Interaction instrumentale : de la manipulation directe à la réalité augmentée

Michel Beaudouin-Lafon
Laboratoire de Recherche en Informatique - URA 410 du CNRS
LRI - Bâtiment 490 - Université de Paris-Sud
91405 Orsay Cedex, France
mbl@lri.fr

RÉSUMÉ

Cet article présente un nouveau modèle d'interaction : l'interaction instrumentale. Nous analysons le modèle d'interaction sous-jacent à la majorité des applications graphiques actuelles, et nous montrons qu'il conduit à une complexité croissante de ces applications. Le modèle que nous proposons est fondé sur les notions d'objet du domaine, d'instrument d'interaction et de réification. Pour valider ce modèle nous montrons qu'il permet de représenter les interfaces actuelles tout en mettant en évidence leurs défauts et leurs limitations. Puis nous montrons qu'il représente également une large gamme de nouvelles techniques d'interaction comme l'interaction bimanuelle ou la réalité augmentée.

MOTS-CLÉS: Interaction instrumentale, Manipulation directe, Modèles d'interaction, Réalité augmentée.

INTRODUCTION

Depuis l'introduction des principes de la manipulation directe par Ben Shneiderman en 1983 [24] et les travaux des chercheurs de Xerox PARC sur le Star [25], les recherches dans le domaine de l'interaction graphique ont surtout porté sur des techniques d'interaction spécifiques et des outils d'aide au développement d'applications interactives. Peu de travaux ont abordé la définition de nouveaux principes et modèles généraux pour l'interaction graphique (citons cependant en France les travaux de Nanard [22] et Baudel [4]). Alors que le nombre d'applications commerciales et le nombre d'utilisateurs ont explosé, les interfaces graphiques sont restées figées sur les principes des interfaces WIMP (Windows, Icons, Menus, Pointing) datant de plus de 15 ans.

L'objet de cet article est d'introduire un nouveau modèle d'interaction générique pour les applications graphiques. Nous commençons par l'analyse et l'évaluation du modèle d'interaction sous-jacent aux applications graphiques actuelles et mettons en évidence les limitations induites par ce modèle. Puis nous présentons notre modèle *d'interaction instrumentale.* Pour valider ce modèle nous montrons qu'il permet de représenter les interfaces actuelles et de caractériser certains de leurs défauts et de leurs limitations. Puis nous montrons qu'il permet également de décrire une large gamme de nouvelles techniques d'interaction comme l'interaction bimanuelle ou la réalité augmentée.

ETAT DE L'ART DES APPLICATIONS

La très grande majorité des applications actuelles sont organisées autour des éléments d'interaction suivants : des menus déroulants, des boîtes de dialogue, des fenê d'application et des palettes ou barres d'outils.

Les menus déroulants sont accessibles par une barre menus et donnent accès à un ensemble de commandes e sous-menus. Certaines commandes ont un effet immé d'autres font apparaître une boîte de dialogue qui permet spécifier les paramètres nécessaires à l'exécution de commande. Les valeurs de ces paramètres sont spécifiées g à un ensemble standard d'interacteurs : champs texte, lis cases à cocher, boutons radio, etc. La plupart des boîte dialogue sont modales : le reste de l'interface est inacces pendant que la boîte de dialogue est affichée et l'effet d modification des paramètres n'est perceptible qu'une fois cel ci validées (bouton "OK") ou par une demande explicit prévisualisation (bouton "Apply").

Les fenêtres d'application présentent les données du dom de l'application dans des documents qui peuvent être enregis sur disque. Ces fenêtres sont souvent bordées de barres défilement pour contrôler la vue sur le document et parfoi barres d'outils, de repères dans les marges, etc. Le contenu fenêtres d'application est accessible par action directe souris. Il est notamment possible de spécifier la *sélect* c'est-à-dire les objets de l'application qui sont affectés par commandes activées par les menus et boîtes de dialogue. sélection apparaît dans la fenêtre sous une forme graphi changement de couleur, poignées, voire animation.

Les palettes flottantes et les barres d'outils ont trois rô Elles peuvent donner un accès direct à certaines comman évitant la sélection dans les menus ; elles contiennent alor simples boutons, souvent iconiques. Elles servent égalem activer des modes temporels qui déterminent l'effet des acti de la souris dans la fenêtre d'application : mode sélection, m création d'objet, etc. Enfin elles peuvent contenir des cha de saisie, à l'instar les boîtes de dialogue. Nous les appel alors *boîtes de propriétés*, car l'entrée d'une valeur dans champ modifie immédiatement les attributs de l'o sélectionné et visualise l'effet de cette modification san validation imposée par les boîtes de dialogue.

Les techniques d'interaction utilisées dans ces applicati reposent sur l'usage de la souris et du clavier. Le clavier utilisé à la fois pour l'entrée de texte, pour l'activation commandes (raccourcis-claviers, touches de fonction), et contrôler les actions effectuées avec la souris. Ce dernier us permet une forme primitive d'interaction bimanuelle laquelle la main dominante contrôle le tracé à la souris e main non-dominante contrôle la fonction du tracé par appui

Actes 9èmes Journées IHM, Poitiers, septembre 1997, Cépaduès Editions

les touches telles que Shift, Alt, Option, etc. La souris sert au pointage, à la sélection et au tracé [15]. L'interaction de pointage+sélection (clic) permet d'activer des commandes (sélection dans les menus), d'entrer des données dans les boîtes de dialogues (sélection des champs et des valeurs) et de définir les objets sélectionnés. L'interaction de pointage+tracé est utilisée pour deux catégories de fonctions : le "drag-and-drop" qui permet de déclencher des commandes en amenant un objet sur un autre ; le cliquer-tirer qui permet le déplacement, la déformation et la transformation d'objets par action directe sur un objet ou sur ses poignées de sélection.

