

Ordinateur porté: dispositifs d'entrée-sortie.

Diane Lingrand, Wagner Ourique de Morais et Jean-Yves Tigli

Équipe Rainbow - Laboratoire I3S (UNSA/CNRS) UMR 6070
B.P. 121 - 06903 Sophia Antipolis - FRANCE
{lingrand,tigli}@i3s.unice.fr

RESUME

L'ordinateur porté pose un certain nombre de défis tels que celui des dispositifs d'entrée. En effet, selon le contexte d'utilisation et la disponibilité des ressources motrices de l'utilisateur, différents dispositifs d'entrée sont utilisés.

Substituer tout dispositif par un dispositif équivalent au niveau purement fonctionnel s'avère insuffisant.

Après avoir introduit une classification des dispositifs d'entrée, nous montrons la pertinence de confronter un modèle de l'utilisateur à celui des dispositifs d'entrée. Un exemple simple basé sur les ressources motrices de l'utilisateur permet d'illustrer notre démarche.

Mots Cles

périphériques d'entrée, ordinateur porté.

ABSTRACT

Among the different challenges induced by Wearable Computing, is the one of IO devices. The choice of appropriate input device is driven by the user context and the availability of motor resources of the user.

The substitution of one device by another with same functionalities is not sufficient.

After the introduction of input devices classification, we show that it is necessary to introduce a user model for input devices considerations. A simple example based on motor resources of the user permit to illustrate our method.

CATEGORIES AND SUBJECT DESCRIPTORS

D.2.11 Software Engineering Software Architectures Data abstraction, Domain-specific architectures

GENERAL TERMS

Design, Theory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. IHM 2005, September 27-30, 2005, Toulouse, France. Copyright ACM X-XXXXX-XXX-X/XX/XXXX \$5.00



Figure 1. Périphériques d'entrée

Keywords

IO devices, wearable computer.

INTRODUCTION

L'ordinateur porté [19] ou vestimentaire [9] est un véritable ordinateur que l'utilisateur porte sur lui. Ce n'est pas un ordinateur portable, actif seulement lorsque l'utilisateur est fixe et éteint lors de ses déplacements, ni un pda qui est inactif lorsqu'il est dans une poche. L'ordinateur porté est constamment en fonctionnement, que ce soit à l'arrêt ou pendant les déplacements de l'utilisateur et doit fournir un mode d'utilisation adapté au contexte d'utilisation.

De ce fait, l'ordinateur porté pose un certain nombre de défis, clairement posés dans [19, 22]. Parmi ces défis, nous nous intéressons dans ce papier à celui du choix des périphériques d'entrée.

La classification des périphériques d'entrée est le sujet de nombreuses études (voir section suivante). Dans le cas précis de l'ordinateur porté, on cherche à établir une classification permettant à la fois de déterminer des équivalences de périphériques en terme de fonction (saisie d'un nombre, déplacement d'un curseur) mais aussi de déterminer les périphériques les plus adaptés à un contexte d'utilisation. Pour un certain nombre d'applications telles que la rédactions de notes ou de courriers, un utilisateur présent à son bureau va préférer le clavier et la souris avec lesquels il est désormais efficace. Cependant, lorsque l'utilisateur va se déplacer vers le lieu de sa réunion, que ce soit à pied, en voiture ou en bus, il est alors impensable pour lui d'utiliser ces mêmes périphériques. D'autres entrées seront bien plus adaptées et font l'objet d'études et de publications régulières. Les plus populaires dans le domaine de l'ordinateur porté sont l'écran tactile, le *Twidler* clavier une main, les *GestureWrist* et *GesturePad* [13, 17, 18, 21] (voir figure 1) ...

Dans la suite de ce papier, on va s'intéresser aux classifications existantes et mettre en oeuvre une nouvelle classification permettant d'introduire une notion d'équivalence de périphérique adapté à l'ordinateur porté et au contexte de

l'utilisateur. Dans cette étude, on vise à la fois une gestion dynamique des périphériques pour une application sur ordinateur porté, la possibilité d'associer des événements d'un périphérique vers un autre (ex: vers clavier/souris) et l'utilisation de nouveaux périphériques sans modification d'applications existantes.

