



CIS : Evaluer les Techniques d'Interaction en Contexte

Caroline Appert

► **To cite this version:**

Caroline Appert. CIS : Evaluer les Techniques d'Interaction en Contexte. Annexes des Actes des 15èmes journées francophones sur l'Interaction Homme Machine, 2004, Caen, France. inria-00539088

HAL Id: inria-00539088

<https://hal.inria.fr/inria-00539088>

Submitted on 24 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CIS : Evaluer les Techniques d'Interaction en Contexte

Caroline Appert
LRI & INRIA Futurs
Université Paris Sud
Orsay, France
appert@lri.fr

INTRODUCTION

La recherche en IHM a introduit de nombreuses techniques d'interaction afin d'améliorer l'utilisabilité des interfaces. Cependant, les produits commerciaux actuels n'en exploitent que très peu. Ce constat est probablement issu de la difficulté d'évaluer l'efficacité d'une technique avant de l'introduire dans une interface réelle. Très souvent, les techniques d'interaction sont évaluées par des expérimentations contrôlées sur une tâche ne reflétant pas une réelle utilisation en contexte, les résultats sont alors difficilement généralisables.

Par exemple, suite à une expérimentation contrôlée sur la tâche « connect the dots », Kabbash & Buxton [3] ont montré un gain d'efficacité de 40% des toolglasses [2] par rapport aux palettes d'outils. Les résultats rapportés par Mackay [4] sont plus nuancés : les utilisateurs préfèrent et sont plus efficaces avec les palettes dans un contexte où le nombre de changements d'outil est minimisé alors que leur choix s'oriente vers la toolglass dans un contexte qui maximise le nombre de changements d'outil. « connect the dots » est une tâche qui maximise les changements d'outil, les résultats rapportés par Kabbash & Buxton ne semblent donc valables que dans un cas particulier.

Cet article donne un aperçu et un exemple d'utilisation du modèle CIS. CIS permet de décrire une technique d'interaction, de l'analyser grâce à un ensemble de critères et de mesurer son efficacité par rapport à une séquence d'interaction représentant le contexte d'utilisation. CIS est un complément pour l'évaluation de techniques en aidant les chercheurs à comprendre l'effet du contexte sur l'efficacité d'une technique.

CIS : UN MODELE DESCRIPTIF

CIS décrit une interface comme un ensemble d'objets que l'utilisateur peut manipuler. Certains objets sont des *objets de travail*, par exemple les formes dessinées, tandis que d'autres sont des *objets outil*, par exemple un item de menu. L'état de l'interface est défini par cet ensemble d'objets et la valeur de leurs attributs.

L'ensemble des manipulations, appelé *espace d'interaction*, est décrit par un ensemble de n-uplets de trois catégories :

- Création : $(t: T, v_1: \text{type}(att_1), \dots, v_n: \text{type}(att_n))$
 T est l'ensemble des types d'objet. Ce n-uplet crée un objet de type t avec les attributs v_1, \dots, v_n ;
- Modification : $(o: O, att, v: \text{type}(att))$

O est l'ensemble des objets de travail. Ce n-uplet affecte la valeur v à l'attribut att de l'objet o ;

- Destruction : $(o: O, \text{détruire})$.

Par exemple, considérons une interface très simple qui permet de dessiner des rectangles, ellipses et triangles de taille prédéfinie avec une palette fixe d'outils (PF) et une toolglass (TG), chacune contenant trois outils de création.

Les trois types d'objets de travail sont :

- Rectangle, Ellipse, Triangle: $\{p: \text{position}, \text{fond: couleur}, \text{contour: couleur}\}$

Les cinq types d'objets outil sont :

- PF: $\{\text{actif: \{tri, rect, ell\}}, p_{\text{tri}}: \text{position}, p_{\text{rect}}: \text{position}, p_{\text{ell}}: \text{position}, t: \text{taille}\}$
- TG_rect, TG_ell, TG_tri: $\{p: \text{position}, t: \text{taille}\}$
- Curseur: $\{p: \text{position}\}$

L'espace d'interaction est l'ensemble de n-uplets de création défini par le type : $(t: \{\text{Rectangle, Ellipse, Triangle}\}, p: \text{position})$

Une *étape d'interaction* est une séquence d'actions qui réduit progressivement l'ensemble des n-uplets possibles à un unique n-uplet. Il y a deux types d'action :

- Une *sélection* identifie un sous-ensemble de l'ensemble actuel de n-uplets. Elle est issue d'un mouvement d'objet, typiquement le curseur, vers un outil ou un objet.
- Une *validation*, typiquement un clic, confirme une sélection et réduit donc l'ensemble de n-uplets.