Évaluation de quelques applications

Afin d'évaluer ce modèle d'interaction, nous avons étudié 10 applications sur Apple Macintosh. Les applications suivantes ont été choisies car elles sont très largement répandues et qu'elles concernent des domaines d'applications différents (traitement de texte, tableur, présentation, retouche photographique, PAO, édition de schémas, modelage 3D) :

- Microsoft Word 5.1 et 6.0 (W5 et W6) ;
- Microsoft Excel 4.0 et 5.0 (E4 et E5) ;
- Aldus Persuasion 3.1 (Pe3) ;
- Adobe Photoshop 2.5 et 4.0 (P2 et P4) ;
- Quark XPress 3.3 (X3) ;
- Deneba Canvas 3.0 (C3) ;
- Metatools Bryce 2.1.

Critère	W6	E5	Pe3	P4	X3	C3	**Moy.**	σ
#menus	8	8	7	8	7	8	**7,7**	0,5
#cmds	106	84	97	111	99	74	**95,2**	13,8
#dlog	69	44	20	27	40	21	**36,8**	18,6
#ssmenu	1	15	27	26	13	22	**17,3**	9,8
#sscmds	3	58	73	82	65	121	**67,0**	38,4
#ssdlog	0	20	20	40	10	28	**19,7**	13,9
Tcmds	108	127	143	167	151	173	**144,8**	24,5
Tdlogs	69	64	40	67	50	49	**56,5**	11,8
Cmds/M	13,3	10,5	13,9	13,9	14,1	9,3	**12,5**	2,1
Cmds/SM	3,0	3,9	2,7	3,2	5,0	5,5	**3,9**	1,1
#palettes	9	13	5	11	6	6	**8,3**	3,2
#outils	125	106	54	77	68	60	**81,7**	28,0
#prefs	12	10	1	8	5	11	**7,8**	4,2
#options	113	76	11	51	82	27	**60,0**	37,7
macros	oui	oui	non	oui	non	oui		

Tableau 1 : évaluation de 6 applications

Les critères qui ont été choisis sont indépendants de la nature de l'application et sont représentatifs de la complexité de l'interface telle qu'elle peut être perçue visuellement lors de l'utilisation de l'application. Bien entendu, ils ne préjugent pas de l'usage effectif qui peut être fait d'une application. Dans le tableau 1, nous avons établi les moyennes (Moy) et les écarts-type (σ) pour 6 applications. Dans le tableau 2 nous avons comparé l'évolution des critères pour des versions différentes de 3 applications. Les paramètres mesurés sont les suivants :

#menus	Nombre de menus de la barre de menus ;
#cmds	Nombre de commandes accessibles directement par ces menus ;
#dlogs	Nombre de ces commandes qui conduisent à une boîte de dialogue ;
#ssmenus	Nombre de commandes qui conduisent à un sous-menu ;
#sscmds	Nombre de commandes des sous-menus ;
#ssdlogs	Nombre de commandes des sous-menus qui conduisent à une boîte de dialogue.
Tcmds	Nombre total de commandes : #cmds - #ssmenus + #sscmds ;
Tdlogs	Nombre total de boîtes de dialogues : #dlogs + #ssdlogs ;
Cmds/M	Nombre moyen de commandes par menu de barre de menu : #cmds / #menus ;
Cmds/SM	Nombre moyen de commandes par sous-mei #sscmds / #ssmenu.
#palettes	Nombre total de palettes et barres d'outils ;
#outils	Nombre total d'interacteurs (boutons, menus etc.) accessibles dans ces palettes ;
#prefs	Nombre total de pages de préférences qui permettent de personnaliser l'interface ;
#options	Nombre total d'options que l'on peut spécifi dans ces pages de préférences ;
macros	Possibilité de définir des macros.

Critère	E4	E5	%	W5	W6	%	P2	P4
#menus	8	8	0%	8	8	0%	7	8
#cmds	93	84	-10%	107	106	-1%	78	11
#dlog	60	44	-27%	55	69	+25%	21	2
#ssmenu	0	15	+∞	0	1	+∞	19	2
#sscmds	0	58	+∞	0	3	+∞	56	8
#ssdlog	0	20	+∞	0	0	+∞	39	4
Tcmds	93	127	+37%	107	108	+1%	115	16
Tdlogs	60	64	+7%	55	69	+25%	60	6
Cmds/M	11,6	10,5	-10%	13,4	13,3	-1%	11,1	13,
Cmds/SM	0,0	3,9	+∞	0,0	3,0	+∞	2,9	3,
#palettes	8	13	+63%	3	9	+200%	6	1
#outils	108	106	-2%	63	125	+98%	49	7
#prefs	0	10	+∞	10	12	+20%	9	8
#options	0	76	+∞	52	113	+117%	58	5
macros	oui	oui		non	oui		non	ou

Tableau 2 : comparaison de versions successives

Ces tableaux appellent plusieurs remarques. D'une part nombre total de commandes, de boîtes de dialogues et d'ou accessibles par les palettes est comparable entre les différe applications (lignes Tcmds et Tdlogs). Par con l'organisation des commandes en menus et sous-menus plus variable. Le nombre de menus dans la barre de me étant limité par la taille de l'écran, il faut choisir entre un grand nombre de commandes dans chaque menu et un nom plus important de sous-menus. On remarque également qu proportion de commandes qui conduisent à une boîte dialogue est considérable : entre un 1/3 et 2/3. D'autre part, effort est fait pour permettre la personnalisation applications : configuration des menus, palettes et ba d'outils, macros, etc. Enfin, les versions successives même logiciel font systématiquement apparaître une augmentation importante du nombre de commandes et/ou de boîtes de dialogue, et du nombre d'outils dans les palettes. Pour résumer cette évaluation, on peut dire qu'une application "standard" offre 150 commandes dans ses menus, 60 boîtes de dialogues, et 80 outils.

