

Interaction à deux souris : Une nouvelle dimension à la multimodalité ?

Stéphane Chatty

Centre d'Études de la Navigation Aérienne

7 avenue Édouard Belin

31055 TOULOUSE CEDEX

+33 62 25 95 42

chatty@dgac.fr

Résumé : Cet article étudie l'interaction multimodale sous l'angle particulier de l'interaction à deux souris. Cette dernière soulève quelques problèmes de réalisation spécifiques, et impose des contraintes nouvelles aux outils de construction d'interfaces graphiques. Mais surtout, elle met en lumière l'existence d'un type de multimodalité particulier, où les interactions entre les modalités se font à faible niveau d'abstraction, et à très faible granularité. Dans la mesure où on retrouve aussi dans cette multimodalité "fine" les caractéristiques telles que le parallélisme ou la synergie, il semble que cela mette en évidence une autre dimension de l'espace des interfaces multimodales : l'échelle ou le grain de l'interaction. Par ailleurs, certains types d'interaction à deux souris suggèrent que la fusion d'événements ne recouvre pas tous les types de combinaison, et qu'il faut aussi introduire la notion de combinaison d'états pour rendre compte de certaines actions.

Mots-clés : Interaction multi-modale, interaction à deux mains, construction d'interfaces.

1. Introduction

Nombre d'études ont porté récemment sur l'interaction multimodale. La plupart d'entre elles combinaient deux modalités : la voix, et l'interaction graphique avec une souris ou un périphérique similaire [Thorisson et al. 92, Nigay & Coutaz 93]. Ces études ont permis entre autres de forger une classification des interfaces multimodales, et d'ébaucher des modèles de construction de telles interfaces. En revanche, assez peu d'études récentes ont porté sur l'interaction à deux mains. Il est vrai que ce type d'interaction peut paraître marginal, dans la mesure où il occupe les deux mains, alors qu'on cherche souvent à utiliser la voix précisément pour libérer une main. Ainsi, Newell [Newell 93] insiste sur l'importance des interfaces pour handicapés, en soulignant que de nombreuses circonstances particulières font d'un utilisateur apparemment normal un handicapé temporaire : pilote aux mains occupées, contrôleur aérien en état de stress, etc. Cependant, cela n'est pas une raison pour ignorer totalement les circonstances où l'utilisateur est entièrement disponible et concentré dans la réalisation d'une tâche unique. Le dessinateur devant sa table en est un exemple, au moins lors de certaines phases de travail. Dans de telles conditions, il peut être intéressant d'utiliser la seconde main. C'est d'ailleurs déjà le cas pour certaines applications. À l'instant où j'écris ces lignes, j'utilise mes deux mains, grâce à un périphérique prévu pour cela : le clavier. Quelques auteurs, et en particulier Buxton [Buxton & Myers 86, Bier et al. 93], suggèrent depuis longtemps l'utilisation des deux mains dans les interfaces graphiques.

Les interfaces graphiques à deux mains soulèvent un certain nombre de questions relevant de l'ergonomie des interfaces. Ces questions ne seront pas abordées dans cet article, et elles sont mentionnées ici pour mémoire. On peut d'abord se demander quels périphériques utiliser. Une solution possible est une souris pour la main dominante, et une boule roulante pour l'autre main. Nous avons choisi au CENA d'utiliser dans un premier temps une seconde souris, pour des raisons matérielles uniquement. Une autre question est celle du rôle de chacune des mains. Quel est l'usage de la main non-dominante, moins précise ? Quelles sont les techniques d'interaction à utiliser ? Il est souvent suggéré d'utiliser la main non-dominante pour des tâches annexes comme le déplacement dans un document ou la sélection d'un outil, tandis que la main droite effectuerait les tâches demandant plus de soin. Ces exemples font agir les deux mains de manière indépendante. D'autres exemples mentionnés dans la suite de cet article combinent les actions des deux mains.