CLASSIFICATION DES PÉRIPHÉRIQUES D'ENTRÉE

Plusieurs auteurs ont proposé des taxinomies des périphériques d'entrée. Foley et Wallace [11] classent les périphériques selon quatre famille: saisie, localisation, bouton et valeur. Dans [12], une version associant les périphériques aux actions pouvant être effectuées est proposée. Cependant, cette classification ne suffit pas à déterminer le périphérique correspondant aux besoins.

De son côté, Buxton [4] classe les périphériques sur un graphe bidimensionnel en fonction du nombre de degrés de liberté et des grandeurs mesurées (mouvement, pression, ...) ainsi que de son type de contact avec l'utilisateur. Card et ses collègues [5] ont étendu cette taxonomie en distinguant les valeurs absolues des valeurs relatives et en séparant les composantes de translation et de rotation. Cette classification, bien que plus précise que les précédentes ne permet pas encore de donner d'équivalence précise entre périphérique ou bien de donner le plus approprié à un cas précis.

Jacob et Sibert [16] ont commencé à percevoir que la dimension manquant aux taxonomies précédentes concerne la perception cognitive. Cette taxonomie fait la distinction entre les périphériques intégrales (considère les degrés de liberté comme un bloc) et les périphériques séparables (traite les degrés de liberté individuellement). Cette classification a été utilisée notamment par [2, 15].

A côté de ces différentes taxonomies, plutôt d'ordre théorique, on trouve également des comparaisons expérimentales basées sur des mesures comme la loi de Fitts [10] ou la loi de trajectoire (*steering*) [1]. On trouve par exemple des études comme [15].

Nous reprenons ici les principales idées des taxonomies précédentes et présentons le résultat sous forme d'arbre (voir figure 2).

AUTO-ADAPTATION PAR SUBSTITUTION DE DISPOSITIFS

Lorsque l'utilisateur d'un ordinateur porté arrive à son bureau, il peut vouloir changer de périphériques d'entrée et utiliser le clavier et la souris classiques afin de continuer le traitement de son courrier, par exemple. Il est impensable de lui demander de relancer l'application, de modifier les paramètres de son système d'exploitation, voire encore pire, de redémarrer sa machine.

Chaque concepteur de périphérique est alors en charge de la conception de son driver selon une interface imposée par le constructeur de l'OS (exemple: Hardware Abstract Layer HAL de Microsoft). Le chargement dynamique de driver

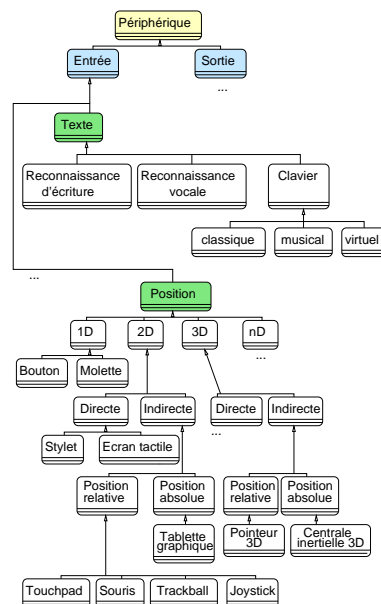


Figure 2. Classification des périphériques d'entrée.

répond alors aux contraintes précitées mais ne gère pas la concurrence des dispositifs.

Il suffit, pour s'en convaincre, d'utiliser simultanément les trois souris sur un portable: souris externe, *TouchPad*, dispositif de pointage *ThinkPad*. Le traitement des événements souris est effectuée au fur et à mesure de l'arrivée des événements. Heureusement, dans ce cas, il s'agit de dispositifs de pointage simple et de déplacements relatifs ce qui permet de passer d'une souris à l'autre sans saut de curseur. Cela ne serait pas gérable pour des dispositifs plus complexes : comment autoriser un événement clavier dans un cycle de reconnaissance vocale ?

