

TIW 8

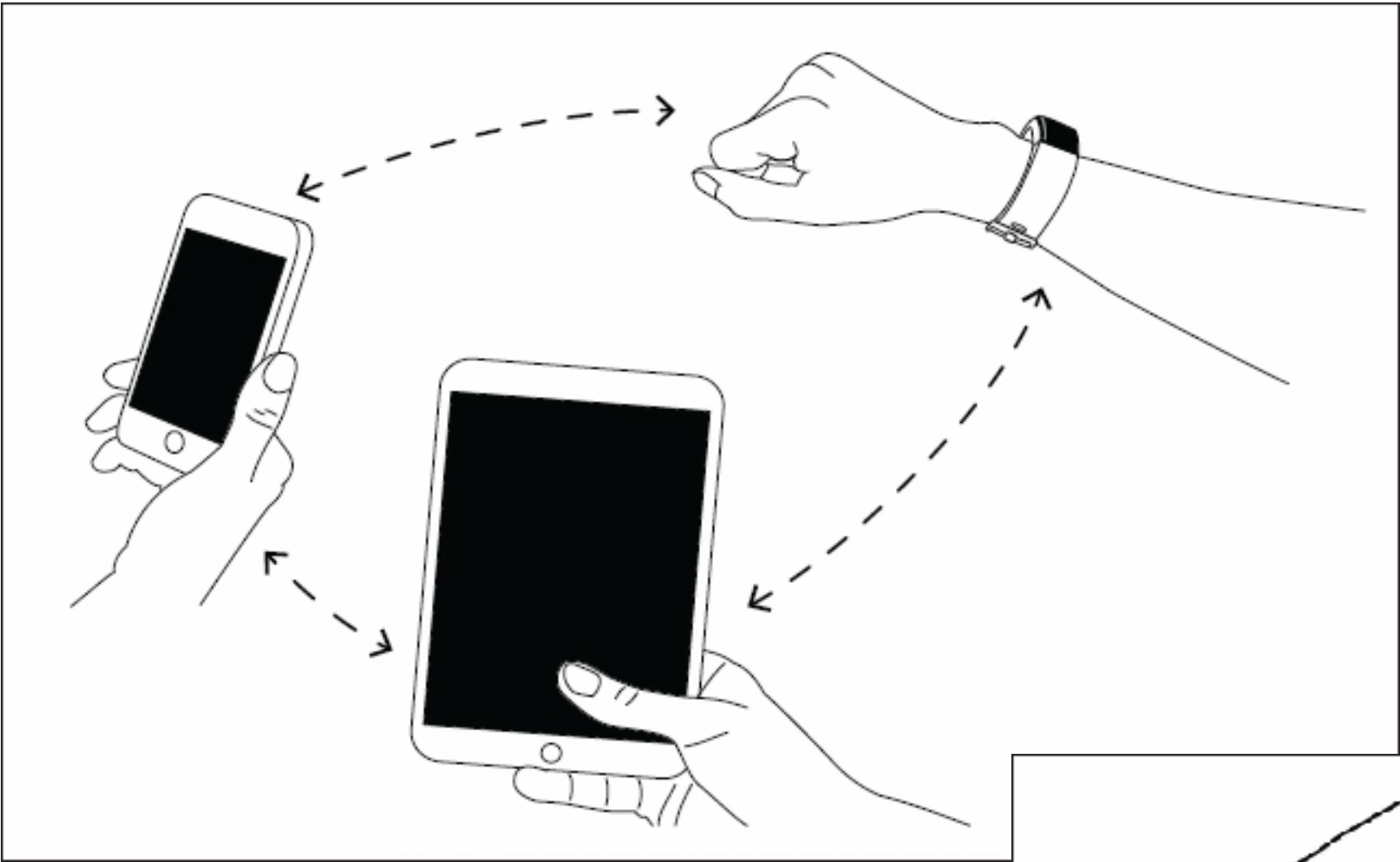
Technologies Web synchrones et multi-dispositifs

CM3 - Design Adaptatif et Reconnaissance de Gestes

<https://aurelient.github.io/react-iga/2020/>

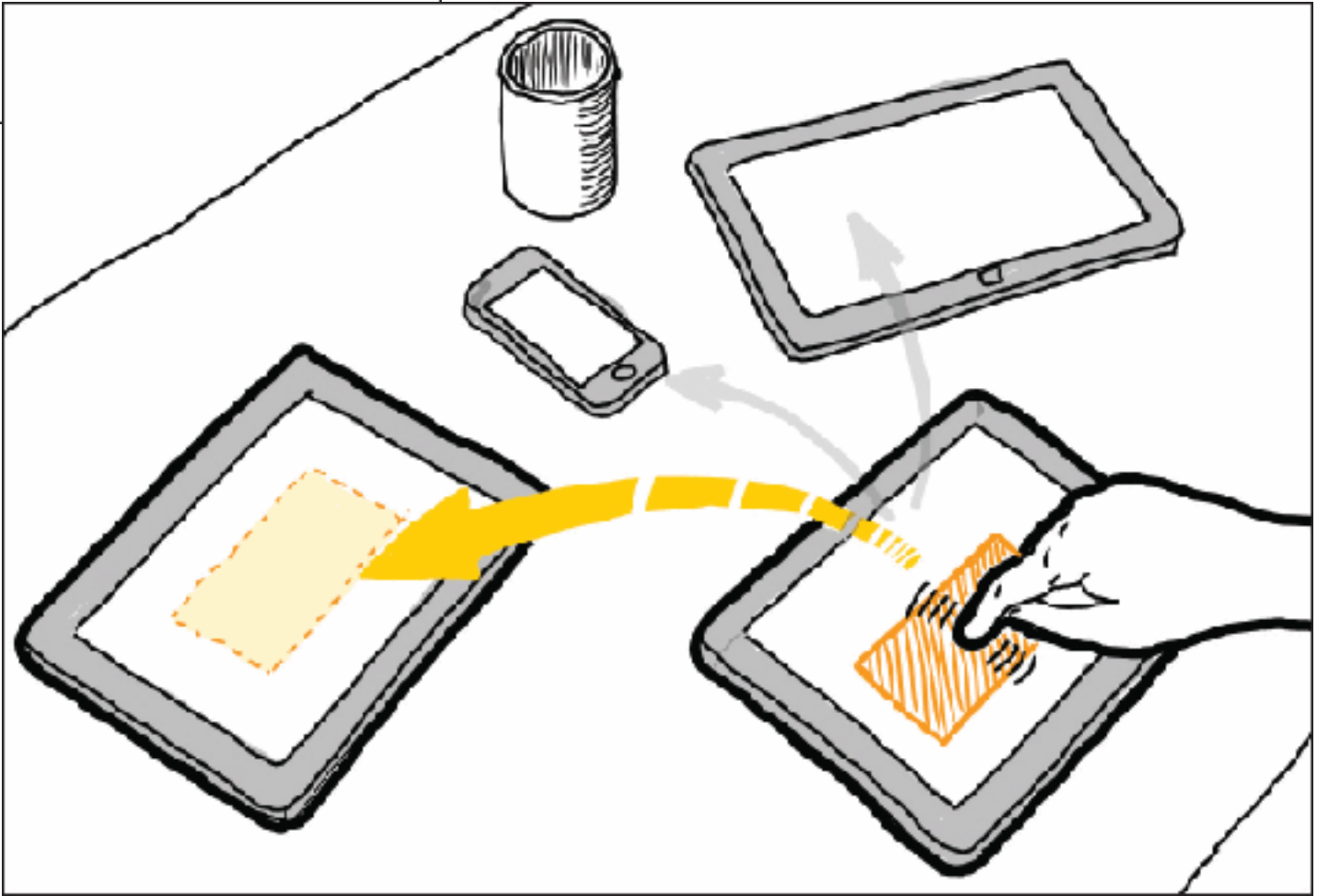
Version automne à Lyon : <https://aurelient.github.io/tiw8/2020/>