Les interfaces à deux mains posent aussi des questions informatiques. Tout d'abord, leur réalisation met à dure épreuve les outils de construction d'interface. Ne parlons pas des éditeurs d'interfaces : ils sont particulièrement hermétiques à ce type d'extension. Mais les boîtes à outils elles-mêmes demandent des modifications pour supporter la présence d'un second moyen de désignation, et surtout la possibilité d'actions simultanées. Ensuite se posent des problèmes de modélisation. Il est agréable, au moins pour des raisons de similitude technique, de considérer les interfaces à deux souris comme des interfaces multimodales. Mais la symétrie entre les deux canaux, de même que la nature des interactions réalisées, amène des considérations nouvelles. En particulier, certaines interactions trouvent mal leur place dans la classification introduite aux ateliers IHM'91 [IHM91]. Ce sont ces sujets qui vont être abordés dans la suite de cet article. Tout d'abord, j'exposerai les problèmes de réalisation des interfaces à deux souris, en m'appuyant sur une expérience d'ajout d'une souris à une interface à manipulation directe existante. Ensuite, j'aborderai les problèmes de modélisation, et je proposerai d'étendre le modèle d'IHM'91 pour prendre en compte de telles interfaces.

2. Écrire une interface à deux souris

Une première approche pour explorer l'interaction à deux mains consiste à reprendre une application à manipulation directe traditionnelle, et à la modifier pour ajouter une seconde souris. De cette manière, on s'abstrait des problèmes de réalisation déjà connus, pour se concentrer sur ceux qui apparaissent avec la deuxième souris. De plus, en se fixant comme objectif une parfaite symétrie entre les souris, on garantit que la plupart des difficultés techniques vont être identifiées. Enfin, cela permet d'étudier dans quelle mesure une interface conçue pour l'interaction à une main peut être étendue. Par la suite, une fois résolus les problèmes d'intégration, il restera à examiner la manière dont on peut combiner les actions des deux mains, et à imaginer de nouvelles techniques d'interaction.

L'application choisie ici est un exemple utilisé lors de la conception de la boîte à outils X_{TV} [Beaudouin-Lafon et al. 90]: un jeu de réussite où toutes les cartes sont alignées sur l'écran, et où le but est de ranger les cartes par ordre croissant, en les déplaçant successivement vers des emplacements laissés libres. Lorsque cette réussite est réalisée avec de vraies cartes, il est courant que les joueurs, ayant plusieurs coups d'avance en tête, déplacent deux cartes à la fois, la seconde venant occuper la place laissée libre par la première. Cet exemple va donc nous permettre d'examiner les obstacles successifs à la création d'une interface à deux mains autorisant les actions en parallèle. En revanche, d'autres exemples seront nécessaires pour étudier les actions combinant les deux souris.

Matériel D'un point de vue matériel, la connection d'une seconde souris est une opération assez simple, et très peu coûteuse : une souris pour micro-ordinateur du type PC coûte environ 300 francs, et se connecte aisément sur un port série (RS-232) de station de travail. Quant au format des messages qu'elle émet, il est public et relativement simple à décoder.

Cohabitation entre souris Une fois que l'on sait décoder les positions et les clics de la seconde souris, il faut la prendre en compte de la même manière que la première, et gérer les possibles interférences. Les boîtes à outils modernes font toutes reposer leur gestion des entrées sur la notion d'événement : chaque action sur un périphérique se traduit par un événement qui est placé dans une file d'attente, laquelle alimente une boucle de gestion d'événements. Le traitement de chaque événement se fait alors en fonction de son type, voire des conditions dans lesquelles il s'est produit. Si l'on veut pouvoir utiliser les deux souris de la même manière, et cela sans dupliquer la gestion des événements, la solution la plus simple consiste à insérer dans la file d'attente les événements en provenance de la seconde souris. Cela suppose d'une part que l'on puisse créer de nouveaux événements, et d'autre part que l'on puisse intégrer la procédure de lecture des messages de la souris dans la boucle de lecture des événements. Des boîtes à outils telles que la X Toolkit, InterViews [Linton et al. 89] ou X_{TV} permettent cela, avec plus ou moins de facilité selon le cas.