Comment gérer intelligemment un système multi-dispositifs ? Cette question amène deux réponses. La première consiste à établir une équivalence des dispositifs. La seconde concerne une gestion commune de ces périphériques, par exemple en utilisant des techniques permettant d'envoyer au système d'exploitation des événements de type souris/clavier ou autres, quelque soient les périphériques utilisés.

Plus récemment, des travaux permettent au concepteur de construire ses propres périphériques d'entrée/sortie et de les intégrer aisément dans une application logicielle comme nouveau composant ActiveX dans l'approche Phidget [14], nouveau composant avec un modèle spécifique basé sur Java Swing dans ICON [7], ou nouveau composant JavaBean dans WCOMP [6]. L'utilisateur choisit alors dans le schéma d'assemblage des composants la gestion qu'il souhaite pour les dispositifs d'entrée.

Dans un soucis d'expérimentation, nous partons d'applications de type WIMP classiques. Ceci nous conduit dans un premier temps à étudier des alternatives à des périphériques simples tels que la souris ou le clavier. Notre méthodologie

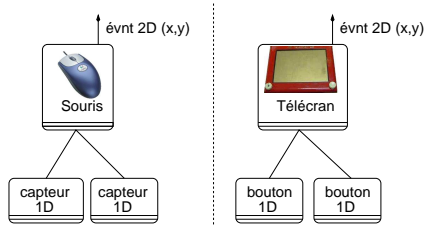


Figure 3. Équivalence au niveau du pointage 2D entre la souris et le télécran.

s'applique néanmoins à toute sorte d'interfaces WIMP ou Post-WIMP basées sur des dispositifs d'entrée de tout type.

NOTION D'ÉQUIVALENCE DE PÉRIPHÉRIQUES

Équivalence purement fonctionnelle

Du point de vue d'une application, deux périphériques sont équivalents lorsqu'ils permettent d'obtenir les mêmes fonctionnalités. On peut aussi définir l'équivalence par la similarité des événements envoyés par deux périphériques différents (mêmes informations).

Prenons le cas de la souris. Un événement envoyé par une souris contient sa position et l'état de ses boutons. Il en est de même pour une *trackball*. Ces deux périphériques sont donc fonctionnellement équivalents. Considérons maintenant uniquement la fonction de pointage 2D de la souris. Le déplacement de la souris sur son tapis (ou tout support plan) permet de déplacer, d'une main, un curseur 2D à l'écran. De son côté, le télécran est muni de deux boutons rotatifs. Le premier permet des déplacements horizontaux du curseur tandis que le second permet des déplacements verticaux. Utilisés ensemble, ces deux boutons rotatifs permettent tout déplacement 2D du curseur. Si on regarde les événements générés par ces deux dispositifs, il s'agit dans les deux cas de déplacements relatifs 2D: ils sont donc équivalents, de ce point de vue.

L'expérience de la comparaison du pointage souris 2D et du télécran met en évidence que l'aspect événements n'est pas suffisant pour établir une équivalence en périphérique. Cette notion d'équivalence nécessite d'être plus clairement définie.

Si on exclue les cas particuliers des lignes horizontales et verticales, le pointage souris est bien plus facilement utilisable que le télécran. La difficulté provient essentiellement de la synchronisation entre les deux mains gérant chacune une dimension.

Prise en compte de l'utilisation

Différents aspects de l'utilisation ou de l'utilisateur doivent être pris en compte.

La nécessité ou non d'un apprentissage

Une utilisation simple et triviale (déjà maîtrisée dans la vie courante) ou nécessitant un apprentissage: même la souris dont l'utilisation nous paraît aujourd'hui triviale ne l'est pas pour de jeunes enfants lors de la première utilisation [23].

Les ressources motrices de l'utilisateur

L'utilisation des ressources motrices de l'utilisateur: la souris occupe une main, le télécran en occupe deux. On en vient à spécifier alors, dans la décomposition des périphériques telle que présentée figure 3, les entrées de l'utilisateur ainsi que les ressources motrices.

