

CIS : Décrire les techniques d'interaction et les évaluer en contexte

Caroline Appert

LRI & INRIA Futurs
Bâtiment 490
91405, Orsay, France
appert@lri.fr

RESUME

Cet article présente le modèle Complexity of Interaction Sequence (CIS). CIS décrit les techniques d'interaction et opérationnalise des contextes d'utilisation pour prédire l'efficacité d'une technique en contexte. CIS a été validé par une expérimentation contrôlée comparant les palettes, les palettes bimanuelles et les toolglasses.

MOTS CLES : Technique d'Interaction, Séquence d'Interaction, Complexité, Contexte, Palette, Palette Bimanuelle, Toolglass, Expérimentation.

ABSTRACT

This paper introduces the (CIS) model. CIS describes interaction techniques and operationalizes contexts of use in order to predict the technique efficiency in context. CIS has been validated by a controlled experiment comparing palettes, bimanual palettes and toolglasses.

KEYWORDS : Interaction Technique, Interaction Sequence, Complexity, Context, Palette, Bimanual Palette, Toolglass, Experimentation.

INTRODUCTION

Afin de rendre les applications graphiques plus utilisables, la recherche en IHM a créé de nombreuses techniques d'interaction. Cependant, très peu de logiciels les utilise. Une des raisons est la difficulté de choisir la technique plus performante parmi un ensemble de techniques. L'évaluation d'une technique d'interaction est souvent faite en comparant sa performance à celle d'autres techniques sur une tâche choisie dans le cadre d'une expérimentation contrôlée. Cependant, les résultats sont difficilement généralisables car ils dépendent de la tâche qui ne représente pas un contexte d'utilisation réel.

Nous présentons ici un nouveau modèle, Complexity of Interaction Sequences (CIS), qui permet de prédire l'efficacité d'une technique d'interaction dans un contexte particulier. Ce modèle permet non seulement de décrire la structure d'une technique d'interaction mais aussi un contexte à travers la notion de *séquence d'interaction*. Le but de CIS n'est pas de remplacer les

autres formes d'évaluation mais d'aider à mieux comprendre l'effet du contexte sur la performance.

Après un bref aperçu des travaux antérieurs, nous présentons, tout d'abord, comment CIS décrit l'interaction, puis, comment l'outil SimCIS donne une mesure de la performance d'une technique d'interaction. Nous présentons, enfin, l'expérimentation contrôlée que nous avons réalisée pour valider CIS. Nous comparons les données expérimentales aux prédictions de CIS sur les palettes, les palettes bimanuelles et les toolglasses.

TRAVAUX ANTERIEURS

Bien que correctement réalisées, la plupart des expérimentations contrôlées ne prennent pas explicitement en compte le contexte d'utilisation. Par exemple, Kabbash et al. [4] comparent les toolglasses aux palettes d'outil sur une tâche qui impose un changement d'outil à chaque interaction. Les toolglasses [1] sont des palettes d'outils semi-transparentes et déplaçables avec la main non dominante. Pour appliquer un outil sur un objet, il suffit de cliquer sur l'objet au travers de l'outil. Avec une palette, il faut quitter l'espace de travail pour aller sélectionner l'outil puis revenir l'appliquer sur l'objet sauf si l'outil est déjà sélectionné. L'expérimentation rapporte que les toolglasses sont plus rapides que les autres palettes d'outils. Cependant, la tâche qu'ils ont utilisée semble défavorable aux palettes : le sujet n'ayant jamais à utiliser deux fois de suite le même outil, il se voit contraint à de nombreux allers retours avec les palettes.

Mackay [6] a mené une expérimentation contrôlée démontrant que les utilisateurs organisent leurs interactions selon leur contexte cognitif. Lorsque les utilisateurs sont dans un contexte "copy", c'est-à-dire qu'ils doivent reproduire un dessin, ils ont tendance à regrouper leurs utilisations d'un même outil. Lorsqu'ils sont dans un contexte "problem solving", c'est-à-dire qu'ils doivent effectuer les bonnes modifications sur un dessin, ils ont plutôt tendance à suivre leur raisonnement de travail en créant les objets incrémentalement à partir d'un focus sans minimiser les changements d'outil. Ce résultat est accompagné des préférences des

utilisateurs qui sont différentes selon le contexte : les palettes flottantes pour un contexte "copy" et les toolglass dans un contexte "problem solving".




