

---

# Les Interacteurs Hiérarchisés<sup>1</sup>

## Une architecture orientée tâches pour la conception des dialogues.

**Patrick Girard, Guy Pierra, Laurent Guittet**

*LISI, ENSMA, Site du Futuroscope - B.P. 109 - 86960 FUTUROSCOPE Cedex  
Tél. (33) 49-49-80-63 - E-mail {girard,pierra,guittet}@ensma.univ-poitiers.fr*

---

*RÉSUMÉ. La conception d'interfaces homme-machine doit satisfaire deux types d'exigences: ergonomiques (définies par la psychologie cognitive) et architecturales (définies par le génie logiciel). On propose dans cet article un modèle d'architecture, les Interacteurs Hiérarchisés, et l'on montre qu'il répond à ces deux exigences ; il permet, en particulier, de supporter la hiérarchie buts/sous-butts propre à une activité de conception.*

*ABSTRACT. Designing man-machine interfaces shall address two kinds of requirements: the requirements from cognitive psychology, and the software engineering principles. In this paper, we propose an architecture model called Hierarchical Interactors. This model fulfills both kinds of requirements, and supports the goal/sub-goal hierarchy that corresponds to engineering design activities.*

*MOTS-CLÉS : CFAO, IHM, Modèle d'architecture, Analyse de tâches, Génie Logiciel*

*KEYWORDS: CAD/CAM, CHI, Architecture Model, Task Analysis, Software Engineering*

---

---

<sup>1</sup> *paru dans Revue d'Automatique et de Productique Appliquée (RAPA), 1995, n° 8, vol. 2-3, pp. 235-240.*

---

# Les Interacteurs Hiérarchisés

## Une architecture orientée tâches pour la conception des dialogues.

**Patrick Girard, Guy Pierra, Laurent Guittet**

*LISI, ENSMA, Site du Futuroscope - B.P. 109 - 86960 FUTUROSCOPE Cedex  
Tél. (33) 49-49-80-63 - E-mail {girard,pierra,guittet}@ensma.univ-poitiers.fr*

---

*RÉSUMÉ. La conception d'interfaces homme-machine doit satisfaire deux types d'exigences: ergonomiques (définies par la psychologie cognitive) et architecturales (définies par le génie logiciel). On propose dans cet article un modèle d'architecture, les Interacteurs Hiérarchisés, et l'on montre qu'il répond à ces deux exigences ; il permet, en particulier, de supporter la hiérarchie buts/sous-but propre à une activité de conception.*

*ABSTRACT. Designing man-machine interfaces shall address two kinds of requirements: the requirements from cognitive psychology, and the software engineering principles. In this paper, we propose an architecture model called Hierarchical Interactors. This model fulfills both kinds of requirements, and supports the goal/sub-goal hierarchy that corresponds to engineering design activities.*

*MOTS-CLÉS : CFAO, IHM, Modèle d'architecture, Analyse de tâches, Génie Logiciel*

*KEYWORDS: CAD/CAM, CHI, Architecture Model, Task Analysis, Software Engineering*

---

### 1.1. Introduction

Les systèmes de Conception et Fabrication Assistée par Ordinateur (CFAO) sont des systèmes interactifs organisés autour d'un modèle qui représente un objet réel en cours d'analyse. À la différence des données manipulées par des outils tels MacDraw ou PowerPoint, les éléments constituant ce modèle sont fortement dépendants les uns des autres. Ainsi, afin d'éviter les approximations inacceptables pour un passage en fabrication sur les outils à commande numérique, des relations entre entités du modèle (tangence, contiguïté) doivent-elles être établies au cours de la construction, et maintenues par la suite. Ces besoins imposent des contraintes fortes sur le dialogue homme-machine de ces systèmes. Après avoir recensé, dans la section 2, quelques problèmes présentés