Le retour du périphérique

Il est impensable d'utiliser un périphérique sans retour de la part de la machine: un texte entré au clavier doit pouvoir être lu à l'écran ou écouté après synthèse vocale. Ce sont donc plutôt des couples périphériques d'entrée / périphériques de sortie qui doivent être pris en compte.

Le contexte de l'utilisation

La dépendance d'un environnement: la souris classique nécessite un support, certaines souris nécessitent des supports spécifiques et la *trackball* ne nécessite aucun support. Cette propriété déterminera la préférence de tel ou tel périphérique en fonction du contexte d'utilisation (bureau, marche, ...).

Certains travaux mettent en avant la nécessité d'introduire l'utilisateur dans l'analyse des interactions entre l'homme et la machine dans le cas des systèmes mixtes et a fortiori pour les ordinateurs portés. Dans ASUR [8], le système complet est représenté par un graphe caractérisant des relations entre composants de type Adaptateurs (A), Système informatique (S), Utilisateur (U) et entités Réelles (R). Le composant utilisateur est alors pourvu d'un certain nombre de caractéristiques permettant d'analyser plus finement ses relations avec les autres composants et notamment les dispositifs d'entrée/sortie A_{in} et A_{out} . Dans IRVO [20], l'approche est similaire mais l'utilisateur est principalement représenté par trois canaux: Audio, Visuel, Kinesthésique/Haptique.

Ces travaux entrent dans le cadre d'une mouvance qui voit le modèle de l'utilisateur se diversifier afin de pratiquer une analyse toujours plus fine des interactions entre utilisateur et ordinateur porté.

Exemple de la prise en compte des ressources motrices

Nous revenons ici sur l'exemple de la comparaison de la souris et du télécran. Une souris nécessite l'utilisation d'une main. Un bouton rotatif nécessite lui aussi l'usage d'une main. Dans ce cas simple, présenté figure 3, les ressources motrices sont des mains. Le télécran somme les différentes ressources motrices de ses boutons pour obtenir la nécessité d'être manipulé par deux mains simultanément. Cela montre pourquoi il est plus difficile de tracer une droite oblique avec un télécran plutôt qu'une souris.

De cet exemple, on peut aisément passer à la souris 3D: celle-ci ne nécessite toujours qu'une seule main. Un télécran 3D nécessiterait quant à lui 3 mains ce qui devient alors inutilisable. Pour d'autres dispositifs, cette décomposition pourra concerner des ressources comme des doigts pour des touches ou boutons poussoir, bras pour des gestes ou la parole pour une commande vocale.

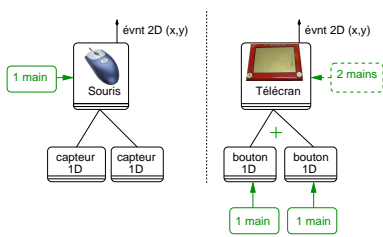


Figure 4. Équivalence au niveau du pointage 2D entre la souris et le télécran: prise en compte des ressources motrices de l'utilisateur.

CONCLUSION ET PERSPECTIVES

Notre modèle assez simple nécessite d'être amélioré selon deux directions.

La première concerne une généralisation à l'ensemble des périphériques d'entrée de la modélisation au niveau de dispositifs physiques atomiques tels que bouton activable par un doigt ou potentiomètre activable par une main afin de remonter aux niveaux traités dans la classification figure 2.

Une autre direction concerne une modélisation plus complète de l'utilisateur en utilisant par exemple le modèle ICS [3] englobant non seulement des aspects mécaniques mais aussi cognitifs. Par exemple, on sait aujourd'hui que l'utilisation d'un téléphone portable au volant d'une voiture perturbe le conducteur si bien que cette utilisation est aujourd'hui interdite. Un modèle cognitif permettant de mesurer la charge cognitive liée à une action permettrait de choisir le mode d'action minimisant cette charge cognitive.