Ces expérimentations sont coûteuses et il est impossible de tester toutes les tâches possibles. Nous avons besoin d'un modèle capable de décrire l'interaction et de prédire sa performance. De nombreux modèles existent, nous ne présentons ici que ceux qui ont un niveau d'abstraction semblable à celui de CIS.

Le modèle de Card et al. [3] décrit les périphériques d'entrées comme des traducteurs de propriétés physiques en propriétés logiques. Ce modèle permet de comparer les périphériques sur un ensemble de propriétés comme leur vitesse de pointage, leur précision, etc. Goals, Operators, Methods and Selection rules (GOMS) [4] est une famille de modèles qui décrivent l'interaction au niveau de la tâche afin de prédire le temps requis. Une tâche est décrite par un ensemble de méthodes permettant d'atteindre un but. Une méthode est une suite d'opérateurs dont le temps est prédéterminé. Les prédictions sont donc obtenues en sommant les temps des opérateurs nécessaires pour mener à bien une tâche.

Alors que les concepteurs d'interface sont confrontés au problème de choisir et combiner des techniques d'interaction, les modèles existants décrivent l'interaction soit au niveau du périphérique d'entrée soit au niveau de la tâche. CIS propose d'aborder l'interaction au niveau de la technique.

DECRIRE L'INTERACTION AVEC CIS

CIS décrit une interface comme un ensemble d'objets graphiques que l'on peut créer, modifier et/ou effacer. L'état de l'interface est l'ensemble courant d'objets et la valeur de leurs attributs. L'espace des interactions est l'ensemble des manipulations (création, modification ou effacement) disponibles à l'utilisateur dans un état donné.

Une technique d'interaction est un ensemble d'étapes organisées sous la forme d'un graphe d'interaction décrivant comment activer une manipulation. Il y a deux types d'étape : les étapes d'acquisition qui permettent d'identifier un sous-ensemble de l'espace des interactions (par exemple, déplacer le curseur sur un outil) et les étapes de validation qui valident ces réductions (par exemple, cliquer sur l'outil pour le sélectionner). Les figures 1 et 2 correspondent aux graphes d'interaction de la palette et de la toolglass contenant chacune trois outils pour créer des triangles, des rectangles et des ellipses :   . Par exemple, le chemin le plus à gauche dans la figure 1 commence par réduire l'espace des interactions à $\{(Create_triangle, p) \mid p \in position\}$ en déplaçant le curseur sur l'outil pour créer des triangles (acquisition/arc) et en cliquant

(validation/nœud). Enfin, déplacer le curseur vers la position pos puis en cliquant, l'espace des interactions est alors réduit à l'unique élément ($Create_triangle, pos$), ce qui permet d'exécuter cette manipulation.

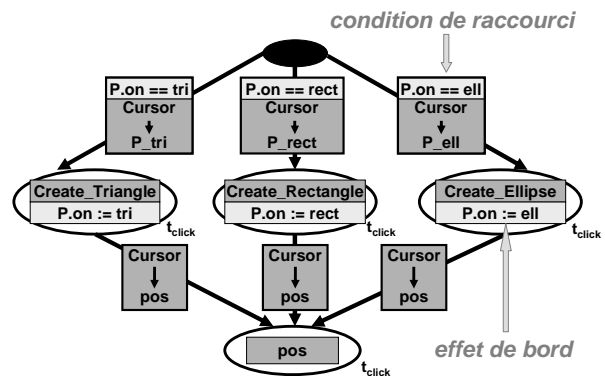


Figure 1: Graphe d'interaction de la palette d'outils.

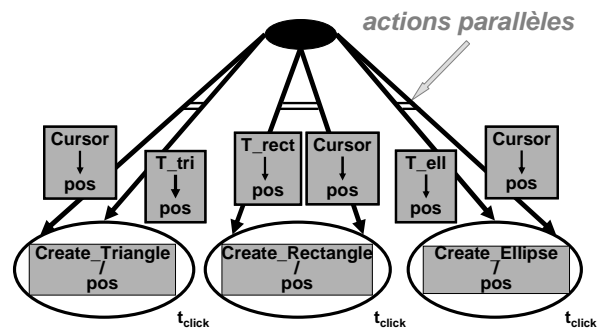


Figure 2: Graphe d'interaction de la toolglass.

