
CONCEPTION, REALISATION ET EVALUATION D'INTERFACES A PARTIR DES SPECIFICATIONS CONCEPTUELLES

Elizabeth FURTADO

*Département de recherche en Informatique, Automatique et Mécatronique
DIAM/IUSPIM, Domaine universitaire de Saint-Jérôme
Université d'Aix Marseille III - Av. Escadrille Normandie Niemen*

RESUME

Cet article décrit une méthode de conception, de réalisation et d'évaluation d'interfaces pour des systèmes d'aide à la décision de procédés complexes. Les caractéristiques de ces interfaces sont les suivantes :

- 1 — adaptatives à l'opérateur; le processus d'adaptation consiste d'une part à sélectionner les assistances devant être fournies à l'opérateur et d'autre part à lui montrer ces assistances sous forme de représentations graphiques qui soient les plus adéquates possibles,*
- 2 — formalisées selon un modèle d'architecture d'interface; la structure de notre modèle d'architecture est formalisée par une grammaire hors-contexte qui spécifie précisément la décomposition structurelle des objets interactifs,*
- 3 — évaluées au niveau ergonomique; le processus d'évaluation de l'interface commence durant les phases de conception et de réalisation et continue après l'utilisation du système.*

1. Introduction

Les problèmes de développement des systèmes interactifs sont, en grande partie, dus à la complexité de la conception et de la réalisation des interfaces. De plus les interfaces ainsi développées s'adaptent mal aux besoins de l'utilisateur. En fonction de ces problèmes, on assiste aujourd'hui à une évolution des travaux qui portent sur la réalisation d'outils d'aide à la conception et à la réalisation d'interfaces homme-machine (outil IHM).

Notre domaine de recherche concerne le développement d'un outil IHM appliqué aux systèmes de supervision d'un procédé complexe et, plus spécifiquement, à leur système d'aide à la décision. Un système d'aide à la décision détermine des tâches d'assistance (telles que, alarmes, plan d'action, etc.) fournies à l'opérateur pendant la conduite et la surveillance d'un procédé. Notre outil aide à générer les interfaces pour ces tâches d'assistance de manière automatique à partir d'un modèle de tâche décrit en fonction du comportement de l'opérateur [Rasmussen, 1983]. Les interfaces générées sont adaptatives à l'opérateur, formalisées selon un modèle d'architecture d'interface et évaluées au niveau ergonomique.

Afin d'utiliser efficacement notre outil et d'orienter le concepteur lors du développement d'un système de supervision, nous avons élaboré une Méthodologie d'Assistance à la Conception, à la réalisation et à l'évaluation des Interfaces Adaptatives (MACIA) [Furtado, 1996]. Cette méthodologie est composée d'une méthode et de ressources. La méthode se réfère aux procédures qui doivent être suivies pendant les

phases de développement d'un système. Les ressources se réfèrent aux représentations nécessaires à la spécification des informations utilisées par la méthode.

La première partie de cet article concerne l'évolution future de l'interaction homme-machine et la description de la méthode MACIA. La deuxième partie concerne la description des liens entre l'approche de cette méthode et les sciences cognitives.

2. Problématique dans la supervision

La complexité des procédés industriels ainsi que la variété des conditions de leurs exploitations (intervention, maintenance, pannes, ...) augmentent continuellement. Face à ce problème, le travail de l'opérateur devient plus difficile à réaliser puisqu'il reçoit du système de supervision une grande quantité d'informations. Afin d'aider l'opérateur on trouve de plus en plus dans les salles de contrôle et de supervision, des systèmes d'aide à la décision qui l'assistent dans ses tâches. Ces systèmes exploitent certaines données en provenance du procédé pour lui fournir une aide au diagnostic, à la correction, à la reprise de défauts, etc.

Il est clair que cette solution soulève des problèmes de communication entre l'opérateur et ces systèmes, puisqu'ils peuvent lui fournir plusieurs assistances à la fois. Pour améliorer cette communication, il est nécessaire de construire des interfaces qui prennent en compte les aspects humains particulièrement cruciaux pour le contrôle de procédé. Grâce à cette prise en compte, il est possible d'obtenir des interfaces, qui d'une

part soient capables de s'adapter au modèle cognitif de l'opérateur lors de la réalisation de ses tâches et d'autre part respectent des principes ergonomiques (tels que la limitation du nombre d'objets impliqués dans un état du système, le fait de montrer seulement les informations qui sont nécessaires et immédiatement utilisables, etc.).

Nous avons analysé les interfaces générées par les divers outils IHM tant dans le domaine de la supervision ([Tendjaoui 1992], [Moussa 1992] et [Ujita 1992]) que dans celui des systèmes informatiques en général ([Normand 1992], [Bodart 1994], [Tarby 1993], [Petoud 1990], [De Baar 1992] et [Foley 1991]). Cette analyse nous a permis de constater que des efforts sont faits pour intégrer des caractéristiques d'adaptation aux interfaces générées et des recommandations ergonomiques dans le processus de développement d'interfaces. Cependant, les interfaces générées s'adaptent mal aux besoins de l'utilisateur, car elles ne s'appuient sur aucun modèle cognitif. L'intégration de recommandations ergonomiques dans le processus de développement d'interfaces est laissée au soin du concepteur. De plus, ces outils entraînent des difficultés d'implémentation de la structure et du comportement des interfaces, car ils n'utilisent aucun modèle d'architecture. Or, l'absence d'un modèle d'architecture guidant la construction de l'interface sur le plan architectural peut occasionner des aberrations aussi bien sur les aspects ergonomiques que sur les aspects structuraux, avec le risque d'une mauvaise décomposition modulaire qui peut entraver la mise au point itérative de l'interface.