Un problème moins facile à résoudre est celui qui consiste à distinguer les souris. On voudra en général que les événements issus des deux souris soient traités de la même manière : cliquer sur un bouton ou un menu doit donner le même résultat, qu'on le fasse de la main droite ou de la main gauche. Cela suggère que les événements doivent être de même type. Mais cela ne signifie pas qu'ils doivent être impossibles à distinguer. En effet, il faut éviter les interférences qui peuvent se produire lors de l'usage simultané des deux mains : par exemple, il faut éviter qu'enfoncer un bouton de la souris gauche et relâcher un bouton de la souris droite soit compris comme un clic. Il faut donc à la fois assurer que les événements puissent suivre le même chemin, et pouvoir les distinguer dans certaines situations. Cela suppose une souplesse dans le mécanisme de sélection d'événements nettement supérieure à celle que l'on connaît aujourd'hui. Même X_{TV} , dont le mécanisme de gestion des événements est particulièrement développé, ne permet pas cela de manière facile. C'est donc un sujet qui devra faire l'objet de précautions particulières dans la conception de futures boîtes à outils.

Plusieurs actions en parallèle L'intégration de la seconde souris et la possibilité de sélectionner les événements qu'elle émet permettent que les deux souris tolèrent leur présence mutuelle. Cela ne garantit pas qu'il soit possible de les utiliser en parallèle. En effet, la plupart des actions réalisées avec une souris mettent en œuvre des mécanismes plus complexes que la simple gestion d'un événement : en général, une action est composée d'une suite d'événements, elle est associée à un ou plusieurs objets de l'application, et elle donne lieu à un retour visuel. Toutes ces informations doivent être mémorisées et mises à jour pendant la durée de l'action. En l'absence de mécanisme spécialisé, elles sont généralement manipulées sous la forme de variables globales (c'est la notion de sélection courante), selon l'hypothèse qu'une seule action est possible à un moment donné. Or cette hypothèse est incompatible avec l'utilisation de deux souris. Il faut donc disposer d'un mécanisme explicite de gestion des actions longues, de manière à ce que plusieurs actions soient possibles en parallèle. Il semble que seul X_{TV} dispose d'un tel mécanisme.

Ainsi, dans le cas du jeu de réussite qui a servi de base aux premières expériences, la

plupart des modifications nécessaires étaient liées à ce problème des actions en parallèle. Pour chaque déplacement de carte, la carte en cours de déplacement est mise en valeur, et un “fantôme” suit le curseur de la souris jusqu’à la fin de l’action. Jusqu’à présent, la carte à mettre en valeur et le fantôme étaient mémorisés en un endroit unique du jeu. Désormais, ils sont associés à un objet *Action*, dont la durée de vie est celle de l’action en cours. C’est ainsi qu’on peut désormais déplacer deux cartes en parallèle, comme on le fait avec de vraies cartes. Il est toutefois intéressant de noter que le retour visuel choisi dans l’interface à une souris n’est plus tout à fait adapté pour l’interface à deux souris. En effet, si la carte déplacée ne suit pas le curseur durant l’action, on ne visualise pas le fait qu’elle libère son emplacement précédent. Si cela ne posait pas de problème important lorsque les actions étaient sérialisées, l’expérience montre que cela perturbe l’utilisateur lors de certaines actions en parallèle.

Si l’on pousse plus loin ce dernier raisonnement, il faut déplacer les cartes et non leur fantôme. On découvre alors que le noyau fonctionnel est lui aussi affecté par l’interaction parallèle. En effet, celui-ci doit vérifier que chaque mouvement est légal. Souvent, un coup n’est possible que grâce aux modifications apportées par le coup précédent. Or dans le cas de l’interaction à deux mains, l’ordre des actions est pour le moins approximatif : avec de véritables cartes, il arrive qu’un mouvement soit fait une fraction de seconde avant celui qui le rend légal. Ici, ajouter cette tolérance sur l’ordre des actions nécessite une intervention majeure dans le noyau fonctionnel. C’est ainsi qu’une modification qui paraissait limitée à l’interface peut avoir un impact sur l’application toute entière.

Combinaisons Nous avons jusqu’à présent examiné les problèmes liés aux actions parallèles des deux souris. On peut construire un certain nombre d’interfaces sur la base de ces actions parallèles. Ainsi, un outil de dessin du type Mac Draw peut en bénéficier : tandis qu’une main dessine, l’autre déplace la feuille ou sélectionne les outils. Mais on peut aussi imaginer des techniques d’interaction combinant les deux mains.