CIS permet de décrire certaines propriétés des techniques d'interaction comme la *persistance* ou le *parallélisme*. Les nœuds peuvent avoir des effets de bord sur l'état de l'interface. Par exemple, la sélection d'un outil dans une palette est persistante : l'outil est actif pour les prochaines interactions. La condition de raccourci indique justement que si l'outil est actif, l'acquisition et la validation qui suivent peuvent être ignorées. Certaines étapes peuvent être *parallèles* : par exemple, le curseur et la toolglass peuvent être déplacés soit simultanément soit séquentiellement.

Les résultats rapportés par Mackay [6] mettent en avant la différence d'organisation du travail des utilisateurs et donc la différence de performance d'une technique d'interaction selon le contexte. Dans un contexte "copy", les utilisateurs créent ensemble les formes de même type alors que, dans un contexte "problem solving", ils travaillent incrémentalement autour d'un focus sans minimiser les changements d'outil. CIS opérationnalise la notion de contexte par la notion de *séquence d'interaction*. Par exemple, pour atteindre l'état S à partir de l'état S_{init} de la figure 3, un contexte "problem solving" est décrit par la séquence Seq_1 tandis qu'un contexte "copy" est décrit par la séquence Seq_2 .

Seq ₁	Seq ₂	
(Ellipse, (300, 100))	(Ellipse, (300, 100))	
(Triangle, (400, 100))	(Ellipse, (700, 100))	
(Rectangle, (500, 100))	(Ellipse, (300, 500))	
(Triangle, (300, 400))	(Rectangle, (500, 100))	
(Rectangle, (300, 300))	(Rectangle, (500, 300))	
(Ellipse, (700, 100))	(Rectangle, (300, 300))	
(Triangle, (600, 100))	(Triangle, (400, 100))	
(Triangle, (600, 200))	(Triangle, (600, 100))	
(Rectangle, (500, 300))	(Triangle, (600, 200))	
(Ellipse, (300, 500))	(Triangle, (400, 400))	
(Triangle, (400, 400))	(Triangle, (300, 400))	
(Triangle, (300, 200))	(Triangle, (300, 200))	

Figure 3: Deux séquences pour passer de S_{init} à S .

PREDIRE LA PERFORMANCE AVEC CIS

Nous avons développé l'outil SimCIS afin de prédire la performance d'une technique d'interaction pour une séquence d'interaction donnée. SimCIS prend en entrée l'état initial de l'interface, la séquence d'interaction et le graphe d'interaction de la technique pour fournir en sortie des mesures de performance : la complexité en nombre d'actions et en temps de la technique. Les complexités correspondent au nombre d'étapes (acquisition et validation) et au temps requis pour mener à bien la séquence. Pour produire ces résultats, SimCIS construit, dans le graphe d'interaction, le chemin représentant la séquence et évalue le temps de ce chemin en utilisant la loi de Hick pour les branchements, la loi de Fitts pour les arcs et les temps fixes pour les nœuds étiquetés. SimCIS fournit également une représentation graphique des actions correspondantes pour aider visuellement à comprendre les différences de résultats.

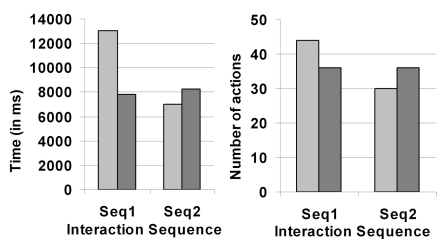


Figure 4: Comparaison des complexités [Palette - Toolglass].

CIS confirme les préférences des utilisateurs : la toolglass est plus efficace pour la séquence Seq1 ("problem solving") alors que la palette est plus efficace pour Seq2 ("copy"). Les représentations graphiques fournies par CIS montrent que la différence de complexité entre la palette et la toolglass dans un contexte "problem solving" est due aux nombreux aller-retour de l'espace de travail à la palette causés par les changements d'outil.