3. Interaction opérateur - système de supervision

L'interaction homme - machine recouvre l'étude des phénomènes cognitifs, logiciels et matériels intervenant dans l'accomplissement des tâches au moyen d'un ordinateur et de ses systèmes informatiques [Coutaz 1993].

Dans le cadre de l'interaction opérateur - système de supervision, le modèle cognitif d'analyse du comportement de l'opérateur le plus répandu est le modèle de Rasmussen [Rasmussen 1983]. Ce modèle comprend quatre étapes de traitement d'informations : la détection d'événement, l'évaluation de la situation, la prise de décision et l'action. Rasmussen distingue trois niveaux de comportement de l'opérateur : le niveau d'entraînement (*skill-based behavior*), le niveau de règles (*rule-based behavior*) et le niveau du savoir (*knowledge-based behavior*). Le premier comportement signifie que l'opérateur réalise des opérations rapides du type stimuli-réponse. Ces opérations sont à la fois sensori-motrices et cognitives. Le second comportement signifie que l'opérateur applique un ensemble de règles pour résoudre un problème. Le troisième comportement

est perçu lorsque l'opérateur est amené à raisonner face à des situations nouvelles.

Due à l'évolution rapide de divers facteurs (tels que des nouvelles technologies, de l'automatisation des procédés, ...), les rôles de l'opérateur et du système changent. L'opérateur est de moins en moins impliqué dans les tâches de contrôle manuel et doit réaliser des tâches complexes de résolution de problèmes. Le système de supervision possède de plus en plus une autonomie complète pendant le fonctionnement normal du procédé, mais ne lui permet pas de traiter les situations non familières. D'après Rasmussen, la machine devrait englober le niveau d'entraînement et partiellement le niveau de règles. D'autres [Ujita 1992] estiment même que l'automatisation englobera totalement le niveau de règles et partiellement le niveau du savoir. Sans vouloir définir précisément les limites de l'intervention de la machine dans ce processus, nous pensons qu'un rôle essentiel des interfaces homme - machine est de conduire le mieux possible le glissement de tâches entre ces deux agents. Autrement dit, les interfaces doivent déterminer et montrer précisément le moment où la machine arrête l'intervention pour que l'opérateur intervienne.

Les informations nécessaires à l'interface (telles que les conditions de déclenchement d'une tâche par l'opérateur ou par le système, les conditions d'intervention de l'opérateur au cours de l'exécution d'une tâche, etc.) peuvent être déjà définies lors de la description des tâches de l'opérateur (spécifications conceptuelles). La technique de génération automatique d'interfaces à partir des spécifications conceptuelles demeure l'objet de recherches ambitieuses. La motivation est double : i) assurer la cohérence entre les spécifications et l'implémentation et; ii) obtenir un gain de temps dans le développement d'interfaces.

Dans le paragraphe suivant, nous présentons une méthode qui décrit un processus de développement d'interfaces construites à partir des spécifications conceptuelles.

4. Méthode "Macia"

La Figure 1 montre les phases de la méthode MACIA et ses ressources associées : les modèles et les règles.

4.2. Analyse du domaine

Cette phase est précédée d'une étude visant l'identification des problèmes existants, des besoins des utilisateurs et des objectifs à atteindre. A la fin de cette étude, le concepteur peut identifier les données et les tâches du domaine de l'application.

4.2. Modélisation

Cette phase se réfère à la modélisation des informations obtenues dans l'analyse, conduisant à la définition des modèles de l'opérateur, du procédé et de la tâche.

4.2.1. Modèle de l'opérateur

Les opérateurs ont des niveaux d'expérience, de connaissances et des besoins différents. Afin que l'interface s'adapte à ces particularités, il est nécessaire de modéliser les opérateurs.