sketch by Lindsay MacDonald Vermeulen

sketch by Nicolai Marquardt



Plan

- ▶ **Responsive Design: rappels et limites**
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées:
 - ▶ reconnaissance de gestes

Design Responsif

Adaptation au dispositif d'affichage en s'appuyant sur :

- ▶ des tailles relatives : % ou em plutôt que px ou cm
 - ▶ pour les images,
 - ▶ pour les fonts,
 - ▶ pour les div,
- ▶ avec des limites min/max
- ▶ des vecteurs / glyphicon (-> relatif plutôt qu'absolu)
- ▶ des grilles fluides via des media queries
 - ▶ Des règles CSS différentes selon le dispositif
 - ▶ Souvent la largeur (width) de l'écran ou de la fenêtre.

Les limites

Adaptation uniquement du côté affichage

- ▶ Pas du côté de l'entrée (touch vs. souris)

Pas de réflexion sur les usages

- ▶ Différents selon le dispositif : au bureau ou dans les transports

Centré sur un dispositif

- ▶ Ne pense pas l'orchestration de plusieurs dispositifs

Plan

- ▶ Responsive Design: rappels et limites
- ▶ **Design adaptatif, Interfaces plastiques**
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées:
 - ▶ reconnaissance de gestes

Design adaptatif

Idée générale :

Une “application” optimisée pour chaque dispositif.

Question :

Comment adapter efficacement ?

Interfaces plastiques

http://iihm.imag.fr/pubs/2012/LivreAmi-Chap9-Plasticite-CoutazCalvary_.pdf

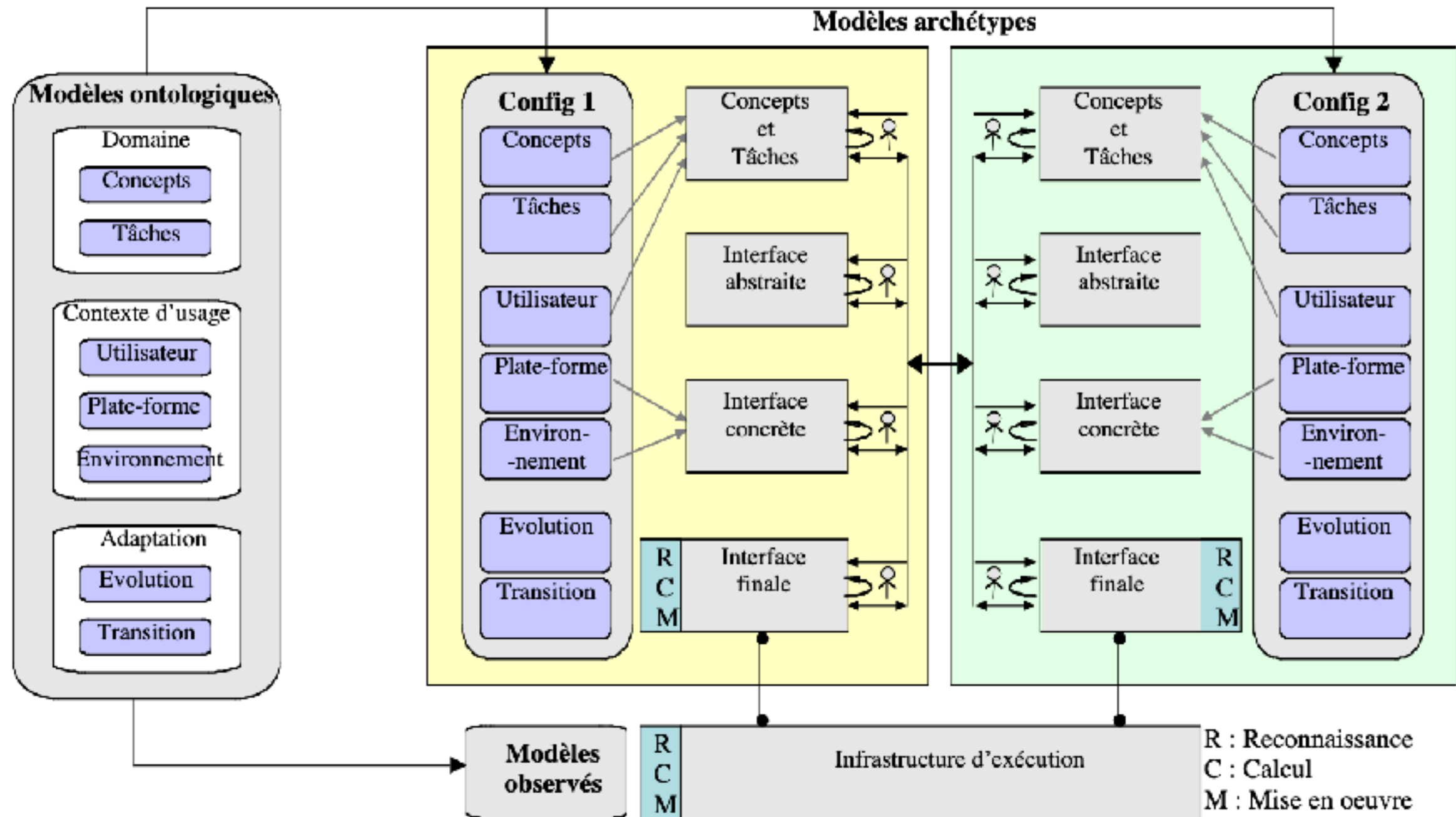
“La plasticité de l’interface homme-machine d’un système interactif dénote la capacité d’adaptation de cette interface au contexte d’usage pour en préserver l’utilité et l’utilisabilité et, par extension, la valeur tout en accordant à l’utilisateur les moyens de contrôle adéquats.” Coutaz et al. ‘12