Evaluation qualitative

Pour poursuivre notre évaluation sur le plan qualitatif, nous reprenons les principes de la manipulation directe tels qu'ils ont été énoncés par Shneiderman [24] :

M1 : Représentation continue des objets d'intérêt ;
M2 : Actions physiques sur les objets plutôt que syntaxe complexe ;
M3 : Opérations rapides, incrémentales et réversibles dont l'effet sur les objets d'intérêt est immédiatement visible ;
M4 : Approche en couches ou en spirale qui permet une utilisation avec un minimum de connaissances.

Nous allons montrer que ces applications ne suivent pas les principes de la manipulation directe, ou plus exactement qu'elles ne les interprètent pas correctement.

Ainsi, les objets d'intérêt (M1) sont limités aux objets de l'application. Ces derniers sont bien représentés de façon permanente (bien que souvent occultés par les palettes et boîtes de dialogue), mais l'interface fait généralement intervenir d'autres objets qui, même s'ils ne font pas directement partie du domaine de l'application, sont indispensables à sa mise en œuvre. Ainsi les feuilles de style de Word ou XPress, les couches de Photoshop ou Canvas, les brosses et les filtres de Photoshop pour ne citer que quelques exemples, sont des objets avec lesquels l'utilisateur doit interagir constamment pour utiliser l'application. Il est donc raisonnable de penser que ces objets existent dans le modèle mental de l'utilisateur et qu'à ce titre ils devraient avoir le statut d'objet d'intérêt : ils devraient être représentables de façon continue dans des fenêtres similaires aux fenêtres d'application, voire enregistrables dans des fichiers. C'est d'ailleurs une évolution que l'on observe dans certaines versions successives du même logiciel, comme par exemple Photoshop.

En ce qui concerne les actions physiques (M2), on peut noter la pauvreté des moyens physiques d'interaction, limités dans l'immense majorité des cas à une souris et un clavier. Certains logiciels de dessin peuvent être pilotés par des tablettes graphiques sensibles à la pression, mais aucun des éléments classiques des interfaces ne sont prévus pour prendre en compte cette dimension, ni même plus d'un seul périphérique de désignation [12]. Il en résulte que le vocabulaire d'entrée des applications est limité à des actions de pointage, de sélection et de tracé, ces dernières n'interprétant la plupart du temps que les positions de départ et d'arrivée et ignorant la trajectoire intermédiaire. Pour pallier cette pauvreté du vocabulaire d'entrée, les applications utilisent des intermédiaires comme les menus et boîtes de dialogue. Dans certains cas, l'activation d'une commande requiert la sélection dans une hiérarchie de menus et le remplissage d'une cascade de boîtes de dialogues : au lieu d'actions physiques sur les objets, l'utilisateur doit se conformer à une syntaxe de saisie imposée par le système, violant ainsi le principe M2. Aussi, bien que les applications étudiées soient dites à manipulation directe, une grande partie des actions de l'utilisateur opèrent en réalité *indirectement* les objets d'intérêt. Dans la suite, nous parlerons *manipulation directe* lorsqu'il y a action directe sur les ob de l'application par l'intermédiaire des périphériques localisation (ici la souris), et de *manipulation indirecte* pou sélection des commandes dans les menus, l'entrée de donn dans les boîtes de dialogue et de propriétés, l'utilisation palettes et barres d'outils.

L'utilisation de la manipulation indirecte prése l'inconvénient que les opérations ne sont ni rapides, incrémentales (M3) : la navigation dans les menus et sc menus et l'usage de boîtes de dialogues à onglets ou en casc diminuent considérablement l'efficacité de l'interaction. En un grand nombre d'actions sont destinées à la sélection paramètres et à la syntaxe d'activation des commandes pl qu'à la spécification des valeurs. La saisie des valeurs également inefficace à cause de l'usage d'un nombre réduit types d'interacteurs. Ainsi, une valeur numérique est souv entrée par saisie textuelle, et des tâches de saisie intégrales [comme la saisie d'une taille ou d'une position sont souv artificiellement séparées en deux saisies pour les valeurs e et en Y. Enfin, la saisie de valeur est rarement incrémentale on ne peut pas, en général, juger de l'effet de la va couramment sélectionnée sur l'exécution de la commande fur et à mesure de cette spécification. Le style d'interaction donc largement conversationnel, similaire à l'entrée d'une li de commande textuelle dont l'effet n'apparaît que lors de l'ap sur la touche RETURN.