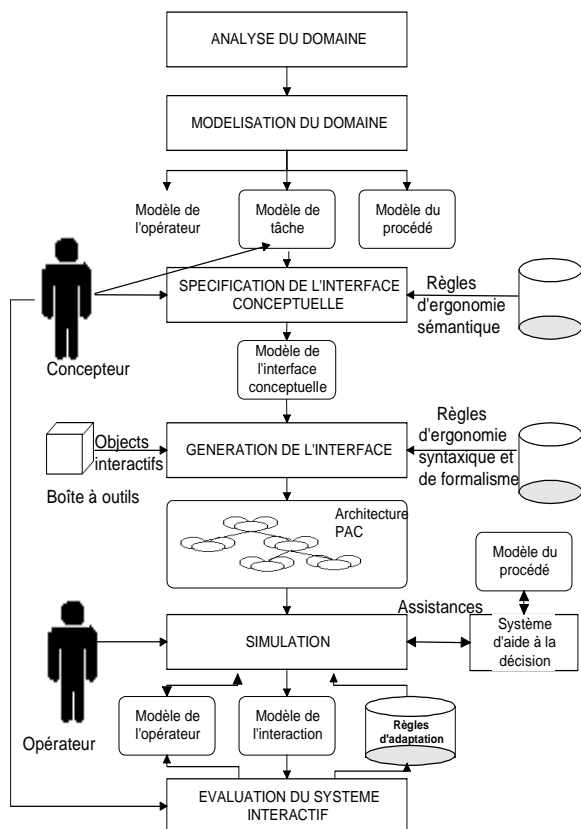


Figure 1— La méthodologie MACIA

Notre modèle de l'opérateur est composé d'une partie statique (le nom, l'âge, le type de l'opérateur...) et d'une partie dynamique, constituée d'informations relatives à l'interaction "opérateur-système" lors du traitement d'un événement anormal (défaut sur le procédé). A chaque événement anormal, nous enregistrons sa fréquence d'occurrence. Pour chacune de ces occurrences, nous enregistrons également la date et heure du début et fin du traitement de l'événement anormal et la séquence d'actions qu'il a effectué lors du traitement.

Ces informations sont utilisées pour déterminer le stéréotype de l'opérateur relatif à un événement anormal. Les facteurs qui constituent le stéréotype sont les suivants : la fréquence d'interaction avec laquelle l'opérateur traite un événement anormal, la préférence de l'opérateur concernant le nombre d'assistances à

recevoir, la complexité d'un événement anormal et la familiarité avec le procédé surveillé.

Au fur et à mesure que l'opérateur acquiert de l'expérience sur le traitement d'un événement anormal, son stéréotype devra changer. Le stéréotype de l'opérateur permet de déduire son niveau d'expérience (novice, expérimenté et expert). Le niveau d'expérience est utilisé pour choisir aussi bien quelles tâches d'assistance doivent être affichées à l'opérateur (*le quoi montrer*), que la façon dont ces tâches doivent être affichées (*le comment montrer*).

4.2.2. Modèle du procédé

Un procédé est composé d'un ensemble de matériels qui ont des grandeurs physiques associées (variables) [Benzian 1993]. Par exemple, dans le contexte d'une centrale thermique, nous citons des matériels (brûleur, turbine) et des variables (température, pression). Ce modèle est composé de la description de la structure du procédé (les entrées, les sorties,...) et de son mode de fonctionnement (par exemple, la finalité du système). Ce modèle permet la mise en œuvre des outils d'assistance spécifiques (tels que, les alarmes, les plans d'actions...).

4.2.3. Modèle de tâche

Le modèle de tâche sert à décrire les tâches effectuées par l'opérateur, celles effectuées par le système et celles effectuées par l'opérateur et le système. Ces tâches, organisées d'une façon hiérarchique, peuvent être spécifiques de l'application (telle que l'intervention sur un procédé) ou spécifiques de l'interface (telle que le traitement d'un fichier).

La figure 2 illustre un exemple du modèle de tâche de la tâche de surveillance d'un procédé, appelée *Surveiller Procédé*. Le but de cette tâche est d'afficher la structure hiérarchique du procédé trouvée à partir des entrées de l'opérateur sur les attributs clés d'une entité du procédé. La structure hiérarchique du procédé qui doit être affichée dépend des modes de recherche d'un nœud dans une structure hiérarchique. Les modes de recherche possibles sont déterminés par les sous-tâches de la tâche *Trouver procédé*. Par exemple, la sous-tâche *Trouver père* permet de rechercher un nœud du niveau hiérarchique supérieur de celui des attributs clés.

Toutes les caractéristiques des tâches modélisées doivent être décrites dans le formulaire montré dans la partie droite de cette figure. Une tâche est identifiée par un numéro, un nom, un type (d'assistance, d'interface, etc.) et la fonction (de saisie, d'affichage, etc.). Chaque tâche est associée à un but et possède des pré-conditions, post-conditions et paramètres. Un paramètre correspond à une liste de données du système qui va éventuellement être modifiée par l'exécution de la tâche (paramètre d'entrée) ou qui a été modifié (paramètre de sortie). Particulièrement, lors de la définition des

paramètres, le concepteur attache les données de l'application aux paramètres des tâches. A un moment, le concepteur peut découvrir une donnée qui n'a pas encore été définie, ce qui le ramène à la phase d'analyse du domaine afin d'analyser de nouveau le procédé. Cette façon de parcourir les phases de la méthode MACIA met en évidence son caractère incrémental, dont les phases ne sont pas décrites dans une séquence rigide.

Une tâche peut être élémentaire ou composée. Une tâche élémentaire est une tâche non décomposable dont le corps prend la forme d'une action. Une tâche composée est une tâche dont le corps exprime la relation entre ses sous-tâches au moyen d'un constructeur. Un *constructeur* permet de définir si les sous-tâches sont séquentielles (SEQ), alternatives (ALT), parallèles (PAR) ou simultanées (SIM). De plus, une tâche peut posséder des attributs booléens décrivant le fait qu'elle est facultative, répétitive, déclenchée par l'opérateur ou par le système, etc.