Une *technique d'interaction* est un ensemble d'étapes d'interaction similaires. Deux sélections sont similaires si elles mettent en jeu des objets de même type. Deux validations sont similaires si elles confirment des n-uplets de même type.

Sur l'exemple précédent, une technique d'interaction pour créer un rectangle en (10, 10) avec la palette est:

S_1 : déplacer le curseur vers p_{rect}

Sélection de $S_1 = \{(\text{Rectangle}, p: \text{position})\}$.

V_1 : clic gauche pour valider S_1

S_2 : déplacer le curseur vers (10, 10)

Sélection de $S_2 = \{(\text{Rectangle}, (10, 10))\}$.

V_2 : clic gauche pour valider S_2

CIS décrit une technique par un graphe orienté appelé le *graphe d'interaction* (figure 1). La racine est étiquetée par le nom de la technique. Un arc est une sélection et est étiqueté par l'objet à déplacer et la cible de cet objet. Un noeud (sauf la racine) est une validation, il est étiqueté par les éléments du n-uplet qu'elle instancie et la durée de l'action physique. Une validation peut changer l'état de l'interface en plus de l'instanciation partielle du n-uplet. Par exemple, la sélection d'un outil dans une palette « active » cet outil pour les futures actions. Ces changements sont

modélisés par des effets de bord du noeud. Les arcs peuvent avoir des pré-conditions. Lorsqu'une pré-condition est vérifiée, la sélection et la validation associée sont ignorées. Par exemple, lorsqu'un outil est sélectionné dans une palette, il peut être réutilisé sans le sélectionner de nouveau. Le *graphe de séquence* décrit toute l'interface. Il est construit en fusionnant toutes les racines des différents graphes et en ajoutant, pour chaque feuille, un arc vers la racine. Une *séquence d'interaction* est une séquence de n-uplets instanciés qui peut être produite par un chemin débutant et finissant à la racine du graphe de séquence.

Les graphes d'interaction peuvent décrire un grand nombre de techniques d'interaction et permettent de visualiser des propriétés utiles pour des comparaisons qualitatives comme, par exemple :

- *Ordre et Parallélisme* : Une technique d'interaction impose une organisation séquentielle et/ou parallèle de ses actions constituantes visible sur la forme du graphe d'interaction. Par exemple, la toolglass est fortement parallèle tandis qu'une palette est fortement séquentielle.
- *Persistance* : Les pré-conditions et les effets de bord permettent de mettre en avant les avantages/inconvénients issus de certaines applications successives de techniques. Par exemple, contrairement à la toolglass, la palette n'impose qu'une sélection d'outil pour créer deux rectangles à la suite.

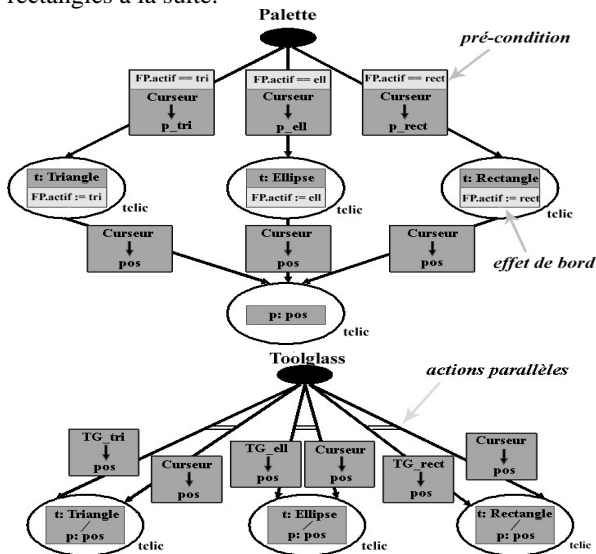


Figure 1: graphes d'interaction de la Palette et de la Toolglass
CIS : UN MODELE PREDICTIF

Afin de mesurer l'efficacité d'une technique d'interaction, nous avons introduit la *complexité*, inspirés par la complexité utilisée pour évaluer les algorithmes. Un *problème* à résoudre est un état à atteindre par une séquence d'interaction. La *taille du problème* est la longueur de la séquence. Les *actions* sont les actions de sélection et de validation qui permettent d'instancier les n-uplets de la séquence, c'est-à-dire de résoudre le problème. Il y a deux types de complexité : le nombre d'actions et le temps d'exécution de ces actions.