VALIDER CIS

Afin d'évaluer CIS, nous avons réalisé une expérimentation contrôlée pour comparer les trois techniques suivantes : Palettes (P), Palettes Bimanuelles (PB) et Toollasses (T). La Palette Bimanuelle est une technique implémentée dans le projet CPN2000 [2] : pour appliquer un outil sur un objet, l'utilisateur sélectionne l'outil à l'aide d'un curseur relié à la main

non dominante et l'objet à l'aide d'un deuxième curseur relié à la main dominante. Comme la palette, la sélection de l'outil est persistante et, comme la Toolglass, les déplacements des curseurs sont parallèles.

Cette expérience avait trois buts :

- Comparer les prédictions aux données empiriques.
- Evaluer la palette bimanuelle jamais testée jusqu'ici.
- Mieux comprendre le lien entre les propriétés d'une technique et son efficacité.

Douze adultes ont volontairement accepté non seulement de participer à l'expérimentation mais aussi de répondre à un questionnaire sur leurs préférences subjectives. Chaque sujet avait pour consigne d'effacer le plus rapidement possible un ensemble de formes dessinées à l'écran soit dans un ordre imposé par une ligne noire reliant les formes entre elles soit dans l'ordre de son choix. Chaque technique contient 3 outils correspondant aux trois types de forme (rectangle, triangle et ellipse) : pour effacer une forme, l'utilisateur doit appliquer le bon outil sur cette forme. La figure 5 montre les différents arrangements que nous avons utilisés afin d'opérationnaliser différents contextes d'utilisation.

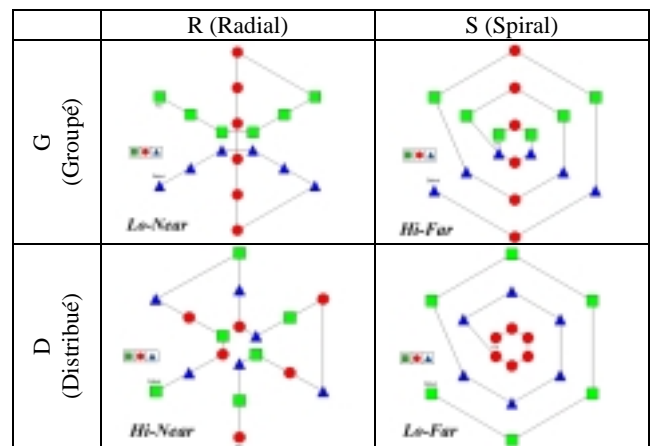


Figure 5: Les 4 types de séquence imposés (Longueur = 18).

Les conditions (3*2*2*3 inter-sujet) étaient :

- La technique : P, PB ou T
- Le groupement : G ou D
- L'ordre : R, S ou L
- La longueur : 6 ou 18 (la figure 4 ne montrent que les conditions où longueur = 18, c.a.d. 18 formes à effacer).

Pour analyser les résultats, nous avons utilisé deux types de regroupement sur la figure 5. D'une part, nous avons confronté les deux colonnes distinguant les contextes où les formes sont proches et ceux où elles sont éloignées (Near vs Far). D'autre part, nous avons confronté les diagonales distinguant les contextes demandant peu et beaucoup de changements d'outil (Lo vs Hi).

Les prédictions de CIS sur cette expérimentation sont les suivantes :

(P₁) P et PB sont sensibles au nombre de changements d'outil à cause de la propriété de persistance.

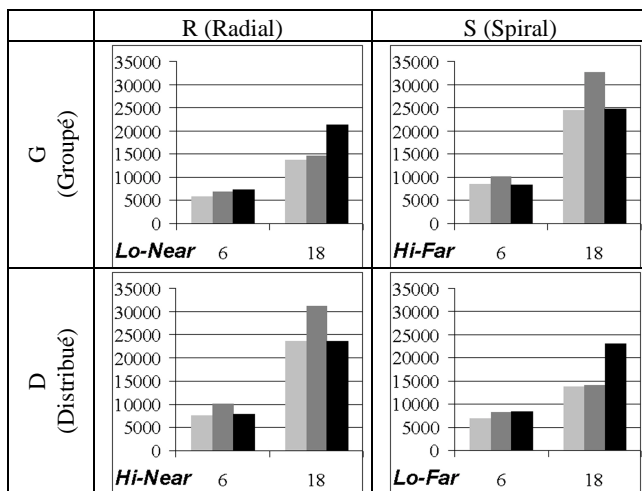
(P₂) PB est plus ou aussi efficace que les deux autres techniques car elle combine parallélisme et persistance.