4.3. Spécification de l'interface conceptuelle

Une fois le modèle de tâche construit, nous pouvons en déduire l'organisation des interfaces et leur ordre d'exécution. Dans cette phase, nous décrivons la démarche que nous adoptons pour générer les interfaces à partir d'un modèle de tâche. Elle repose sur la construction automatique d'un modèle de l'interface conceptuelle (MIC). Ce modèle est composé des composants suivants [Santoni 1995] : les espaces d'interaction, les sous-espaces d'interaction et les objets d'interaction. Un espace d'interaction est le lieu d'activité virtuel offrant les éléments nécessaires à la réalisation d'une ou plusieurs tâches de l'utilisateur [Normand 1992].

Un espace d'interaction constituera une fenêtre. Un sous-espace d'interaction décrit une vue sur les données et les tâches du système. Un sous-espace d'interaction décrit une vue sur les données et les tâches du système et constituera une sous-fenêtre. Un objet d'interaction est un élément de l'interface conceptuelle et permet le dialogue avec l'opérateur.

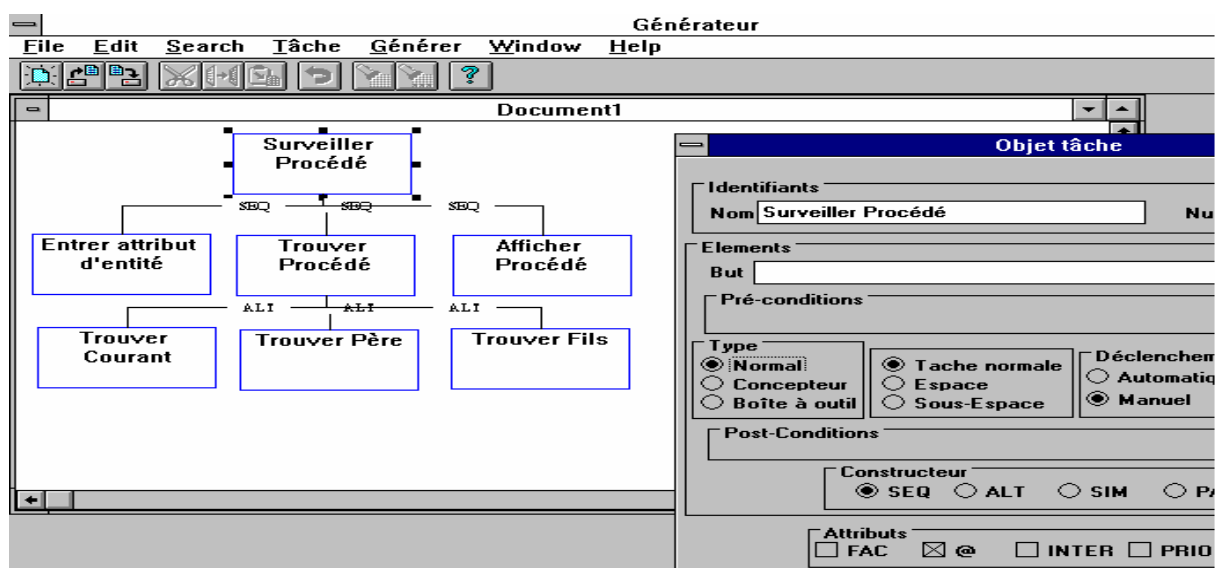


Figure 2 - Un exemple d'une tâche et ses sous-tâches pour une application

L'organisation structurelle du modèle de l'interface conceptuelle est similaire à celle du modèle de tâche : on obtient la décomposition des composants MIC et leur ordre d'exécution à partir de l'organisation des tâches. Cela permet au concepteur d'envisager la manière dont les informations seront affichées dans l'interface qui va être générée. Cependant il faut observer que le modèle conceptuel ne contient que les informations qui doivent être présentées sans aucune spécification sur la manière dont cette présentation sera effectuée. Autrement dit, on ne tient pas encore compte des objets interactifs qui composent l'interface (tels que boutons, menus).

La Figure 3 présente le MIC associé au modèle de tâche de la Figure 2. Le sous-espace *Entrer attribut d'entité* gère l'exécution des tâches *Entrer attribut d'entité*, *Trouver* et de ses sous-tâches. Ces tâches sont groupées dans le même sous-espace car le nombre de leurs paramètres est inférieur à 9.

La tâche interactive *Afficher procédé* est séparée (générant un sous-espace), car son paramètre *procédé* est graphique. Un paramètre graphique occupe tout un sous-espace, à l'exception de la demande explicite du concepteur. Le sous-espace *Entrer attribut d'entité* est composé de cinq objets d'interaction : *Entrée Type entité*, *Entrée Nom entité*, *Clic Trouver courant*, *Clic*

Trouver père et *Clic Trouver fils*. Les deux premiers sont les paramètres d'entrée de la tâche *Entrer attribut d'entité*, et ils constituent les attributs clés d'une entité. Les trois autres représentent les sous-tâches de la tâche *Trouver procédé*. Le mot *Clic* de ces objets signifie que ces sous-tâches sont déclenchées par l'opérateur.