REFERENCES

1. J. Accot and S. Zhai. Beyond fitts' law: Models for trajectory-based hci tasks. In *Proc. of CHI'97, ACM Press*, pages 295–302, 1997.
2. R. Balakrishnan, T. Baudel, G. Fitzmaurice, and G. Kurtenbach. The rockin' mouse: Integral 3d manipulation on a plane. In *Proc. of CHI'97, ACM Press*, pages 311–318, 1997.
3. P. Barnard. *Interacting Cognitive Subsystems: modelling working memory phenomena within a multi-processor architecture*, pages 298–339. Cambridge University Press, 1999.
4. W. Buxton. Lexical and pragmatic considerations of input structures. *Comp. Graphics*, 17(1):31–37, 1983.
5. S. Card, J. Mackinlay, and G. Robertson. A morphological analysis of the design space of input devices. *ACM Tr. on Information Systems*, 9(2):99–12, Apr.91.
6. D. Cheung, G. Joulie, F. Grillon, J. Fuchet, and J.-Y. Tigli. Wcomp : Rapid application development toolkit for wearable computer based on java. In *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 5, pages 4198–4203, Washington, Oct. 2003.
7. P. Dragicevic and J.-D. Fekete. Input device selection and interaction configuration with icon. In *Proc. of IHM-HCI, Lille, France*, pages 543–448, 2001.
8. E. Dubois, P. Gray, and L. Nigay. Asur++: a design notation for mobile mixed systems. *Journal Interacting With Computers*, 15:497–520, 2003.
9. E. Dubois, D. Scapin, C. Bach, S. Dupuy, B. Mansoux, J.-Y. Tigli, and C. Vachet. Méthodes et outils pour les systèmes mixtes. In *Annexes des actes de la conférence IHM 2004, Namur*, Sept. 2004.
10. P. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *J. of Experimental Psychology*, 46:381–391, 1954.
11. J. Foley and V. Wallace. The art of natural graphics man-machine conversation. *Proceedings of the IEEE*, 62(4):462–471, 1974.
12. J. Foley, V. Wallace, and P. Chan. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Appl.*, 4(11):13–48, 1984.
13. F. Gemperle, N. Ota, and D. Siewiorek. Design of a wearable tactile display. In *Proc. of ISWC'01, Zurich, Switzerland*, page 5, Oct. 2001.
14. S. Greenberg and C. Fitchet. Phidgets: Easy development of physical interfaces through physical widgets. In *Proc. of UIST'01, Florida*, Nov. 2001.
15. K. Hinckley, J. Tullio, R. Pausch, D. Proffitt, and N. Kassell. Usability analysis of 3d rotation techniques. In *Proceedings of UIST'97*, pages 1–10, 1997.
16. R. Jacob and L. Sibert. The perceptual structure of multidimensional input devices selection. In *Proc. of CHI'92, ACM Press*, pages 211–218, 1992.
17. K. Lyons, D. Plaisted, and T. Starner. Expert chording text entry on the twiddler one-handed keyboard. In *Proc. of ISWC'04, Virginia*, pages 94–101, Nov. 2004.
18. K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. Looney. Twiddler typing: One-handed chording text entry for mobile phones. In *Proc. of CHI'04. Vienna, Austria.*, Apr. 2004.
19. N. Plouznikoff and J.-M. Robert. Caractéristiques, enjeux et défis de l'informatique portée. In *Actes de la conférence IHM 2004*, Sept. 2004.
20. R. Chalon and B. T. David. Irvo: an interaction model for designing collaborative mixed reality systems. In *HCI International 2005, Las Vegas, USA*, July 2005.
21. J. Rekimoto. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Proc. of ISWC'01, Zurich, Switzerland*, page 21, Oct. 2001.
22. T. Starner. The challenges of wearable computing: Part 1 and 2. *IEEE Micro*, pages 54–67, july-august 2001.
23. D. Norman. *The Design of Everyday Things*, 1989.