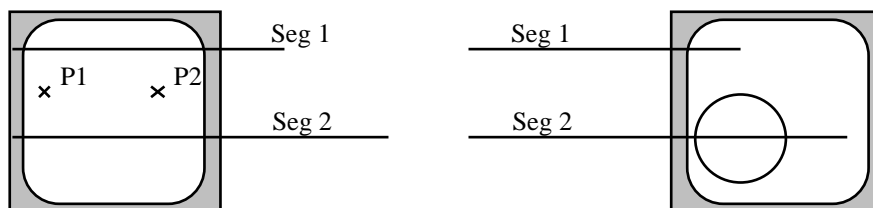
par les interfaces homme-machine des systèmes de CFAO, nous présenterons dans la section 3, la solution des Interacteurs Hiérarchisés.

## 2. Spécificités des systèmes de CFAO

Les techniques de manipulation directe [SHN 82], popularisées par les systèmes WIMP (Windows, Icons, Mouse and Pointers) sont centrées sur la manipulation individuelle des objets graphiques. Ainsi, les redimensionnements d'objets, ou encore leurs déplacements, peuvent-ils être facilement exécutés. Cependant, les opérations précises se limitent généralement aux alignements d'objets. La prise en compte de besoins du genre *"Je veux faire un trou dont le diamètre est égal au diamètre de ce cercle + 0,2, et situé à mi-distance des bords"* se révèle très difficile. Dans le meilleur des cas, elle passera par une reformulation complète du besoin, ou par le calcul manuel des valeurs numériques résultant de ces contraintes.

Les travaux de Norman [NOR 86] sur la théorie de l'activité ont permis de modéliser le comportement de l'utilisateur d'un système interactif. À partir d'un objectif (*"je veux construire une voiture"*), décomposé en sous-objectifs (*"la carrosserie, puis ..."*), l'utilisateur formule les buts à atteindre, puis l'arbre des sous-buts. On rentre alors dans le domaine de la tâche, durant laquelle l'exécution d'action et l'évaluation des résultats s'enchaînent. Afin de minimiser la distance cognitive entre le modèle mental des buts et l'activation des tâches, le système doit permettre d'exprimer en pré-ordre l'arbre des buts [COU 90][MEI 91] et disposer de différents niveaux de tâches correspondant aux différents niveaux de buts.

Illustrons cette démarche par un exemple : au cours d'une construction, le dessinateur a réalisé deux segments parallèles, de longueurs inégales, représentés sur la figure 1 (partie gauche). La fenêtre graphique ne permet pas de visualiser totalement le dessin en cours. Le but de l'utilisateur est de *"construire un cercle de rayon égal à la moitié de la distance entre les deux segments, et dont le centre est la projection de l'extrémité du petit segment sur le grand"* (partie droite de la figure 1).



**Figure 1.** Les situations initiale (à gauche) et finale (à droite)

En termes d'analyse de tâches, l'ordre d'expression des buts est le suivant : on veut créer un cercle de rayon connu et de centre déterminé ; le rayon est

déterminé par une formule grapho-numérique ; quand au centre, il résulte de l'application de l'expression purement graphique faisant intervenir projection et extrémité de segment. Cependant, l'inaccessibilité visuelle de l'extrémité du segment court suppose une translation de la fenêtre de visualisation en cours de construction.

Les interactions permettant à l'utilisateur d'exprimer son cheminement mental apparaîtraient dans l'ordre suivant :

**Créer\_Cercle ; Distance ; Seg 1 ; Seg 2 ; / ; 2 ;**

**Translation\_Fenêtre ; P 1 ; P 2 ;**

**Projection ; Extrémité ; Seg 1 ; Seg 2 ;**

Cet exemple illustre parfaitement les besoins des systèmes de CFAO :

(1) Le premier concerne la prise en compte des expressions, qu'elles soient numériques (Longueur = 3 x Hauteur), graphiques (Projection d'un point sur un segment), ou encore grapho-numériques (Longueur = Distance (Segment 1, Segment 2) / 2). Dans la majorité des systèmes actuels, la prise en compte de ces expressions est partielle, limitée à certaines constructions, ou encore donne lieu à l'établissement de sous-dialogues particuliers.