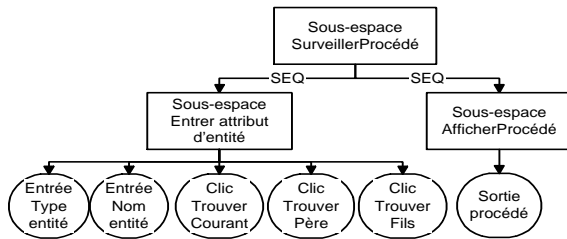


Figure 3 — Le MIC associé au modèle de tâche précédent

La construction automatique du MIC est faite à travers l'application de règles ergonomiques sur les caractéristiques des tâches modélisées. Dans les paragraphes suivants nous décrivons la stratégie utilisée pour déterminer ces règles.

Dans un premier temps nous avons sélectionné des recommandations ergonomiques de facteurs humains qu'on trouve tant dans le domaine des systèmes interactifs [Scapin 1990] que dans le domaine des systèmes de supervision [Kolski 1993]. Les recommandations sélectionnées sont classées en 4 catégories :

- *spécifiques des caractéristiques des tâches* : ces recommandations portent sur les objectifs des opérateurs.
- *d'aide à l'organisation de vues* : les recommandations qui se trouvent dans cette catégorie comportent des conseils ergonomiques nécessaires pour définir, simplifier et alléger des vues.
- *visant à limiter les actions à parcourir* : les recommandations qui se trouvent dans cette catégorie concernent la limitation autant que possible des étapes par lesquelles l'opérateur doit passer pour atteindre un but.
- *visant à faciliter la navigation* : ces recommandations portent sur la facilité d'utilisation du système. L'ordre d'exécution d'une action doit être compatible avec l'ordre tel que se l'imagine l'opérateur.

Dans un deuxième temps nous avons choisi certaines caractéristiques des tâches. Les caractéristiques auxquelles nous nous sommes intéressés portent sur : le but de la tâche, les attributs concernant la spécification d'interface (la participation de l'opérateur, la nature de la tâche, le type de la tâche, le type de ses pré-conditions), la définition des paramètres et l'ordre

d'exécution (le constructeur et la position hiérarchique de la tâche).

Dans un troisième temps nous avons appliqué les recommandations sélectionnées sur les caractéristiques des tâches pour déduire les règles ergonomiques de construction du MIC. Les règles déduites sont de deux types (Tableau 1 de la page suivante) :

- *de définition de la structure du MIC* : ces règles réalisent un découpage automatique de tâches en espaces d'interaction et en sous-espaces d'interaction. Ces règles concernent la création d'espaces, la création de sous-espaces, l'optimisation de la structure MIC (telle que l'élimination de sous-espaces) et la création d'objets d'interaction.
- *de définition de la logique de fonctionnement du MIC* : ces règles consistent à définir l'ordre de séquence du dialogue et les types de navigation entre les espaces.

Une fois le modèle de l'interface conceptuelle construit, le concepteur peut en améliorer les descriptions, jusqu'à ce qu'il soit satisfait. Il peut effectuer les opérations suivantes : éliminer un sous-espace du dernier niveau de la hiérarchie MIC, changer le mode de déclenchement des espaces et sous-espaces, modifier le libellé des composants MIC, demander le recouvrement d'une tâche dans le même sous-espace, éliminer des objets d'interaction d'un sous-espace, etc. Pour maintenir la cohérence entre les spécifications conceptuelles et l'interface générée, le concepteur ne peut pas insérer un composant MIC. Il doit réaliser les insertions dans le modèle de tâche, ce qui le ramène à l'étape de modélisation de tâches.

4.4. Génération de l'interface

Cette phase concerne la mise en œuvre des interfaces à travers la génération automatique du modèle d'agents de l'interface.

Ce modèle possède une structure hiérarchique similaire au modèle d'architecture de l'interface PAC [Coutaz 1991]. La structure du modèle d'agents de l'interface est une composition de trois types d'agents PAC : espace, sous-espace et objet d'interaction. Un agent est composé des trois parties suivantes :

- la partie *abstraction* représente, de manière simplifiée, les éléments abstraits sous-jacents à une présentation (par exemple, la valeur d'une chaîne affichée),
- la partie *contrôle* est responsable du maintien de la cohérence entre les objets affichés et les objets de l'application et assure le traitement des événements. Un événement est un signe ou une condition qui,

lorsqu'il est vérifié, peut changer le comportement de l'objet concerné,

- la partie *présentation* a pour rôle de mettre en œuvre la présentation physique de l'interface, ainsi que l'interception des interactions de l'utilisateur sur les périphériques d'entrée de la machine.

La partie centrale de la Figure 4 de la page suivante détaille le modèle d'agents de l'interface pour le sous-espace *Entrer attributs d'entité*. Ce modèle d'agents correspond au MIC de ce sous-espace présenté précédemment (Figure 3).