Une première technique consiste à combiner les mouvements des deux souris : par exemple, un objet verrouillé pourrait ne se déplacer que si les deux souris tirent dans la même direction ; ou encore, un élément d’un polygone pourrait se briser si les deux souris l’étirent brutalement dans des directions opposées. De telles interactions sont plus difficiles à réaliser qu’il n’y paraît. En effet, pour comparer ou combiner des mouvements, il faut pouvoir manipuler des événements “simultanés” : un mouvement de la souris gauche, puis de la souris droite une demi-seconde plus tard, ne font pas un mouvement coordonné. Or, la notion de simultanéité est délicate à établir : la seule solution consiste à retarder le traitement de chaque événement pendant un délai réduit. Si pendant ce délai un événement est reçu en provenance de l’autre souris, les deux événements sont combinés. Si rien ne se produit avant l’expiration du délai, l’événement est traité de manière normale. On retrouve là une notion proche du creuset introduit par L. Nigay et J. Coutaz, mais le temps tient ici une place déterminante.

Une autre technique d’interaction consiste à utiliser la main non dominante de manière plus statique, pour “tenir” des objets. Ainsi, nous avons expérimenté au CENA une technique de manipulation à deux mains de polygones : lorsqu’on saisit une arête avec une seule souris, on ne peut que la déplacer. Pour la modifier, il faut maintenir une extrémité en place avec une souris, et manipuler l’autre extrémité avec l’autre souris. Dans ce dernier cas, il ne s’agit plus de combiner des événements, mais de modifier le traitement d’une souris en fonction de l’état de l’autre.

Ces deux techniques d’interaction ne sont sans doute que deux exemples parmi d’autres combinant les deux souris. Elles mettent cependant en évidence le besoin d’outils logiciels

pour combiner des flots d'événements et des états de périphériques. Des modèles et des techniques ont déjà été mis au point pour des interfaces multimodales du type voix+souris. Nous allons examiner leur adéquation à cette situation.

3. Vers une multimodalité fine ?

Au cours des journées IHM'91, a été mis au point un modèle de classification des interfaces multimodales. Ce modèle est depuis lors utilisé par de nombreux auteurs français [Faure & Julia 93, Bellik & Teil 92], principalement pour des applications mettant en jeu la voix. Dans cette classification, les interfaces combinant deux souris trouvent naturellement leur place dans la catégorie des interfaces synergiques : les actions sont à la fois parallèles et combinées. Cependant, retrouver ces interfaces dans la même catégorie que celles du type "Mets-ça ici" n'est pas tout à fait satisfaisant. En effet, les techniques pour les mettre en œuvre sont substantiellement différentes. Dans le cas du "Mets-ça ici", on manipule et on fusionne des entités chargées de sens, en faible nombre. Ce n'est pas l'événement de bas niveau de la souris qui est pris en compte, mais la désignation d'un objet ou d'une position. Ce sont en fait des commandes qui sont fusionnées.

A l'opposé, le "drag combiné" de deux souris fusionne au plus bas niveau d'abstraction des flots d'événements, au rythme de plusieurs dizaines par seconde. Alors que la fusion de commandes du "Mets-ça ici" intervient au niveau du dialogue, celle-ci intervient au niveau de l'interaction avec la couche de présentation de l'interface. Par ailleurs, la notion de temps a ici un rôle encore plus essentiel que dans le "Mets-ça ici". La quasi-simultanéité est recherchée à un grand niveau de précision, comme s'il s'agissait de combiner des forces physiques. On voit donc apparaître dans les interfaces multimodales, comme dans les interfaces graphiques traditionnelles, une distinction entre la description du dialogue et de l'enchaînement des commandes, et celle de la présentation et de l'interaction "physique". Cette distinction se situe autant au niveau sémantique qu'au niveau temporel, et va de pair avec l'utilisation de techniques différentes. Si on ajoute cette distinction dans le modèle d'IHM'91, cela introduit une 3^{ème} dimension que l'on peut nommer "grain de l'interaction", et qui comporte à ses deux extrémités le dialogue et l'interaction physique.