Principes:

- ▶ UI remodelage (Suppression, Insertion, Substitution, Réorganisation)
- ▶ UI Redistribution (niveau Application, espace de travail, interacteur, pixel)
- ▶ Maintien l’état (système, tâche, action)
- ▶ Meta-UI (avec ou sans negociation)

Un méta-modèle

http://iihm.imag.fr/pubs/2012/LivreAmi-Chap9-Plasticite-CoutazCalvary_.pdf



Plan

- ▶ Responsive Design: rappels et limites
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées
 - ▶ Interaction gestuelle
 - ▶ Rubine
 - ▶ \$1 Recognizer

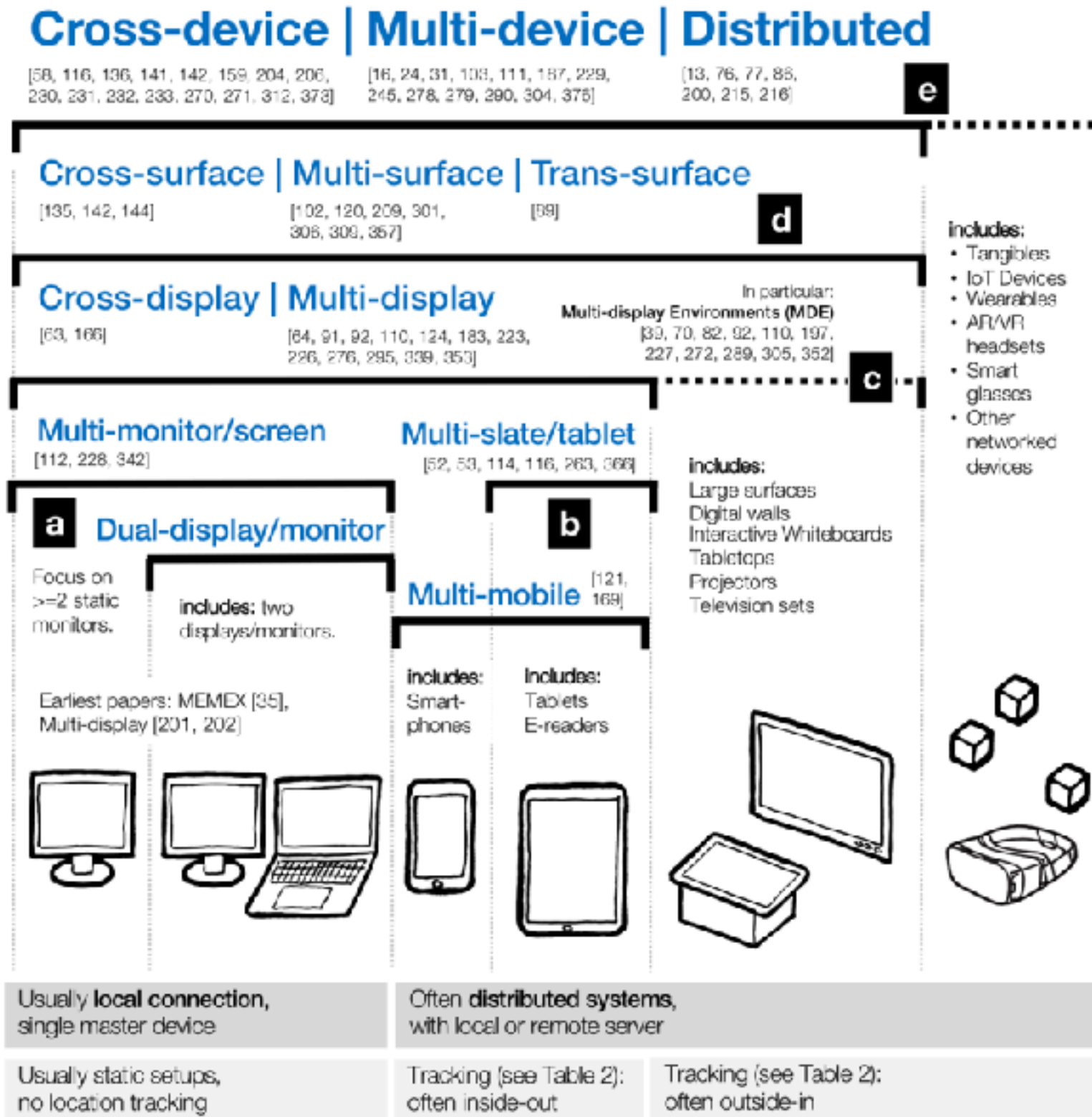
Pick and Drop – Rekimoto '97

**Multiple Computer User Interfaces:
"Beyond the Desktop"
Direct Manipulation Environments**

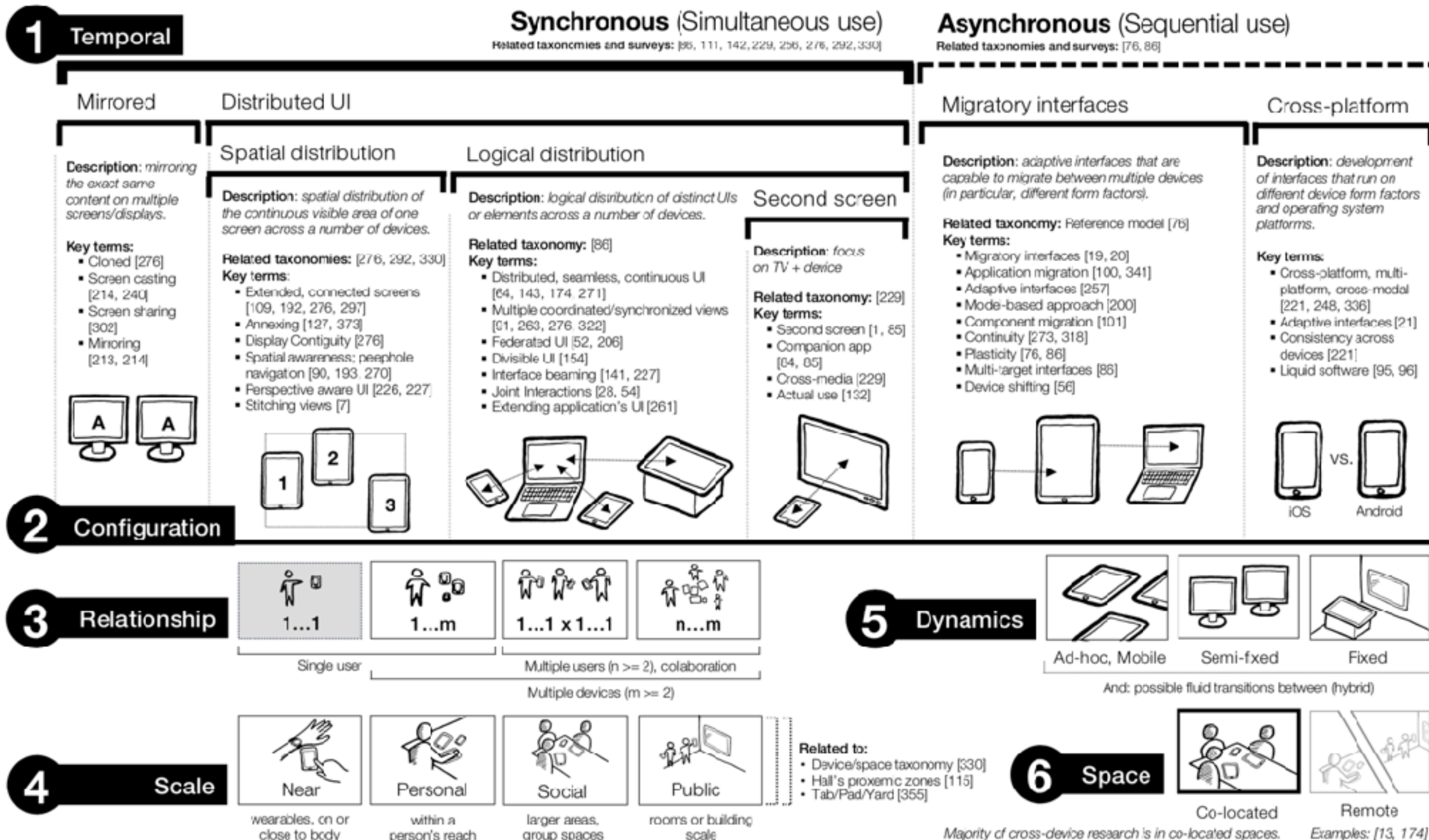
Jun Rekimoto

**Interaction Laboratory
Sony Computer Science Laboratories, Inc.**

Les types d'interaction cross-device



Les éléments à considérer



Les phases à gérer

Phase 1

Configuration

Setting up cross-device configurations of devices including pairing, combining, connecting devices.

Phase 2

Content Engagement

Techniques aimed at interacting with, transferring or exploring content, data, visualisations, or interfaces that spread across multiple devices.

Phase 3

Disengagement

Techniques to disconnect cross-device setups or configurations, interactions and applications.

Apple Continuity

<https://support.apple.com/en-us/HT204681>



Handoff: Start work on one device, then switch to another nearby device and pick up where you left off.