Tableau 1 - Les ressources qui influent la création du MIC

Type de recommandation	caractéristiques destâches	types de règles de construction du MIC	Exemples de règles
spécifiques à la tâche	but de la tâche	création d'espaces	<i>si une tâche définit un but principal, il faut créer un espace, dont le nom est celui de la tâche</i>
d'aide à l'organisation de vues	définition des paramètres et attributs concernant l'interface	création de sous-espaces et création d'objets d'interaction	<i>si le nombre de paramètres dans un sous-espace est < 9, alors la tâche courante peut être groupée dans ce sous-espace</i>
visant à limiter les actions à parcourir	ordre d'exécution	optimisation de la structure du MIC	<i>si un sous-espace est composé seulement d'un sous-espace, alors il faut éliminer son composant</i>
visant à faciliter la navigation	ordre d'exécution	définition de la logique de fonctionnement du MIC	<i>si on a le choix entre deux espaces, alors il faut permettre l'accès à ces espaces en même temps, mais un seul espace pourra être déclenché</i>

A gauche du modèle d'agents, la Figure 4 illustre l'interface qui sera fournie à l'opérateur. Chacun des composants de l'interface est attaché à la partie présentation des agents du modèle. A droite du modèle, cette figure montre les données de l'application qui représentent la partie abstraction des agents du modèle. Par exemple, nous avons montré la description de la classe *Entité*, ensuite la description de l'instance de cette classe qui est affichée dans l'interface. La présentation de ce sous-espace est représentée par une sous-fenêtre. Cette sous-fenêtre est composée de cinq objets interactifs. Chacun de ces objets représente la partie présentation des agents *objets d'interaction*. La partie abstraction de ces agents se réfère soit aux objets de l'application (par exemple, *Nom d'entité* et *Type d'entité*), soit aux tâches de l'application (par exemple, *Trouver courant*, *Trouver père* et *Trouver fils*).

Nous présentons, maintenant, les informations des parties abstraction, contrôle et présentation d'un agent. La partie abstraction est constituée d'attributs des données de l'application et des concepts du MIC. La partie contrôle est constituée de méthodes qui gèrent les enchaînements entre les espaces et entre les sous-espaces. Chaque partie contrôle de chaque agent n'a droit de regard que sur ses informations propres concernant des enchaînements (tels que, par exemple,

l'ordre de séquence du dialogue). Ce sont les autres agents qui prennent en compte la gestion de leurs informations associées. Cela est possible grâce à la structure hiérarchique du modèle d'agents de l'interface. La partie présentation est représentée par une liste des objets interactifs. Cette liste est constituée de pointeurs dirigés vers les objets interactifs qui représentent les agents au niveau externe.

Dans cet article, nous décrivons plus précisément la génération de la partie présentation des agents. Il importe que cette génération respecte aussi bien les relations entre agents définies à travers les règles de formalisme que les critères ergonomiques syntaxiques à travers les règles de présentation de l'interface.

4.4.1. Règles de formalisme

La structure du modèle d'agents de l'interface est formalisée par une grammaire hors-contexte où sont définis les groupes de présentation qui contiennent plusieurs objets interactifs. A partir de cette grammaire, nous avons spécifié des règles de formalisme qui permettent de :

- définir la relation entre les agents : à travers la lecture de la grammaire du formalisme nous avons distingué quatre types de relations. Nous les décrivons à travers les exemples :

- relation de désignation : par exemple, si (*la partie contrôle = espace*) alors (*la partie présentation = fenêtre*),
- relation de composition : par exemple, si (*la partie présentation de l'espace = fenêtre*) alors (*la partie présentation du sous-espace = sous-fenêtre de contrôle ou sous-fenêtre graphique*),

- choisir le groupe de présentation : ces règles permettent d'indiquer le groupe qui contient les objets interactifs les plus appropriés aux objets d'interaction. Par exemple, si (*l'objet d'interaction fournit plusieurs options d'entrée à l'opérateur, mais il ne peut choisir qu'une option*) alors (*le groupe-présentation = interacteur-choix-unique*).

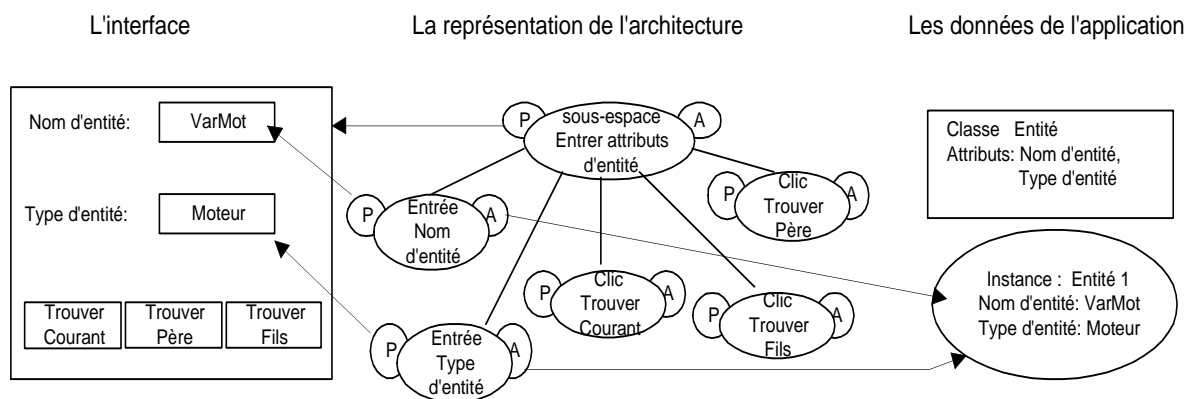


Figure 4 - Le modèle d'agents de l'interface pour le sous-espace "Entrer attributs d'entité"