4. Fusion d'événements et combinaisons d'états

La technique prédominante dans le domaine du dialogue multimodal est celle de la fusion. Des modèles comme celui du creuset ont été développés pour la faciliter : le creuset recueille les informations contenues dans une commande en attente de fusion, puis fabrique une commande complète lorsque les informations complémentaires lui parviennent [Nigay & Coutaz 93]. Si l'on transpose cela à l'interaction physique, on retrouve une notion similaire à celle des modules de Whizz [Chatty 92], qui filtrent et combinent des flots de données. La dimension temporelle, essentielle pour l'interaction physique, peut être introduite grâce à des modules spéciaux qui ne combinent leurs entrées que sous certaines contraintes de temps. De tels modules sont actuellement ajoutés à Whizz pour faciliter la construction d'interfaces graphiques à deux souris. Cependant, cette technique ne répond pas à tous les besoins. L'exemple de la déformation d'un segment, en tenant une extrémité et en tirant l'autre, ne fait pas appel à la combinaison de flots de données : c'est l'état d'une souris qui influence le traitement des données émises par l'autre. Cette situation est à rapprocher de la notion de mode dans les dialogues, à une échelle de temps réduite. Pour rendre compte de ces modes engendrés par une modalité, il faudra donc

que les modèles de construction d'interfaces multimodales prennent en compte la notion d'état en plus de celle d'événement ou de commande.

5. Conclusion

Bien que l'interaction à deux mains soit toujours considérée comme marginale, des études prouvent son intérêt. Les expériences décrites ici montrent que leur réalisation et leur modélisation posent un certain nombre de problèmes. Certains de ces problèmes se rattachent au domaine de l'interaction et des boîtes à outils graphiques, tandis que d'autres sont liés à l'interaction multi-modale. Dans les deux cas, les modèles précédemment développés nécessitent des aménagements. En particulier, les interfaces à deux mains, plus encore que les interfaces à manipulation directe traditionnelles, mettent en évidence l'intérêt de considérer l'interface comme un monde composé d'objets obéissant à des lois quasi-physiques, et avec lesquels on interagit. Il reste toutefois à construire les modèles adéquats pour décrire de telles interfaces et offrir des outils pour les réaliser.

Bibliographie

- [Beaudouin-Lafon et al. 90] Michel Beaudouin-Lafon, Yves Berteaud, et Stéphane Chatty. Créer des applications à manipulation directe avec X_{TV} . *IHM'90, Biarritz, France*, 1990.
- [Bellik & Teil 92] Yacine Bellik et Daniel Teil. Les types de multimodalités. *IHM'92 - Quatrième journées sur l'ingénierie des interfaces homme-machine*, 1992.
- [Bier et al. 93] Eric Bier, Maureen Stone, William Buxton, et Tony DeRose. Toolglass and magic lenses : the see-through interface. *Proceedings of the ACM SIGGRAPH*. Addison-Wesley, 1993.
- [Buxton & Myers 86] William Buxton et Brad Myers. A study in two-handed input. *Proceedings of the ACM SIGCHI*, pages 321–326. Addison-Wesley, 1986.
- [Chatty 92] Stéphane Chatty. Defining the behaviour of animated interfaces. *Engineering for Human-Computer Interaction*, pages 95–109. North-Holland, Août 1992.
- [Faure & Julia 93] C. Faure et L. Julia. Interaction homme-machine par la parole et le geste pour l'édition de documents : TAPAGE. *L'interface des mondes réels et virtuels, Montpellier*. EC2, 1993.
- [IHM91] Chapitre 1 - multimodalité. *Actes des journées IHM'91*, 1991.
- [Linton et al. 89] Marc A. Linton, John M. Vlissides, et Paul R. Calder. Composing user interfaces with InterViews. *IEEE Computer*, pages 8–22, Février 1989.
- [Newell 93] Allan Newell. CHI for everyone. Opening address at INTERCHI'93, 1993.
- [Nigay & Coutaz 93] Laurence Nigay et Joëlle Coutaz. A design space for multimodal systems: concurrent processing and data fusion. *Proceedings of the ACM SIGCHI*, pages 172–178. Addison-Wesley, 1993.
- [Thorisson et al. 92] K. Thorisson, D. Koons, et R. Bolt. Multi-modal natural dialogue. *Proceedings of the ACM SIGCHI*. Addison-Wesley, 1992.