Universal Clipboard: Copy content such as text, images, photos, and videos on one Apple device, then paste the content on another Apple device.



iPhone Cellular Calls: Make and receive calls from your Mac, iPad, or iPod touch when those devices are on the same network as your iPhone.



Text Message Forwarding: Send and receive SMS and MMS messages from your iPhone on your Mac, iPad, and iPod touch.



Instant Hotspot: Connect to the Personal Hotspot on your iPhone or iPad (Wi-Fi + Cellular) from your Mac, iPad, iPod touch, or another iPhone, without entering a password.



Auto Unlock: Get instant access to your Mac when wearing your Apple Watch, and quickly approve other requests to enter your Mac administrator password.



Continuity Camera: Use your iPhone, iPad, or iPod touch to scan documents or take a picture and have it appear instantly on your Mac.



Continuity Sketch: Create a sketch on your iPad, iPhone, or iPod touch, and easily insert it into a document on your Mac.



Continuity Markup: Use your iPad, iPhone, or iPod touch to add sketches, shapes, and other markup to a Mac document, and see the changes live on your Mac.



Sidecar: Use your iPad as a second display that extends or mirrors your Mac desktop. Or use it as a tablet input device to draw with Apple Pencil in Mac apps.



AirDrop: Wirelessly send documents, photos, videos, websites, map locations, and more to a nearby iPhone, iPad, iPod touch, or Mac.



Apple Pay: Shop online on your Mac and complete your purchase using Apple Pay on your iPhone or Apple Watch.