Ce groupe est composé des objets interactifs : bouton radio, liste extensible, etc. L'objectif de l'association entre un objet d'interaction et un groupe a été de construire une base de règles ergonomiques de présentation de l'interface (voir §4.4.2) par groupe de présentation. Cette finesse de groupement nous permet de :

- diminuer l'écart entre les concepts de l'interface conceptuelle et les concepts de l'interface réelle, en considérant les sélections qui ont déjà été réalisées sur les objets d'interaction MIC,
- déterminer le plus efficacement à quel objet interactif il est le plus pertinent d'attacher chaque recommandation ergonomique,
- diminuer le temps de recherche d'objet interactif approprié à l'objet d'interaction.

4.4.2. Règles ergonomiques de présentation de l'interface

Les règles ergonomiques de présentation de l'interface concernent les critères ergonomiques syntaxiques qui permettent soit la sélection des objets interactifs soit la définition des attributs de présentation de ces objets (par exemple, la taille, la couleur d'une fenêtre). Nous nous sommes inspirés sur les recommandations ergonomiques de [Bodart 1994], [Vanderdonck 1993], [Moussa 1992] et [Kolski 1993] pour définir les bases de règles ergonomiques qui illustrent les attributs véhiculés par ces règles. Prenons l'exemple d'une règle qui permet de découvrir l'objet interactif du groupe de présentation proposé précédemment *interacteur - choix - unique* : si (*le*

numéro de valeurs possibles d'un objet d'interaction < 5) alors (*l'objet interactif = bouton radio*).

4.5. Evaluation de l'interface

Cette phase a pour objectif principal de vérifier que le système interactif donné aux opérateurs répond à leurs besoins recensés lors des premières phases du projet. L'évaluation du système concerne d'abord une exploitation du système réalisée par l'opérateur, ensuite une évaluation de cette exploitation.

4.5.1. Simulation

Imaginons le scénario où pendant que l'opérateur est en train de réaliser une tâche de surveillance, il arrive un événement anormal. Notre outil dispose d'un mécanisme, appelé gestionnaire d'interaction, pour calculer les informations dynamiques du modèle de l'opérateur concernant cet événement. Il applique les règles d'adaptation suivantes pour découvrir :

- *le mode de représentation des tâches*, par exemple, si (*assistance est diagnostic superficiel, la situation du procédé est anormale et le niveau d'expérience de l'opérateur est novice ou expérimenté*) alors (*le mode de représentation = textuel*),
- *les valeurs des facteurs du stéréotype de l'opérateur*, par exemple, si (*l'opérateur a déjà résolu le même événement plusieurs fois*) alors (*la fréquence de confrontation avec l'événement = haute*).

Pendant l'interaction, le gestionnaire de l'interaction enregistre toutes les actions physiques effectuées par l'opérateur dans l'ordre où il les exécute dans un modèle

de l'interaction [Balbo 1992]. Les informations concernant une action physique se réfèrent généralement à : la tâche choisie, les paramètres d'entrées associés à la tâche, les dates de début et de fin d'exécution de la tâche, etc.

4.5.2. Evaluation

Le processus d'évaluation de l'interface commence durant les phases de conception et de réalisation et se poursuit après l'utilisation du système. En ce qui concerne la phase de conception, à partir de 23 recommandations ergonomiques de facteurs humains, nous avons déduit 26 règles d'ergonomie sémantique (de construction du MIC). Nous citons notamment la recommandation suivante :

Le chiffre 7 ± 2 est lié à la capacité de notre mémoire à court terme, et justifie par exemple, le contrôle réalisé sur la quantité d'informations présentées à l'utilisateur dans un même formulaire. [Beaudouin 1991]

L'utilisation de ces recommandations permet une évaluation de l'interface au niveau sémantique, où l'interface est encore traitée en termes de données et tâches du domaine.

En ce qui concerne la phase de génération, à partir de 40 critères ergonomiques syntaxiques, 50 règles d'ergonomie syntaxique (de présentation de l'interface) ont été déduites. Nous citons notamment la recommandation suivante : "*si le type de message est d'erreur alors l'alignement du message dans la sous-fenêtre est centralisé et la couleur est rouge* [Scheinerman 1992]". L'utilisation de ces recommandations permet une évaluation de l'interface aux niveaux d'abstraction syntaxique et lexical, où l'interface est traitée en termes de commandes et objets interactifs.

Le processus d'évaluation de l'interface après l'utilisation du système est fait au travers du modèle de l'interaction. Les actions physiques de l'opérateur sont comparées aux facteurs liés à sa phase cognitive et à la situation de gravité du procédé en face d'un événement anormal. Par exemple, si (*la situation du procédé n'était pas grave et l'opérateur a fait l'annulation immédiate des tâches d'assistance affichées*) alors (*il y avait une incohérence entre les tâches d'assistance et le comportement de l'opérateur basé sur l'entraînement*).