En ce qui concerne la facilité d'apprentissage (M4), la plu des applications sont utilisables avec un minimum connaissance, et ceci d'autant plus qu'elles sont conformes modèle que nous avons décrit. Cependant, il reste difficile progresser en découvrant de nouvelles fonctionnalités de fa exploratoire. Bien que les interfaces soient configurables, il a pas en général de pré-configuration qui permettrait de cho entre plusieurs niveaux d'utilisation, et le grand nom d'options et la complexité de l'interface correspond réservent la personnalisation aux utilisateurs expe L'utilisateur novice est donc confronté d'emblée à l'ensembl l'interface et doit extraire de cette complexité les éléments c reconnaît ou qui sont suffisamment évocateurs.

En conclusion, les applications graphiques interacti actuelles présentent une grande complexité visuelle de interface et une faible efficacité de l'interaction. Celle résultent respectivement de la complexité du modèle concep de ces applications et de la pauvreté des techniques d'interac utilisées, en particulier l'usage de la manipulation indirecte.

L'INTERACTION INSTRUMENTALE

Le modèle d'interaction que nous proposons a pour but définir un espace de conception pour l'interaction graphique facilite la définition de nouvelles techniques d'interac permettant de résoudre les problèmes tels que ceux identifiés dessus. Ce modèle englobe le modèle classique décrit au dé de cet article. En particulier, plutôt que d'opposer manipula directe et manipulation indirecte, il décrit un continuum al de l'une à l'autre.

Le modèle est fondé sur la notion d'*instrument d'interaction* jouant le rôle de médiateur entre les actions de l'utilisateur et les objets de l'application. Le terme d'instrument fait bien sûr référence à l'instrument de musique, médiateur privilégié entre l'artiste et l'œuvre, mais doit également s'entendre dans le sens du dictionnaire (Petit Robert) : "Objet fabriqué servant à exécuter qqch., à faire une opération (Instrument est plus général et moins concret que outil; désigne des objets plus simples que appareil, machine)." Pour présenter ce modèle, nous définissons d'abord les objets manipulés par les instruments, puis les instruments eux-mêmes et enfin le principe de réification permettant de créer de nouveaux objets et de nouveaux instruments.

Les objets

Les objets manipulés par les instruments sont en premier lieu les objets du domaine de l'application. Dans un traitement de texte, ce sont les caractères, mots, phrases, paragraphes ; dans un tableur les cellules et feuilles de calcul ; dans un logiciel de dessin les objets graphiques qui composent l'image. Chaque objet est constitué d'un ensemble d'attributs, qui sont des valeurs simples ou des objets complexes. Par exemple, le matériau d'un élément d'une scène 3D est un objet qui peut être édité, copié et sauvé au même titre qu'un élément de la scène. Le principe de réification, décrit plus loin, permet de définir d'autres objets nécessaires à l'interaction, comme les feuilles de styles d'un traitement de texte. Enfin, les instruments sont eux-mêmes des objets, ce qui permet d'éditer leurs caractéristiques, voire d'en créer dynamiquement de nouveaux.

Chaque objet a une ou plusieurs représentations. Dans la suite nous ne considérons que les représentations graphiques. Chaque représentation correspond à une vue sur l'objet et présente un sous-ensemble de ses attributs. Ainsi, un éditeur de texte peut proposer un affichage en mode plan ou en mode page, un modeleur 3D une vue fil-de-fer ou une vue solide, etc. Dans le Finder du Macintosh, l'icone d'un fichier et la boîte d'informations qui affiche sa taille, sa date de création, etc. sont deux vues sur le même objet, à savoir le fichier.

Les instruments

Toute création, modification et destruction d'un objet est réalisée par l'intermédiaire d'un instrument. Un exemple simple d'instrument est une barre de défilement manipulée par l'intermédiaire d'une souris. L'action de la souris sur la barre contrôle l'attribut position (en X ou en Y selon l'orientation de la barre) de l'objet document lié à la barre. Les menus, les boîtes de dialogues et de propriétés, les palettes et les poignées de manipulation sont également des instruments puisqu'ils permettent d'agir de façon plus ou moins directe sur des objets.

Un instrument est composé d'une partie physique et d'une partie logique. La partie physique comprend les transducteurs d'entrée-sortie utilisés par l'instrument, en entrée pour capter l'action physique de l'utilisateur et en sortie pour lui présenter un retour d'information. En général, il s'agit de la souris et du clavier en entrée, et de l'écran en sortie. On peut cependant imaginer (et souhaiter) une plus grande palette de périphériques d'entrée et un retour tactile et/ou auditif de l'information. La partie logique de l'instrument comprend en entrée la méthode de transformation des actions de l'utilisateur sur l'instrument logique et en sortie la représentation de l'instrument. Par exemple, une barre de défilement verticale interprète seulem les mouvements verticaux de la souris et sa représentation constituée d'un ascenseur et de deux flèches.

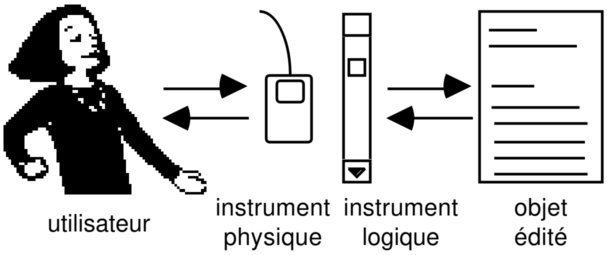

Fig. 1 : l'instrument, médiateur entre l'utilisateur et l'objet

L'association entre instrument logique et instrum physique peut être explicite ou implicite. Elle est explicite exemple lorsque l'on pointe une barre de défilement ou lors l'on active une boîte de dialogue ; elle est implicite exemple dans le cas de la roulette de la souris IntelliMouse Microsoft, qui est assignée à la barre de défilement verticale la fenêtre courante. L'association explicite rend l'interact moins efficace puisqu'elle impose une action de sélection l'instrument logique par l'utilisateur. L'association implicite son côté peut conduire à un grand nombre de périphériq physiques, comme dans le cas des consoles de mixage au qui comportent couramment plusieurs centaines potentiomètres. La combinaison des deux types d'associat permet d'établir un compromis et d'adapter l'interface périphériques disponibles.