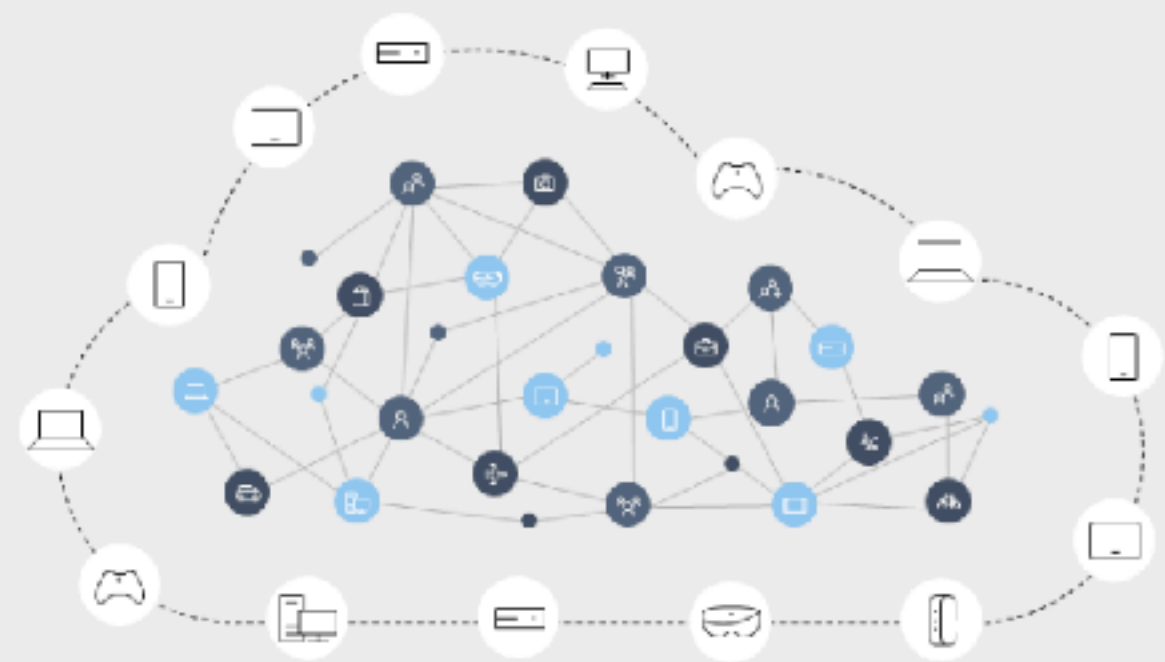
Microsoft

<https://developer.microsoft.com/en-us/windows/project-rome>

Project Rome

Build people centric experiences on all devices.

A device-independent platform for building people-centric experiences that span all devices.



Check out Graph Notifications.

[Read more](#)

Technologies

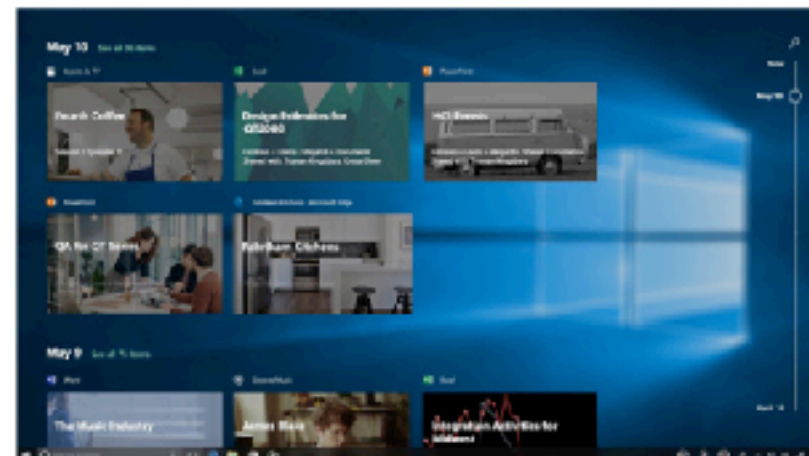


Microsoft Graph Notifications

New at Ignite 2018

Graph Notifications offers an enterprise-compliant, people-centric, and cross-platform notifications platform using the Microsoft Graph.

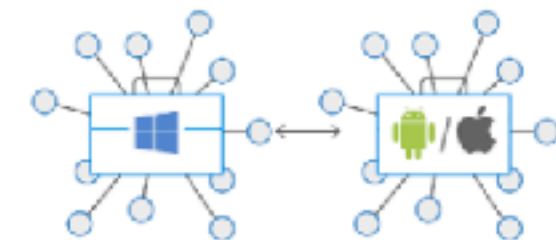
[LEARN MORE >](#)



Activities/Timeline

Activities make users more productive by helping them resume important tasks in your app quickly across devices and platforms in experiences like Windows Timeline, Cortana and Microsoft Launcher.

[LEARN MORE >](#)



Device Relay

Users often start productivity tasks and entertainment on one device and continue them on another. You can use the device relay APIs to make these experiences seamless.

[LEARN MORE >](#)

Quels outils de développement pour le Web ?

- ▶ Progressive Web Apps, que pour un dispositif.
- ▶ Manque
 - ▶ Des bibliothèques
 - ▶ D'outils de debugging
 - ▶ Des liens entre applications (traitées en silo)