(2) Le second problème est connu dans le domaine des interfaces homme-machine sous le nom de dialogue multi-fils. Il consiste à permettre à l'utilisateur d'effectuer une sous-tâche dont la nécessité apparaît lors de la réalisation d'une tâche de plus haut niveau, et qui s'intercale au milieu de cette dernière. Dans notre exemple, il s'agit de l'action de translation de la fenêtre de visualisation. Un autre exemple plus complexe peut être envisagé : au cours du positionnement de différents objets dans un assemblage, l'utilisateur éprouve le besoin de consulter la documentation qui spécifie la procédure à suivre. Un dialogue multi-fils lui permet d'accéder à cette documentation sans quitter le contexte en cours de l'application d'assemblage, voire d'aller chercher dans cette documentation certaines valeurs qu'il souhaite introduire dans l'application.

### **3. Une architecture modulaire adaptée à l'arbre des buts**

Le modèle des Interacteurs Hiérarchisés que nous proposons s'appuie sur les principes de l'analyse modulaire [PAR 72] [WIL 93] pour isoler les éléments constituant l'interface utilisateur de l'application proprement dite. Nous présentons dans un premier temps la notion d'interacteur élémentaire, puis nous identifions les principaux Interacteurs nécessaires pour une application de CFAO. Enfin, nous décrivons l'architecture générale des Interacteurs Hiérarchisés, dont nous montrons l'adéquation avec les exigences ergonomiques.

### 3.1. Interacteurs élémentaires

Les Interacteurs élémentaires sont des *unités modulaires* dont le but principal est de *transformer* des éléments *d'entrées*, valeurs, objets ou commandes (notion de consommation) en éléments de *sortie*, valeurs ou objets (production). Ainsi, un interacteur élémentaire de position graphique transforme-t-il un *pointé* dans une fenêtre (couple x,y en coordonnées écran) en une *position graphique* dans l'espace utilisateur. Le comportement d'un interacteur peut être autonome, autorisant par exemple les modifications ou éditions. C'est le cas pour un Interacteur de saisie de chaîne de caractère, pour lequel l'effacement ou le retour arrière devront être autorisés. L'existence de cette autonomie de comportement nécessite, de la part de l'Interacteur, la possibilité de produire son propre écho, représentant son état.

Pour s'acquitter de sa tâche de transformation, l'Interacteur doit, dans certains cas, accéder à des informations externes, comme celles constituant le modèle du système de CFAO. C'est le cas par exemple de l'Interacteur de désignation, qui permet de sélectionner graphiquement (à l'aide d'un *pointé graphique*) une entité visible à l'écran (droite, cercle, ...), et d'en produire l'*identification*.

Enfin, l'interface devant supporter la totalité du dialogue d'une application, certains Interacteurs doivent permettre d'accéder aux actions constructives du système de CFAO. Un interacteur peut ainsi être représenté par la figure 2.

