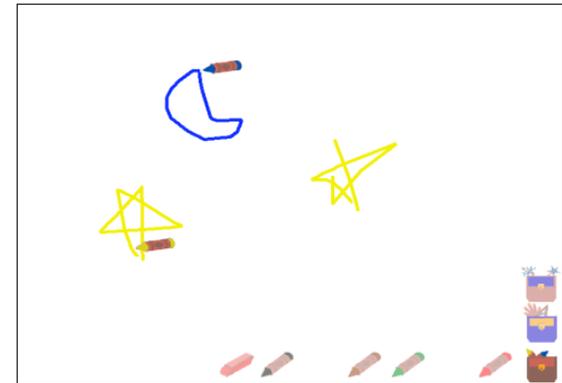


Single display groupware (2)

- **Modes et variables d'état**
 - dupliques ? choix et représentation ?

- **a) Eviter les modes**
 - exple: local tools (*KidPad / Jazz MID*)
 - intégration du mode dans l'interacteur
 - implémentation: ~ 1-1-N

- **b) Représenter les modes pour chaque U**
 - exple: *PebblesDraw (Amulet)*
 - lisibilité ?
 - impl.: complexité, toolkit spécifique ?



« Collaboration » multi-dispositifs

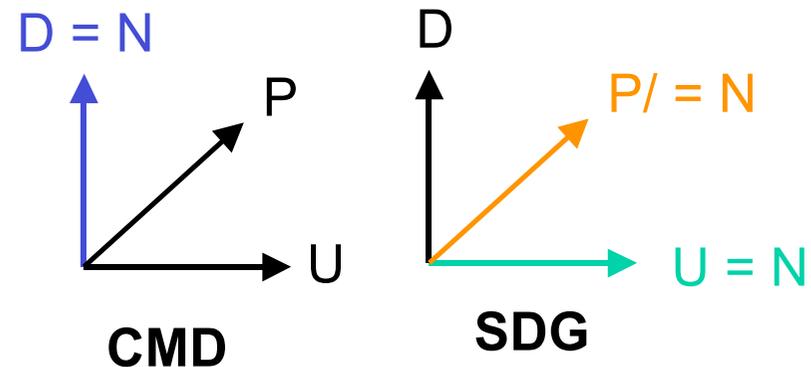
■ Contrôle de plusieurs dispositifs par un utilisateur (« Mighty Mouse »)

- ubiquité, mais pas de partage
 - 1 Utilisateur **actif**
 - N Displays (*en un même lieu*)
 - 1 Pointeur
- **UDP/ = 1-N-1**



■ Cas inverse du SDG

- rappel SDG: $UDP/ = N-1-N$
- impl. radicalement différente :
 - applications standard

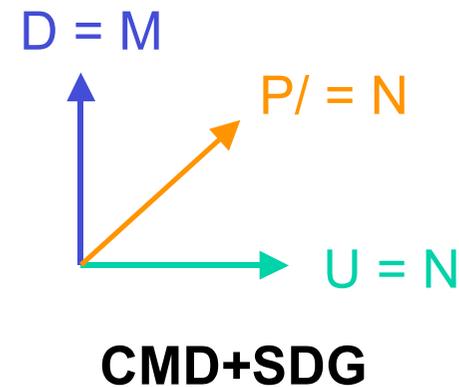


« Collaboration » multi-dispositifs (2)

■ Extensions

- **plusieurs** Utilisateurs actifs mais **pas** sur le même Display
 - exple: *PointRight (iRooms)*
 - **UDP/ = N-N-1**
 - implementation: ~ 1-N-1

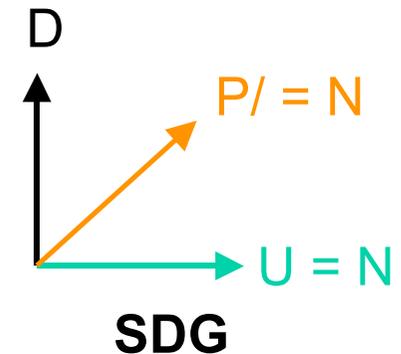
- synthèse avec SDG
 - **UDP/ = N-M-N**



Logiciels collaboratifs synchrones

■ Partage d'un même document sur plusieurs dispositifs

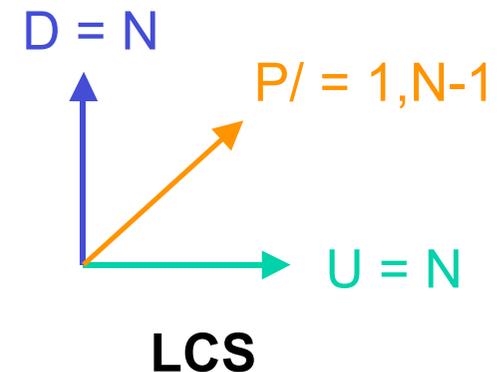
- partage, comme SDG
- pas en un même lieu
 - plusieurs vues (pas forcément identiques)
 - représentation de l'activité



■ Télépointeurs vs. multi-pointeurs

- multi-pointeurs: E/S
- télépointeurs: S seulement (réplication)

■ $UDP/ = N-N-(1,N-1)$





Synthèse

Cas	Notation UDP/C
a) cas usuel	1-1-1 (exéc. distante: 1-1-1-*)
b) interaction bi-manuelle	1-1-2
c) single display groupware	N-1-N (exéc. distante: N-1-N-*)
d) contrôle multi-dispositif	1-N-1 (mighty mouse) N-N-1 (point right)
c+d) SDG + CMD	N-N-N
é) collecticiels synchrones	N-N-(1,N-1)

■ Cohérence, complétude, extensions ?