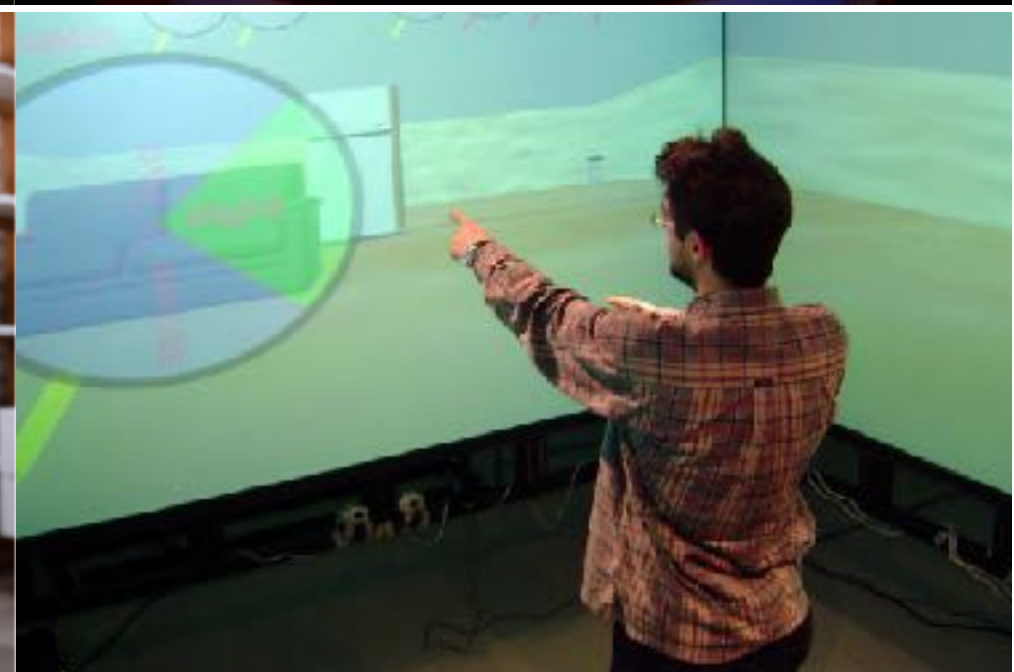
Plan

- ▶ Responsive Design: rappels et limites
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées
 - ▶ Interaction gestuelle
 - ▶ Rubine
 - ▶ \$1 Recognizer

Contexte: l'interaction post-WIMP

WIMP: Windows Icon Menu Pointer



L'interaction gestuelle

<https://uxmag.com/articles/new-design-practices-for-touch-free-interactions>



Deux familles d'interaction gestuelles

Interaction sur écran tactile :

- ▶ gestes 2D mono ou multi-points.

Interaction libre (free-form) :

- ▶ peu de contraintes directes
- ▶ Utilisation d'un contrôleur et/ou du corps

Qu'est qu'un geste

“A gesture is a form of *non-verbal communication* or non-vocal communication in which *visible bodily actions* communicate *particular messages*, either in place of, or in conjunction with, speech. Gestures include *movement of the hands, face, or other parts of the body*. Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention.”

A. Kendon, *Gesture: Visible Action as Utterance*, Cambridge University Press, 2004

Propriétés des gestes

1. Statique (posture) ou dynamique (mouvement)
2. Définit spatialement et temporellement
3. Transmet de l'information
4. Délibéré

Les fonctions du geste de Cadoz (1994)

Gestes sémiotiques : communique de l'information à l'environnement. Ex: gestes qui accompagnent le langage, langue des signes, de chefs d'orchestres...

Gestes ergotiques : transforme physiquement l'environnement (la matière) pour la former, la transporter la casser...

Gestes épistémiques : pour apprendre de l'environnement à travers une exploration tactile et haptique.

Les gestes sémiotiques (nous intéressent)

Une sous-classification:

- ▶ Gestes symboliques, spécifiques à une culture (ex: 🤞), seuls à pouvoir être interprétés sans besoin de contexte.
- ▶ Gestures déictiques, ex: pointage
- ▶ Gestes iconiques, pour transmettre des informations sur la taille, la forme ou l'orientation d'un objet du discours.
(ex: mouvement vers le bas)
- ▶ Gestures pantomimes, qui mime le mouvement d'un objet tel qu'utilisé (ex: j'ai tourné le volant comme ça).

5 questions en interaction gestuelle

- ▶ Comment les utilisateurs s'adressent au système ?
- ▶ Comment le système établit qu'il est prêt à recevoir une entrée?
- ▶ Qu'est ce que le système peut faire ?
- ▶ Comment le système répond ?
- ▶ Comment le système permet d'éviter ou de réparer une erreur ?

Définir un vocabulaire de gestes

- ▶ Comprendre le contexte
- ▶ Choisir le type d'interaction
- ▶ Définir le vocabulaire de gestes, et les commandes associées
- ▶ Implémenter le système de reconnaissance
- ▶ Définir des moyens d'apprentissage / de guidage
- ▶ Évaluer la qualité en terme de reconnaissance, de compréhension, d'efficacité, de mémorisation, ...

Type d'interaction gestuelle

Directe



Distance



Autour



Intégré



Plan

- ▶ Responsive Design: rappels et limites
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées
 - ▶ Interaction gestuelle
 - ▶ **Rubine**
 - ▶ **\$1 Recognizer**

Les approches de reconnaissance

Basé sur des templates (modèles): on aligne et calcule une distance, pour comparer le geste en entrée à des modèles pré-enregistrés.