L'interaction instrumentale implique deux nive d'interaction (fig. 1) : l'interaction entre l'utilisateur l'instrument, et l'interaction entre l'instrument et l'objet éd L'instrument joue ainsi le rôle de *médiateur* entre l'utilisat et l'objet édité. Les retours d'information sont à trois niveau retour de l'action sur l'instrument physique, retour visuel vi représentation de l'instrument logique, et retour de l'effet l'interaction sur l'objet édité. Dans le cas de la barre défilement manipulée à la souris, le retour d'information su souris est proprioceptif (position de la souris), le ret d'information de l'instrument est donné par la position l'ascenseur, et le retour de l'effet de l'interaction sur l'objet é est le défilement du document. Des dispositifs à retour d'ef comme le clavier rétro-actif modulaire [9] permettraient retour d'information plus riche sur le périphérique d'entrée exploitant le sens tactile et kinesthésique.

La réification

Dans la plupart des applications, l'interface ne peut organisée autour des seuls objets du domaine et instruments. Des concepts additionnels doivent être introd pour structurer l'interface, pour simplifier l'interaction ou p rendre le modèle plus puissant. Un exemple commun dans traitements de texte consiste à grouper des attributs des ob pour définir des "styles" qui peuvent être partagés par plusie objets : en modifiant un attribut d'un style, on mod plusieurs objets simultanément. Selon notre modèle les sty sont des objets, éditables dans leurs propres fenêt enregistrables dans des fichiers, et nécessitant leurs prop instruments d'édition. Un autre exemple commun est la notion de palette d'outils : les instruments sont considérés comme des objets, sélectionnables par un nouvel instrument, la palette.

Nous appelons l'opération qui permet d'identifier les styles ou les instruments d'une palette comme des objets une *réification* : elle transforme un concept en objet. La réification permet une économie de moyens : le modèle de base reste celui des objets et des instruments, la réification servant seulement à définir de nouveaux objets. De plus, la réification permet de définir un ensemble de commandes génériques s'appliquant à diverses catégories d'objets. Ainsi, lorsqu'un concept est réifié, il devient possible d'éditer ses attributs, de le dupliquer, de le stocker, etc.

De nombreuses applications bénéficieraient de l'usage de la réification. Ainsi, les styles de Word n'ont pas réellement un statut d'objet : ils ne sont éditables que dans des boîtes de dialogue modales, et ne peuvent pas apparaître de façon permanente (à part dans le menu des styles) ; on peut cependant les stocker dans des fichiers. Dans Photoshop, les filtres sont des instruments, accessibles uniquement par l'intermédiaire de commandes et de boîtes de dialogue modales. Pourtant, les filtres sont centraux dans cette application de retouche photographique. En réifiant les filtres, il serait possible de créer des configurations prêtes à l'emploi, sans avoir besoin de spécifier les paramètres à chaque utilisation.

Evaluation des interfaces instrumentales

Plusieurs critères ont été mis en avant pour évaluer la "qualité" de l'interaction. Norman [23] a intoduit les distances d'exécution et d'évaluation dans sa théorie de l'action ; Bastien et Scapin [3] ont défini un ensemble de critères ergonomiques ; Baudel [4] définit les notions d'interaction efficace (utilisation optimale de la bande passante utilisateur-système informatique), naturelle (correspondance entre modèle mental et modèle d'interaction), conforme et complète (correspondance entre les actions possibles et les comportements des objets informatiques). Nous proposons d'opérationnaliser certains de ces critères en définissant les degrés d'indirection et d'intégration d'un instrument.

Le *degré d'indirection* d'un instrument se mesure par les décalages spatiaux et temporels qu'il met en jeu. Le décalage spatial entre le périphérique logique et le périphérique physique est pratiquement inévitable, sauf dans le cas d'un écran tactile. Il ne semble pas en pratique poser de problème particulier. Le décalage spatial entre le périphérique logique et l'objet édité est plus gênant car il peut induire un va-et-vient du point d'attention. Typiquement, ce décalage est nul dans le cas des interactions à manipulation directe (poignées de sélection par exemple), mais il est important dans les manipulations indirectes (boîtes de dialogue ou de propriétés par exemple). Même s'il est faible dans le cas des barres de défilement qui sont proches des documents contrôlés, il suffit à provoquer un va-et-vient du point d'attention qui distrait l'utilisateur et provoque des erreurs de manipulation. Les décalages temporels concernent les intervalles de temps entre l'action sur l'instrument et le retour d'information. Si ces décalages sont trop importants (à l'échelle de la perception humaine), ils risquent de briser la boucle perception-action et de faire perdre à l'utilisateur le lien de causalité entre ses actions et les réponses du système [21, décrit dans 10]. Les décalages temporels niveau des instruments physiques et logiques sont en gén faibles et ne posent pas de problème. Au niveau de l'o édité, le décalage temporel peut être dû à un problème performances que l'on peut résoudre en dégradant représentation de l'objet [26]. Par exemple si le déplacem d'une fenêtre avec son contenu en temps réel est trop coûte on peut afficher seulement son cadre. Le décalage tempore niveau de l'objet peut également être dû à l'action différé l'instrument logique. Ainsi, une boîte de dialogue n'agit l'objet édité que lorsqu'on la valide, créant ainsi un décal temporel maximal. Au contraire, une boîte de propriétés immédiatement sur l'objet édité et induit donc un décal temporel nul.