- nature des écrans, adaptation, entrées clavier...

■ Efficacité ?

- guide pour l'implémentation

Implémentation

■ **Serveur multi-pointeurs UMS**

- extension d'un serveur X
- compatible avec applis standard

■ **Sources et flux d'événements**

- **source**: dispositif physique d'interaction
 - souris série, MIMIO
 - modèle extensible
- **flux**: « canal » événementiel logique
 - envoie des événements X modifiés
 - pseudo-flux 0 = flux natif X
- nombre et correspondance arbitraire

Serveur multi-pointeurs

■ Gestion générique uniforme

- de toutes les sources
- de tous les pseudo-pointeurs au niveau du Display

■ Une application

- « compatible UMS » :
 - peut identifier **tous** les flux (y compris le flux natif X)
- « standard » :
 - intègre **tous** les flux (y compris ceux des pseudo-pointeurs)

■ Optionnellement

- événements étendus

Bi-manuel et single display groupware

Cas	Notation UDP/C
b) interaction bi-manuelle	1-1-2
c) single display groupware	N-1-N (exéc. distante: N-1-N-*)

■ Même traitement

- UMS : gère 2 ou N flux (et pointeurs)
- applications “**compatibles**” (séparation de flux)

■ Cas simplifié

- SDG sur **applications** distinctes
- applications **standard**

“Collaboration” multi-dispositifs

d) contrôle multi-dispositif	1-N-1 (mighty mouse) N-N-1 (point right)
c+d) point-right généralisé	N-N-N

- **Contrôle à distance des serveurs UMS**
 - applications **standard**
- **Topologie de Displays**
 - chaînage de serveurs UMS
 - “espace virtuel d’affichage”
- **Cas général CMD + SDG**
 - applications “**compatibles**” (séparation de flux)

Logiciels collaboratifs synchrones

e) collecticiels synchrones

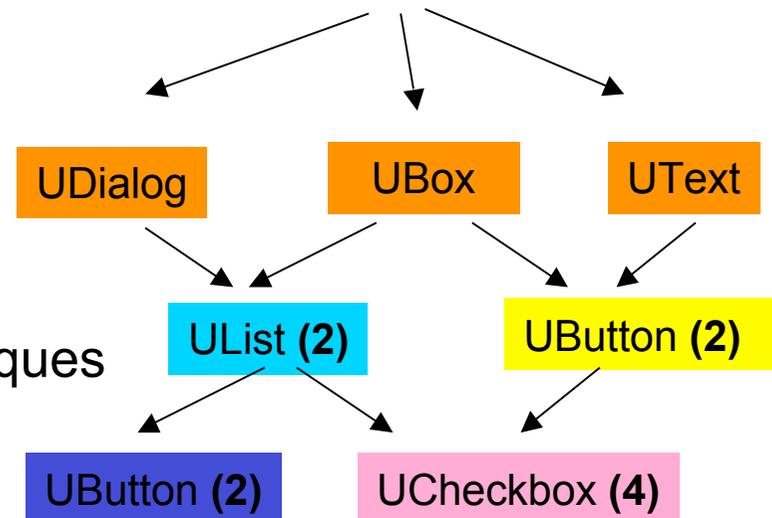
N-N-[1,N-1]

■ Besoins

- vues synchrones
- télépointeurs

■ Toolkit graphique Ubit

- architecture “moléculaire”
 - *bricks* = objets élémentaires
 - *brickgets* = combinaisons de briques
- objets tous partageables
 - synchronisation implicite
 - brickgets -> réplication graphique



Toolkit Ubit

■ Replication multi-displays

- 1 brickget --> N vues sur différents Displays
- pas de restriction sur le niveau de partage
 - brick, brickget, arbre, DAG
- réplication logique
- télépointeurs logiques



■ Séparation des flux événementiels

- compatibilité UMS
- séparation logique interne





Conclusion

■ Notation UDP/C

- tentative de classification des cas d'interaction multi-pointeurs
- guide pour l'implémentation
- à enrichir, améliorer...

■ Implémentation

- couvre tous les cas principaux
- serveur multi-pointeurs **UMS**
 - quelques limitations
- toolkit **Ubit**
 - C++, X Window, Open GL
- logiciel libre; www.enst.fr/~elc --> **Ubit**

 **That's all folks**