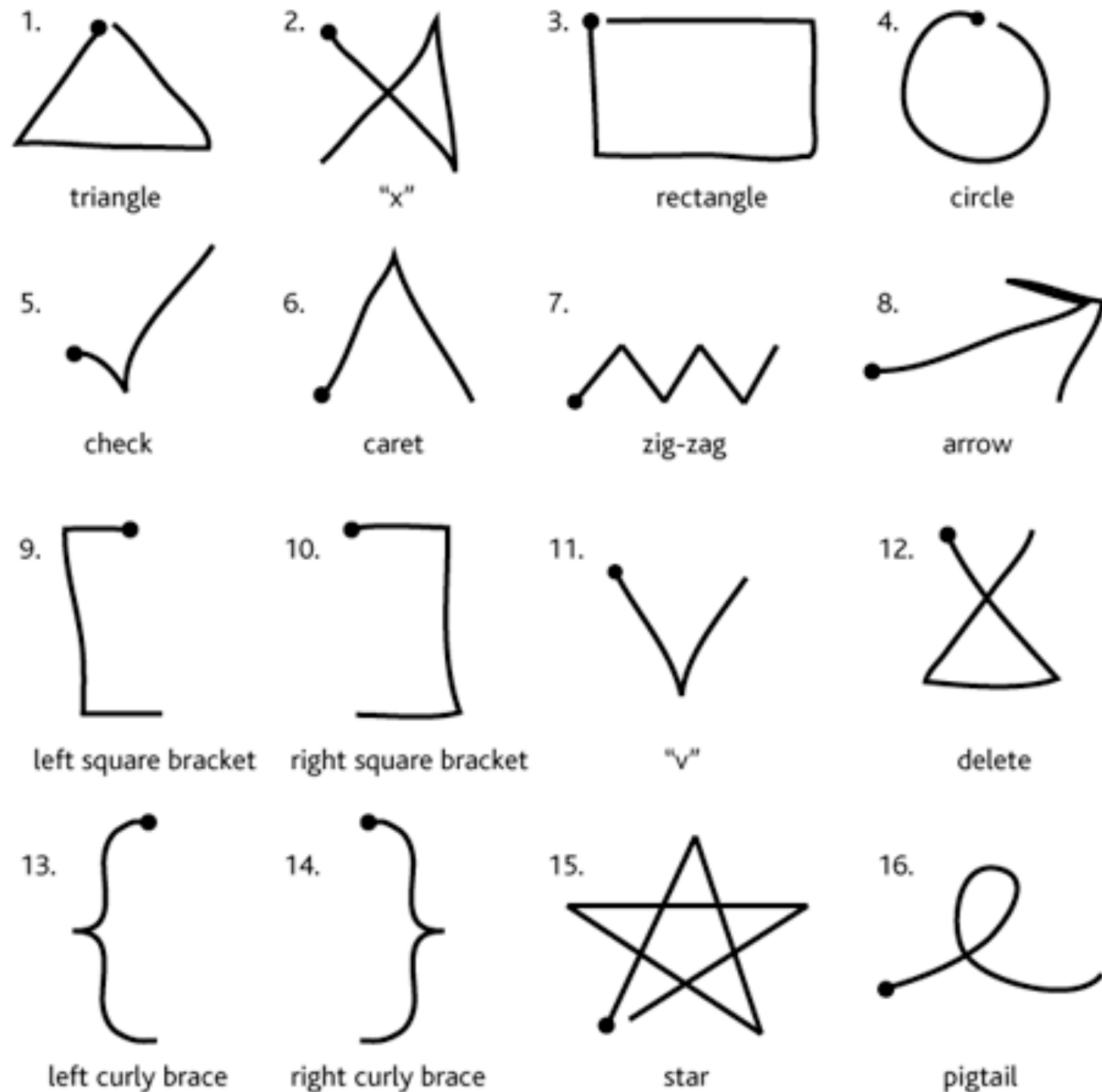
- ▶ Rubine, 1991
- ▶ \$1, Wobbrock et al., 2007

Approches par Machine Learning: on apprend sur un jeu d'entraînement. Puis on calcule l'appartenance du geste en entrée aux classes de gestes apprises.

- ▶ kNN - HMM - SVM - ANN

Gestes discrets et continus: unistroke

Gestes réalisés
“d’un seul trait”



Plan

- ▶ Responsive Design: rappels et limites
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)
- ▶ Modalités d'entrées avancées
 - ▶ Interaction gestuelle
 - ▶ **Rubine**
 - ▶ \$1 Recognizer

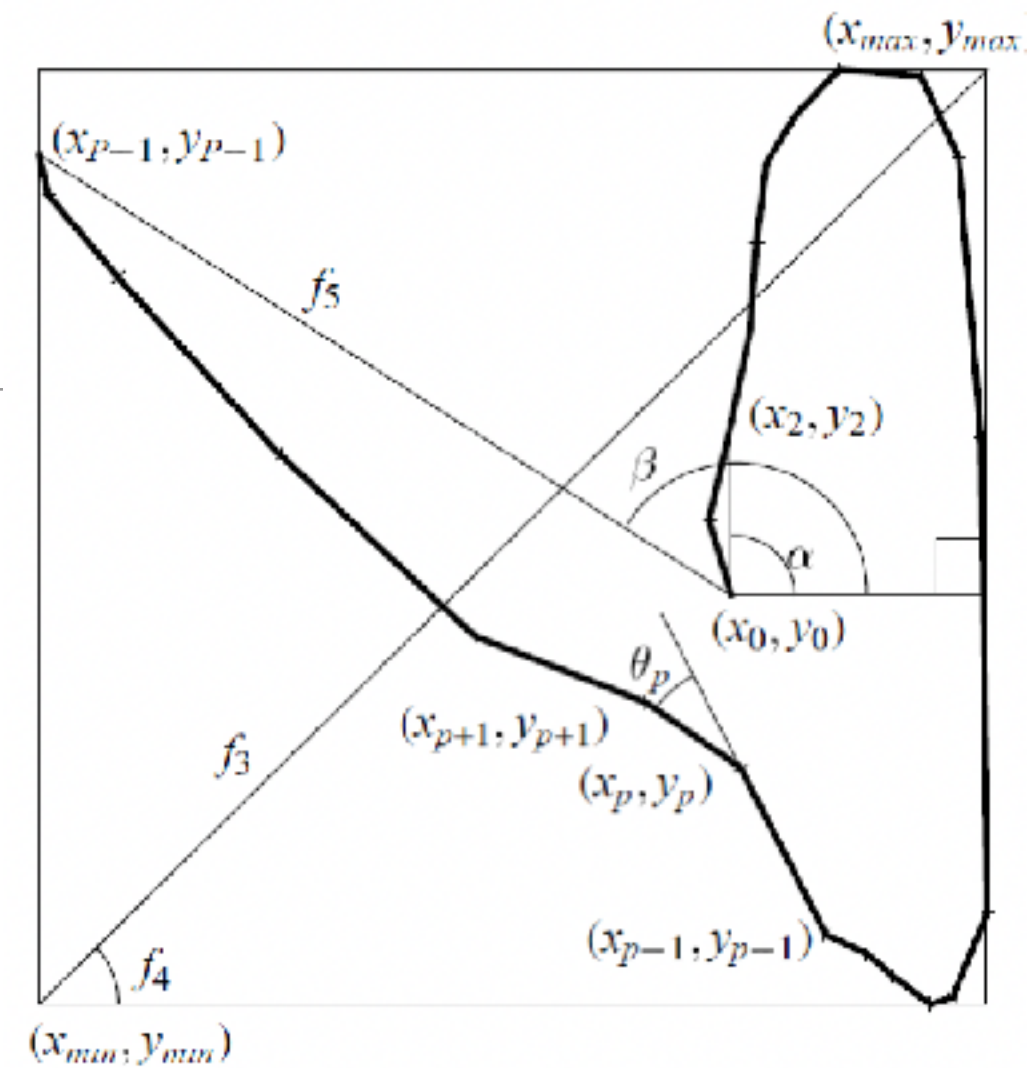
Rubine

Les étapes :