Le *degré d'intégration* d'un instrument est le rapport entr nombre de degrés de liberté utilisés par l'instrument logiqu ceux fournis par l'instrument physique. Par exemple une b de défilement utilise un degré de liberté sur les deux que fournit la souris tandis que le défilement direct d'un docum par l'outil "main" exploite les deux degrés de liberté d souris. Le degré d'intégration recouvre les notions de tâc intégrales et séparables [17]. Un plus grand degré d'intégra correspond à une interaction plus efficace car la bande pass en entrée est mieux exploitée.

Un exemple : Bryce 2

Parmi les logiciels commerciaux, peu s'écartent du mo décrit au début de cet article. La version 2 de Bryce (Metato est une exception remarquable que l'on peut analyser selo modèle d'interaction instrumentale. Il s'agit d'un modeleur qui permet la réalisation de scènes réalistes. C'est donc logiciel complexe, comparable aux applications évaluées haut. Pourtant, l'évaluation de Bryce contraste nettement ces applications, comme le montre la table 3.

L'interface utilise diverses palettes munies d'instruments efficaces : un grand nombre d'entre eux permettent de contr deux paramètres simultanément par le déplacement XY d souris (degré d'intégration égal à 1). Toute action a un e immédiat et incrémental, soit sur la scène principale, soit une version réduite lorsque les calculs sont trop coûteux (fa degré d'indirection). L'instrument qui permet de définir paramètres d'un matériau fait appel à 57 éléments d'interf Ils sont regroupés sur un seul écran sans que cette comple ne soit apparente. A titre de comparaison, Word utilis boîtes de dialogue pour spécifier les 84 attributs d'un style Excel utilise un enchaînement de 5 boîtes pour spécifier options d'affichage d'un graphique. Enfin, les matériaux Bryce sont des objets à part entière : ils peuvent être cr édités, copiés, collés, et sauvés dans des fichiers.

Critère	Moy	Bryce2	% de Moy
#menus	7,7	3	38,9%
#cmds	95,2	45	47,3%
#dlog	36,8	18	48,9%
#ssmenu	17,3	0	0,0%
#sscmds	67,0	0	0,0%
#ssdlog	19,7	0	0,0%
Tcmds	144,8	45	31,1%

Tdlogs	56,5	18	31,8%
Cmds/M	12,5	15,0	120,0%
Cmds/SM	3,9	0,0	0,0%
#palettes	8,3	9	108,4%
#outils	81,7	71	86,9%
#prefs	7,8	1	12,8%
#options	60,0	5	8,3%

Table 3 : évaluation de Bryce

NOUVELLES TECHNIQUES D'INTERACTION

De nombreuses techniques ont été proposées pour améliorer les interfaces graphiques. Nous en analysons un certain nombre dans cette section afin de montrer que le modèle d'interaction instrumentale dépasse largement le cadre des interfaces graphiques classiques.

Couplage action/perception

L'importance du couplage entre action et perception a été noté par de nombreux auteurs et appliqué notamment aux systèmes de visualisation d'information comme l'Information Visualizer [11] et les Dynamic Queries [1]. Dans ces systèmes, l'action continue sur des paramètres produit une réaction continue des objets affichés, facilitant l'ajustement de réglages. Cela correspond pour nos instruments au cas où le décalage temporel du retour d'information sur les objets édités est nul.

Dans Pad++ [6], une surface plane infinie peut être affichée à n'importe quelle résolution. La navigation dans cet espace utilise un instrument permettant de se déplacer en XY et de contrôler le facteur de zoom. Le fort couplage entre les actions de navigation et les informations affichées permet de se repérer aisément. Cette interface a permis l'invention d'un nouveau type d'outil que l'on peut laisser sur le plan de travail et retrouver plus tard là où on les a laissés, comme de vrais outils [7]. Selon notre modèle, ces outils sont des instruments parfaitement réifiés.

Interaction gestuelle

Dans l'interaction de tracé ("drag-and-drop", cliquer-tirer), seules les positions de départ et d'arrivée sont utilisées, sauf pour les logiciels de dessin qui permettent la saisie de tracés à main levée. Dans l'interaction de "drag-and-drop", un retour d'information est fourni au niveau de l'instrument, indiquant les zones sensibles lorsque l'on passe dessus. Mais les objets édités ne prennent pas part à cette interaction. Dans la nouvelle version du Finder du Macintosh (MacOS 8.0), il est possible de parcourir une hiérarchie de dossiers par une simple interaction de tracé : si l'on reste immobile sur l'icône d'un dossier, la fenêtre du dossier s'ouvre et permet de naviguer vers un sous-dossier. Il s'agit donc d'un instrument de navigation sophistiqué, qui interagit avec l'objet édité (le système de fichiers) tout au long de l'interaction et non pas seulement à la fin.

Dans les interfaces gestuelles à reconnaissance de marques [4], l'instrument logique est la feuille transparente superposée à la surface de travail et sur laquelle s'inscrit l'encre électronique. Cette feuille interprète le tracé et transmet la commande correspon-dante à l'objet édité. Dans les Marking Menus [19] deux instruments sont combinés : les marques commandent un menu radial qui n'apparaît qu'en cas d'hésitation de l'utilisat qui lui-même contrôle l'objet édité.