L'application SimCIS prend en entrée l'état initial de l'interface, le *graphe d'interaction* d'une technique et la *séquence d'interaction*. Elle construit le graphe de séquence, recherche le chemin *P* qui instancie la séquence *Seq* et évalue la complexité. La complexité en nombre d'actions est égale au nombre de nœuds et arcs de *P*. La complexité en temps est la somme des temps pris par chaque arc et nœud de *P*. Pour un arc, c'est la somme du temps pris pour choisir cet arc, évalué par la loi de Hick, et du temps pris par le déplacement d'objet, évalué par la loi de Fitts. SimCIS génère également un diagramme, le Object Movement Diagram (OMD), qui illustre les mouvements d'objets nécessaires pour la séquence. Les OMD expliquent les différences de complexité : *S₁* qui maximise les changements d'outil nécessite beaucoup plus d'aller retour vers la palette que *S₂* qui les minimise (figure 2).

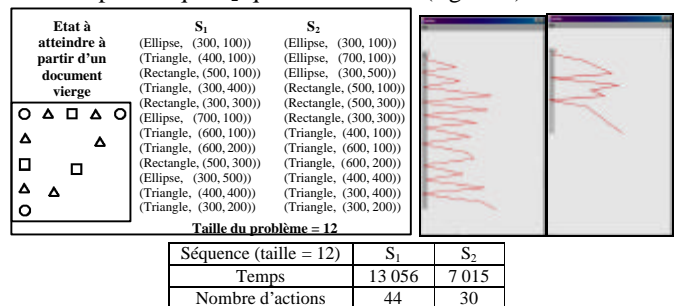


Figure 2 : OMD et complexité de la palette pour *S₁* & *S₂*

RESULTATS ET CONCLUSION

Les prédictions de CIS sur l'efficacité des toolglasses et des palettes sont présentées sur la figure 3. Elles confirment les résultats empiriques rapportés par Mackay [4].

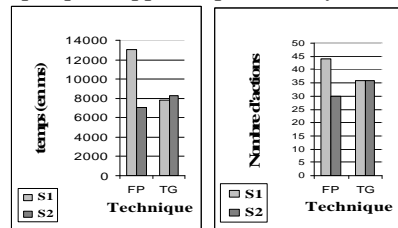


Figure 3 : Complexité des palettes et toolglasses sur *S₁* & *S₂*

Cet article a donné un aperçu du modèle CIS qui permet de décrire les techniques d'interaction et de prédire leur efficacité en contexte et les résultats obtenus sur un exemple simple (cf. [1] pour une description plus détaillée). L'efficacité d'une technique dépendant de son contexte d'utilisation, CIS permet aux chercheurs d'appréhender l'efficacité des techniques dans différents contextes afin de fournir des résultats généralisables.

REFERENCES

- [1] Appert, C., Beaudouin-Lafon, M. & Mackay, W. (2003). Context Matters: Evaluating Interaction Techniques with the CIS Model, Rapport de Recherche N°1372, Octobre 2003, LRI, Université Paris-Sud, France
- [2] Bier, E.A., Stone, M.C., Pier, K. & Buxton, W. (1993). Toolglass and Magic Lenses: the See-Through Interface. *Proc. Siggraph*, p. 73-80
- [3] Kabbash, P., Buxton, B. & Sellen, A. (1994). Two-handed Input in a Compound task. *Proc. ACM Human Factors in Computing Systems (CHI'94)*, p. 417-423
- [4] Mackay, W.E. (2002). Which Interaction Technique Works When? Floating Palettes, Marking Menus and Toolglasses support different task strategies. *Proc. ACM Working Conference on Advanced Visual Interfaces (AVI'02)*, p. 203-208