1. L'utilisateur effectue un geste
2. Le geste est capturé sous forme d'une liste de points
3. On calcule un vecteur de propriétés (features)
4. On compare le vecteur à ceux des modèles (templates)
5. The geste reconnu est celui avec le score le plus haut

Les features

- F1** cos of the gesture's initial angle
- F2** sin of the gesture's initial angle
- F3** length of the bounding box diagonal
- F4** angle of the bounding box diagonal
- F5** distance between the first and the last point
- F6** cos of the angle between the first and the last point
- F7** sin of the angle between the first and the last point
- F8** total gesture length
- F9** total angle traversed
- F10** sum of the absolute value of the angle at each point
- F11** sum of the squared value of those angles
- F12** maximum speed of the gesture
- F13** duration of the gesture



Entrainement / Classification

Phase d'entraînement



Reconnaissance / classification phase

$$v_{\hat{c}} = w_{\hat{c}0} + \sum_{i=1}^F w_{\hat{c}i} f_i$$

Plan

- ▶ Responsive Design: rappels et limites
- ▶ Interfaces plastiques, design adaptatif
- ▶ Interaction inter-dispositif (cross-device)

- ▶ Modalités d'entrées avancées
 - ▶ Interaction gestuelle
 - ▶ Rubine
 - ▶ **\$1 Recognizer**

\$1 recognizer

Marche pour l'unistroke, mais nombreuses extensions pour d'autres types de gestes

Reconnaissance invariante selon

- ▶ La rotation
- ▶ L'échelle
- ▶ La position

\$1 Les étapes

1. Creation de templates
2. Capture du geste
3. Ré-échantillonnage
4. Comparaison à des templates invariant

Ré-échantillonnage

On ré-échantillonne le geste capturé pour avoir N points équidistants (généralement $N=64$)

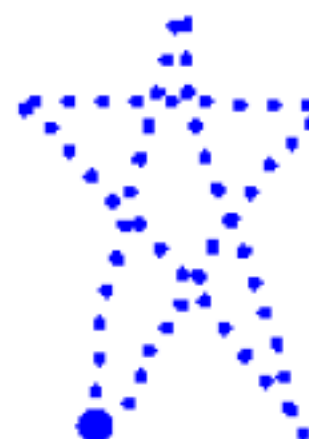
- ▶ Measure la longueur L du chemin de M points initiaux
- ▶ On calcule l'incrément $I = L/(N-1)$
- ▶ On rajoute un point (coordonnées calculées par régression linéaire) si la distance entre deux points du signal initial est $> I$



Raw gesture



$N = 32$



$N = 64$

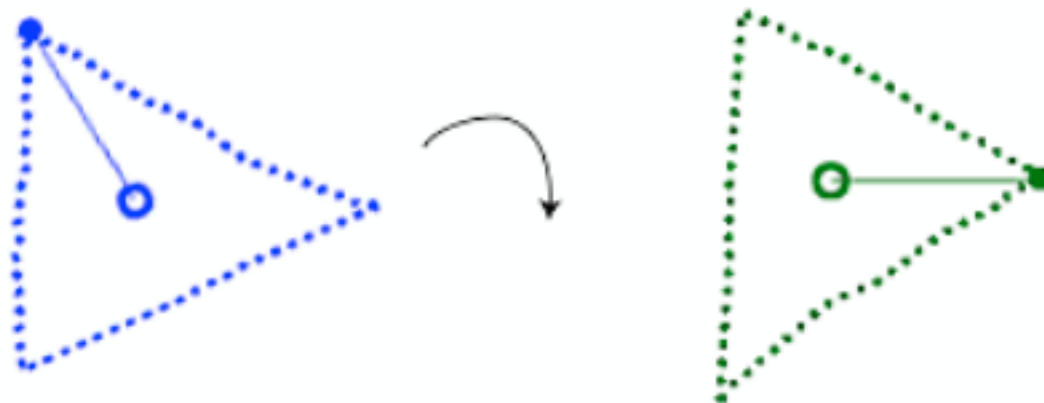


$N = 128$

Alignement angulaire

On “tourne” le geste selon son “angle indicatif”, en cherchant l’espace des angles permettant un meilleur alignement des points

- ▶ Calcul du centroid du geste
- ▶ Calcul de l’angle entre le 1^e point de la trajectoire, le centroïde et la ligne d’horizon.
- ▶ Rotation des points selon cet angle



Mise à l'échelle

Le geste est passé à l'échelle d'un carré de référence (non-uniforme). Après le passage à l'échelle le geste est translaté vers un point de référence (le centroïde est positionné en (0,0))

- ▶ Calcul de la bounding box (x_{\min} , x_{\max} , y_{\min} , y_{\max}) -
- ▶ Passage à l'échelle du carré de référence (à la taille prédéfinie)
- ▶ Translation du centroid vers (0,0)

Reconnaissance

Le geste C est comparé à chaque Template de geste T_i

- Calcul de distance d_i entre C et T_i pour chaque partie du chemin. Le template avec la distance la plus faible à C est le résultat.

$$d_i = \frac{\sum_{k=1}^N \sqrt{(C[k]_x - T_i[k]_x)^2 + (C[k]_y - T_i[k]_y)^2}}{N}$$

- d_i est converti sur une échelle $[0,1]$.

$$score = 1 - \frac{d_i^*}{\frac{1}{2} \sqrt{size^2 + size^2}}$$

Avantages et limites du \$1 recognizer

Avantages:

- ▶ Facile à comprendre
- ▶ Facile à implémenter, léger, rapide.
- ▶ “Apprend” sur un seul geste
- ▶ Résilient aux variations de vitesse et d'échantillonnage
- ▶ Résilient aux variations d'orientation et d'échelle

Inconvénients:

- ▶ Ne sait pas distinguer un cercle d'un carré, une ellipse d'un cercle
- ▶ Ne peut pas distinguer l'orientation d'une flèche