Interaction bimanuelle

Les interfaces actuelles utilisent souvent la main dominante pour la sélection de mode par l'intermédiaire touches du clavier. D'autres formes d'interaction bimanue sont obtenues en affectant un second périphérique localisation (en général une track-ball) à la main dominante.

Un bon exemple d'interaction bimanuelle est la technique ToolGlasses [8]. Une ToolGlass est une palette se transparente dont la position est contrôlée par la main dominante. L'activation d'un outil de la palette est réalisée un clic "à travers", qui spécifie simultanément l'outil utilis l'objet auquel il s'applique. Si nécessaire, une interaction tracé peut être enchaînée avec le clic "à travers". Cette fo d'interaction s'apparente à des actions familières comme le t d'un trait avec la main dominante en utilisant une règle te par la main non dominante. Elle permet des gains de te appréciables [20] dus au parallélisme possible entre l'action deux mains. Les ToolGlasses s'interprètent immédiate comme des instruments dont la caractéristique principale d'éliminer le décalage spatial entre l'instrument logique l'objet édité, ce qui les rend plus direct que les boîtes dialogue et de propriétés. De plus ces instruments complètement réifiés puisque l'utilisateur peut composer propres ToolGlasses.

Nouveaux périphériques physiques

Nous avons vu que l'affectation d'un instrument physiqu un instrument logique peut nécessiter une action explicite sélection de la part de l'utilisateur, à moins de multiplier périphériques d'entrée ou d'augmenter leurs degrés de liberté.

Plusieurs chercheurs ont introduit un dispositif permettan capturer la rotation de la souris dans le plan horizontal avec des applications immédiates aux éditeurs graphiques. Rocking Mouse [2] est une souris 2D qui mesure le tangag le roulis, soit deux nouvelles dimensions. Wacom propose tablettes qui captent la pression et l'inclinaison du stylet et peuvent distinguer plusieurs stylets, notamment un st réversible dont une extrémité sert à dessiner et l'autre à effa Les chercheurs de Toronto et du Media Lab ont commenc explorer la notion de "graspable interface" [14, 16], qui per de manipuler les objets de l'interface par action directe sur objets physiques. Tous ces travaux ont pour objet de virtualiser" les instruments de l'interaction graphique, transférant les instruments logiques vers les instrum physiques afin de mieux exploiter notre dextérité manuelle.

Réalité augmentée

Une approche symétrique consiste à prendre des instrum et des objets physiques, et à y intégrer des capacités traitement de l'information de façon à utiliser nos capac naturelles à interagir avec des objets physiques. C approche, dite de réalité augmentée [27, 5], utilise plusi techniques pour réaliser cette fusion entre monde physiqu monde informatique : intégration de capacités de calcul dans objets physiques, projection de données informatiques sur objets physiques comme le papier, etc.

Dans tous les cas, ce sont des instruments physiques qui supportent l'interaction (stylos, briques de Lego, papier, etc.). Les instruments logiques tendent à disparaître car les objets eux-mêmes ont une existence physique : l'interaction instrumentale a lieu alors principalement dans l'univers physique. La réification n'est pas absente des systèmes de réalité augmentée. Ainsi, le répondeur téléphonique de Durell Bishop distribue une bille à chaque message qu'il reçoit. La bille peut être insérée dans le répondeur pour lire le message ou dans le téléphone pour rappeler son auteur [13]. On peut aussi simplement la garder dans sa poche comme pense-bête. En réifiant le message en une bille, le système permet d'utiliser celle-ci non seulement comme objet contenant le message mais également comme instrument de communication et d'interaction sociale.

La communication instrumentale de C. Cadoz

Dans son étude du geste comme canal de communication, Claude Cadoz [9] décrit les trois fonctions du geste : la fonction epistémique (toucher pour acquérir de l'information), la fonction ergotique (transformer les objets par action physique) et la fonction sémiotique (communiquer de l'information). Puis il introduit la notion de geste instrumental qui combine ces trois fonctions : c'est un geste qui s'applique à un objet matériel, qui nécessite une interaction physique avec celui-ci, et dont les effets sur l'objet sont traduits en messages communicationnels. Cadoz classe ces gestes en gestes d'excitation qui fournissent l'énergie qui sera traduite en information, gestes de modulation continus ou discrets qui modifient les propriétés de l'instrument, et gestes de sélection qui permettent de choisir des éléments.

Bien que notre notion d'instrument soit légèrement différente de celle de Cadoz, on retrouve ces trois types d'interactions dans notre modèle : les geste de sélection choisissent les instruments logiques, les gestes d'excitation démarrent les interactions et les gestes de modulation contrôlent leur déroulement.

CONCLUSION

Nous avons montré dans cet article que la majorité des applications actuelles se conforment à un modèle qui détourne les principes de la manipulation directe. Nous avons proposé un nouveau modèle, l'interaction instrumentale, qui permet de décrire les applications existantes ainsi qu'un grand nombre de nouvelles techniques d'interaction.

Au-delà de ce pouvoir descriptif, l'objet de notre modèle est de servir de guide d'une part aux concepteurs d'interfaces, d'autre part aux développeurs d'outils de construction d'interface. Si l'on veut améliorer la qualité des applications les plus répandues en développant l'interaction instrumentale, il faut que les environnements de développement disposent de composants prêt à l'emploi permettant de la mettre en œuvre. C'est la principale direction dans laquelle nous souhaitons poursuivre ces travaux.