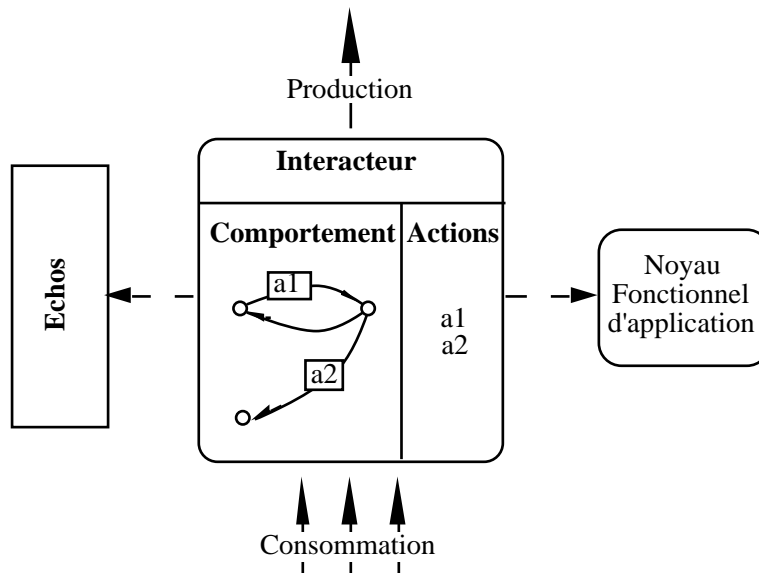


Figure 2. Interacteur élémentaire

### **3.2. Les Interacteurs dans un système de CFAO**

Dans cette section, nous présentons les différents types d'Interacteurs par *niveau d'abstraction* croissant. Il est ainsi relativement aisé d'isoler, au sein du dialogue d'un système de CFAO, les Interacteurs de bas-niveau, tels ceux de *position graphique*, de saisie de *chaîne de caractères*, et de saisie de *nombres*. Ces Interacteurs ne nécessitent pas d'accès à l'application.

L'Interacteur de *désignation* permet d'effectuer la relation avec les entités géométriques du modèle, en transformant des *pointés graphiques* en désignations d'entités du modèle. Un accès en lecture à ce dernier est donc nécessaire.

Les Interacteurs *d'expressions* (purement géométriques ou grapho-numériques) permettent de transformer ces entrées de bas-niveau. Ce sont eux qui supportent les tâches correspondant à certains des sous-buts identifiés précédemment.

Au plus haut niveau, les Interacteurs de *gestion du modèle* (création, destruction, structuration) se servent des résultats produits par les autres interacteurs. A la différence des Interacteurs précédents, ces Interacteurs ont également besoin d'un accès en *écriture* sur le modèle. Ce sont eux qui permettent à l'utilisateur d'exprimer ses buts de haut niveau. De plus, ils ne produisent pas d'éléments de sortie.

Enfin, dans cette progression logique, s'intercalent l'interacteur de *présentation*, qui permet d'effectuer la mise en page de l'image, sans affecter le déroulement des autres tâches, et l'interacteur *d'interrogation* qui permet à l'utilisateur de visualiser, ou de faire intervenir dans le flot du dialogue, des informations disponibles dans les bases de données intégrées dans l'environnement de conception.

### **3.3. Les Interacteurs Hiérarchisés**

La présentation que nous venons de faire des divers Interacteurs d'un système de CFAO définit la méthode à retenir pour les organiser. Une organisation hiérarchique, dans laquelle les Interacteurs produisent pour les niveaux d'abstraction supérieurs, et consomment la production des niveaux inférieurs, permet de répondre aux besoins de l'analyse en buts/sous-buts. La séquence d'interactions présentée à la section 2 est acceptée par cette architecture. Chaque entrée utilisateur y est traitée par l'interacteur correspondant à son niveau d'abstraction.

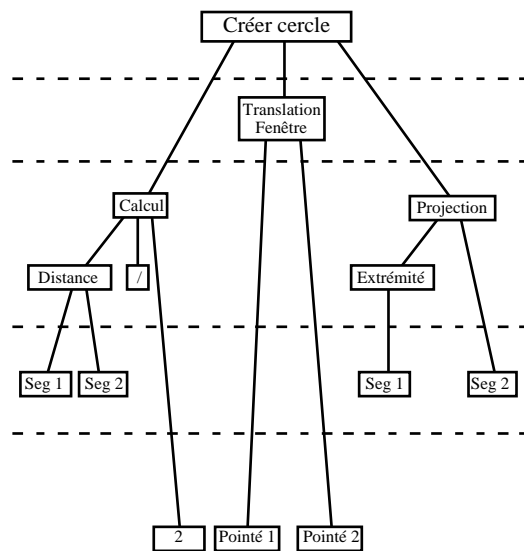
Le besoin de dialogues multi-fils, exprimé dans notre exemple par l'utilisation de la translation de la visualisation, est résolu en intercalant un interacteur de présentation dans la hiérarchie constructive (voir figure 3).