(P₃) Les techniques sont sensibles aux distances entre objets à cause de l'effet de la loi de Fitts.

(P₄) Plus la séquence est longue, plus la différence est importante entre le pire et le meilleur cas car l'optimisation est fonction de l'anticipation.

La condition dans laquelle les utilisateurs sont libres de l'ordre dans lequel effacer les formes (L) et les préférences subjectives collectées lors du sondage vont nous permettre de vérifier l'hypothèse :

(H_{users}) Les utilisateurs sont capables d'optimiser l'utilisation d'une technique et de l'adapter au contexte.



La figure 5 présente les données empiriques collectées :

- valident P₁ : Les palettes (P et PB) sont plus rapides dans les conditions Lo que dans les conditions Hi.
- valident P₂ : Le temps minimal pour un contexte donné est toujours celui de la palette bimanuelle.
- ne valident pas P₃ : Les histogrammes des contextes Near et Far sont très similaires, la différence n'est pas significative. CIS surestime certainement l'effet de la loi de Fitts.
- valident P₄ : Les différences entre le temps minimal et le temps maximal sont plus de trois fois supérieures lorsque longueur = 18 que lorsque longueur = 6.

Les temps réalisés sur les essais dans lesquels l'ordre n'est pas imposé sont très proches du meilleur temps

réalisé sur les essais dans lesquels l'ordre est imposé. De plus, les utilisateurs préfèrent PB pour effacer un ensemble de formes dans l'ordre de leur choix. S'ils doivent choisir entre P et T, ils préfèrent P lorsque les formes sont groupées et T lorsqu'elles sont distribuées. Ces résultats nous permettent de valider H_{users}.

CONCLUSION ET PERSPECTIVES

Cet article a présenté le modèle CIS pour décrire les techniques d'interaction et les évaluer en contexte. Nous l'avons utilisé pour montrer le lien entre contexte et efficacité. CIS est un outil pour aider à la conception d'évaluation de techniques d'interaction.

Le travail à venir peut prendre plusieurs directions. Premièrement, le modèle permet déjà la description de plusieurs techniques au sein d'une même interface mais il n'a pas été testé dans ces conditions. Deuxièmement, nous avons vu la sensibilité du modèle à l'effet de la loi de Fitts, le modèle doit donc être raffiné pour fournir de meilleures prédictions. Enfin, il faudrait automatiser la recherche de la pire et de la meilleure séquence pour atteindre un but. Ce dernier point doit aborder l'explosion combinatoire des séquences possibles.

BIBLIOGRAPHIE

1. Bier, E.A., Stone, M.C., Pier, K. & Buxton, W. Toolglass and Magic Lenses: the See-Through Interface. *Proc. ACM Siggraph*. (1993). ACM Press, pp. 73-80.
2. Beaudouin-Lafon, M. & Lassen, H.M. (2000). CPN2000: A Post-WIMP Graphical Application.. *Proc. ACM Symposium on User Interface Software and Technology (UIST'00)*. p. 181-190
3. Card, S.K., Robertson, G. & Mackinlay, J. A. (1991). Morphological Analysis of the Design Space of Input Devices. *Proc. ACM Transactions on Information Systems*, 9(2), p. 99-122.
4. John, B. E. & Kieras, D. E. (1996). The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *Proc. ACM Transactions on Computer-Human Interaction*. 3(4), p. 320-351.
5. Kabbash, P., Buxton, B. & Sellen, A. (1994). Two-handed Input in a Compound task. *Proc. ACM Human Factors in Computing Systems (CHI'94)*, pp. 417-423.
6. Mackay, W.E. (2002). Which Interaction Technique Works When? Floating Palettes, Marking Menus and Toolglasses support different task strategies. *Proc. ACM Conference on Advanced Visual Interfaces (AVI'02)*. pp. 203-208.