REMERCIEMENTS

L'auteur remercie particulièrement l'un des relecteurs anonymes dont les commentaires détaillés ont permis d'améliorer la qualité de cet article.

BIBLIOGRAPHIE

1. C. Ahlberg, C. Williamson, B. Shneiderman. Dyna Queries for Information Exploration: An Implementa and Evaluation. In *Proc. ACM Conference on Hu Factors in Computing Systems, CHI'92*, ACM P 1992, pp 619-626.
2. R. Balakrishnan, T. Baudel, G. Kurtenbach, Fitzmaurice. The Rockin' Mouse: Integral Manipulation on a Plane. In *Proc. ACM Conference Human Factors in Computing Systems, CHI'97*, A Press, 1997, pp 311-318.
3. C. Bastien, D. Scapin. *Ergonomic Criteria for Evaluation of Human-Computer Interfaces*. Rap Technique INRIA n° 156, juin 1993.
4. T. Baudel. *Aspects morphologiques de l'interac humaine-ordinateur : étude de modèles d'interac gestuels*. Thèse de doctorat, LRI, Université de Paris- décembre 1995.
5. M. Beaudouin-Lafon. Beyond the Workstat Mediaspaces and Augmented Reality. In *Proc. HCI People and Computers IX*, Cambridge University P 1994, pp 9-18.
6. B. Bederson, J. Hollan. Pad++: A Zooming Graph Interface for Exploring Alternate Interface Physics. *Proc. ACM Symposium on User Interface Software Technology, UIST'94*, ACM Press, 1994, pp 17-26.
7. B. Bederson, J. Hollan, A. Druin, J. Stewart, D. Rog D. Proft. Local Tools : an Alternative to Tool Pale In *Proc. ACM Symposium on User Interface Softw and Technology, UIST'94*, ACM Press, 1994, pp 170.
8. E. Bier, M. Stone, K. Pier, W. Buxton, T. De R Toolglass and Magic Lenses : the See-Through Interf In *Proc. ACM SIGGRAPH*, 1993, pp 73-80.
9. C. Cadoz. Le geste canal de communication hom machine - la communication "instrumentale". *Techni et Science Informatique*, 13(1), janvier 1994, pp 31-6
10. S. Card, T. Moran, A. Newell. *The Psychology Human-Computer Interaction*. Lawrence Erlb Associates, 1983.
11. S. Card, G. Robertson, J. Mackinlay. The Informa Visualizer, an Information Workspace. In *Proc. Conference on Human Factors in Computing Syste CHI'91*, ACM Press, 1991, pp 181-187.
12. S. Chatty. Extending a Graphical Toolkit for Handed Interaction. In *Proc. ACM Symposium on Interface Software and Technology, UIST'94*, A Press, 1994, pp 195-204.
13. G. Crampton Smith. The Hand That Rocks The Cra *I.D.*, mai/juin 1995, pp 60-65.
14. G. Fitzmaurice, H. Ishii, W. Buxton. Laying Foundations for Graspable User Interfaces. In *ACM Conference on Human Factors in Compu Systems, CHI'95*, ACM Press, 1995, pp 442-449.

15. J. Foley, A. van Dam. *Computer Graphics, Principles and Practice*. Addison-Wesley, 1992.

16. H. Ishii, B. Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proc. ACM Conference on Human Factors in Computing Systems, CHI'97*, ACM Press, 1997, pp 234-241.

17. R. Jacob, L. Sibert, D. McFarlane, M. Preston Mullen. Integrability and Separability of Input Devices. *ACM Transactions on Human Computer Interaction*, 1(1), mars 1994, pp 3-26.

18. P. Kabbash, W. Buxton, A. Sellen. Two-Handed Input in a Compound Task. In *Proc. ACM Conference on Human Factors in Computing Systems, CHI'94*, ACM Press, 1994, pp 417-423.

19. G. Kurtenbach, W. Buxton. The Limits of Expert Performance Using Hierarchical Marking Menus. In *Proc. ACM Conference on Human Factors in Computing Systems, InterCHI'93*, ACM Press, 1997, pp 482-487.

20. G. Kurtenbach, G. Fitzmaurice, T. Baudel, W. Buxton. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. In *Proc. ACM Conference on Human Factors in Computing Systems, CHI'97*, A Press, 1997, pp 35-42.

21. A. Michotte. *La perception de la causalité.* Publicati Universitaires de Louvain, 1946.

22. J. Nanard. *La manipulation directe en interaction hom machine*. Thèse d'état, Université de Montpellier 1990.

23. D. Norman, S. Draper (Eds). *User-Centered Sys Design: New Perspectives on Human-Comp Interaction*. Lawrence Erlbaum Associates, 1986.

24. B. Shneiderman. Direct Manipulation : a Step Bey Programming Languages. *IEEE Computer*, 16(8), 1983, pp 57-69.

25. D. Smith, C. Irby, R. Kimball, B. Verplank, Harslem. Designing the Star User Interface. *Byte*, 7 avril 1982, pp 242:282.

26. S. Tang, M. Linton. Pacers, Time Elastic Objects. *Proc. ACM Symposium on User Interface Software Technology, UIST'93*, ACM Press, 1993, pp 35-44.

27. P. Wellner, W. Mackay, R. Gold. Computer-Augmer Environments. *Special Issue of Communications of ACM*, 23(7), juillet 1993.