Passer à la réalisation d'applications utilisant le modèle des Interacteurs Hiérarchisés suppose de choisir un *mode de contrôle* des interacteurs, ainsi que de définir des *règles de visibilité* entre Interacteurs.

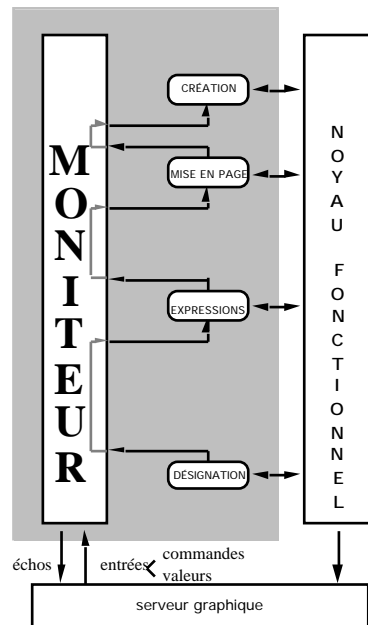
Afin de respecter la règle de faible couplage préconisée par le Génie Logiciel [PIE 91], notre implémentation met en œuvre un contrôle externe à l'aide d'un moniteur qui connaît (et lui seul) tous les interacteurs. Bien qu'il soit possible d'autoriser l'utilisateur à modifier la hiérarchie, nous utilisons une hiérarchie fixe, qui nous semble plus en rapport avec les possibilités cognitives du concepteur. Chaque interacteur est spécifié dans un langage formel déclaratif. Cette spécification est compilée, pour générer l'ensemble des unités Ada qui constituent le contrôleur de dialogue. Après compilation de ce code généré, l'édition de liens avec l'application génère le système de CFAO. Cette implémentation permet d'effectuer aisément des modifications de l'architecture du dialogue.

#### 4. Conclusion

La conception d'interfaces utilisateur nécessite à la fois de respecter les grands principes du Génie Logiciel (abstraction, encapsulation et faible couplage [PIE 91]), et de satisfaire aux exigences de la psychologie cognitive en permettant à l'utilisateur de parcourir en pré-ordre l'arbre de ses buts. Le modèle des Interacteurs Hiérarchisés, proposé dans cet article, satisfait ces deux exigences. Il permet même, ainsi que nous l'avons montré, de *dériver* la spécification du contrôleur de dialogue à partir de l'analyse des tâches du système.



Le modèle cognitif...



et sa traduction architecturale

**Figure 3. Modèle cognitif et Interacteurs Hiérarchisés**

**5. Bibliographie**

- [COU 90] COUTAZ J. *Interfaces homme-ordinateur, Conception et réalisation*, Dunod-Informatique, 1990.
- [MEI 91] MEINADIER J.P., *L'interface utilisateur, pour une informatique conviviale*, Dunod, 1991.
- [NOR 86] NORMAN D. et al., *User Centered System Design*, Lawrence Erlbaum Associates, 1986.
- [PAR 72] PARNAS D., « On the criteria to be used in decomposing systems into modules », *Comm. ACM*, 15, 1972, ACM Press, pp. 1053-1058.
- [PIE 91] PIERRA G., *Les bases de la programmation et du génie logiciel*, Dunod-Informatique, 1991
- [SHN 82] SHEIDERMAN B., « Direct Manipulation : A Step Beyond Programming Languages », *IEEE Computer*, 16, 8, 1983, IEEE Pub., pp. 57-69.
- [WIL 93] WILSON L. et al., *Langages de programmation comparés*, Addison-Wesley, 1993.