5. Sciences cognitives et approche "conception - réalisation - évaluation"

Comme nous l'avons déjà décrit, notre méthode pour le développement des interfaces est basée sur l'approche "conception - réalisation - évaluation". Dans le paragraphe suivant, nous étudions les techniques de

conception et d'évaluation des interfaces en soulignant l'apport des sciences cognitives.

5.1. Conception d'IHM

Depuis quelques années, il est de plus en plus nécessaire d'intégrer les facteurs humains (tels que les besoins de l'utilisateur, son comportement, ses connaissances, ses efforts, sa surcharge de travail) dans la conception d'interfaces. L'objectif aujourd'hui est de fournir des techniques permettant cette intégration. Pour ce faire, plusieurs études ont été menées dans les domaines de la psychologie cognitive, de l'ergonomie et du développement des méthodes de conception de systèmes informatiques.

5.1.1. Psychologie cognitive

La psychologie cognitive étudie les processus cognitifs d'un individu mis en jeu dans l'accomplissement de tâches à travers les modèles et les théories. Un des modèles qui a influencé la conception d'interfaces est le modèle de Rasmussen. C'est un modèle explicatif de traitement des informations et notamment des erreurs de l'individu. Un autre travail important est la théorie de l'action de Norman [Norman 1986], laquelle définit des stratégies pour la décomposition hiérarchique d'une tâche en buts et sous-butts jusqu'à ce que les sous-butts correspondent à des actions élémentaires non décomposables.

Ces deux travaux ont été utilisés dans notre outil de la façon suivante : notre outil propose au concepteur un *modèle de tâche initial* où les tâches hiérarchiquement modélisées représentent les tâches décisionnelles de résolution de problème, définies dans le modèle de Rasmussen. L'objectif est de fournir au concepteur un modèle qui lui permet de mieux comprendre et valider le modèle cognitif de l'opérateur. A partir de ce modèle initial, le concepteur génère un modèle de tâches pour sa propre application, en rajoutant et/ou en enlevant des sous-tâches.

5.1.2. Ergonomie

L'ergonomie est l'étude scientifique des conditions de travail et des relations entre l'homme et la machine [Le Robert 1993]. Cette étude vise à réduire la distance entre le monde réel de l'utilisateur (son modèle mental) et le monde représenté par la machine [Norman 1986]. Pour cela, il est souhaitable que des recommandations ergonomiques soient utilisées dès la conception des systèmes informatiques. Ces recommandations consistent à dériver des aspects d'interface (e.g. style d'interaction, présentation des informations) à partir de facteurs humains (e.g. la nécessité d'une information déterminée). Au sein de notre générateur, l'analyse du domaine, la définition des besoins des utilisateurs et la spécification de l'interface conceptuelle constituent les principales étapes permettant d'intégrer l'ergonomie au

niveau de la conception. A travers les besoins des utilisateurs et du modèle de l'interface conceptuelle, les premières spécifications de l'interface (une maquette) sont obtenues en appliquant des recommandations ergonomiques de facteurs humains.

5.1.3. Méthodes de conception de systèmes informatiques

Les méthodes de conception peuvent être divisées en deux types [Coutaz 1988] : les méthodes qui s'appuient sur une théorie de la communication homme-machine et les méthodes qui s'appuient sur des expérimentations. La difficulté d'utilisation du premier type de méthodes découle du fait qu'il faut une connaissance approfondie de la psychologie de l'utilisateur, tandis que la démarche du deuxième type de méthodes repose sur l'évolution vers cette connaissance à partir d'observations. Ces observations concernent les éléments suivants : les utilisateurs, le domaine d'application et le matériel. Néanmoins, la diversité de ces composants rend difficile cette démarche.

Notre approche de "conception-réalisation-évaluation" concerne le deuxième type de méthodes et traite le problème de la diversité des composantes énumérées de la façon suivante :

- les utilisateurs sont classés selon leur niveau d'expérience et leur fréquence d'utilisation du système, de manière à ce qu'il soit possible de répondre à l'évolution de chacun à partir de ces informations,
- notre domaine d'application se situe dans le cadre des systèmes de supervision de procédé complexe et les tâches du domaine les plus fréquentes sont des tâches d'assistance,
- le traitement de la diversité du matériel consiste à choisir un style d'interaction qui soit approprié aux utilisateurs et aux tâches, c'est-à-dire, à choisir le mode de représentation des tâches d'assistance selon le niveau d'expérience de l'opérateur.

5.2. Evaluation d'IHM

L'évaluation de la qualité des interfaces peut être réalisée à deux moments : avant la mise en œuvre du prototype (a priori) et après l'utilisation du système (a posteriori).

Les modèles d'évaluation a priori s'appuient sur les théories fondamentales de la psychologie cognitive. Ces modèles sont capables d'évaluer la surcharge cognitive de l'utilisateur (par exemple, [Ujita 1992] utilise l'électrocardiogramme) et de prédire les performances de l'utilisateur (par exemple, le temps qu'il va mettre pour réaliser une tâche).

Les modèles d'évaluation a posteriori étudient les facteurs de qualité des interfaces liés à : i) l'ergonomie, tels que, la concision, la compréhensibilité et la rapidité; ii) la psychologie cognitive de l'utilisateur, tels que, la simplicité et la facilité d'utilisation d'une interface et iii) l'aide proposée par le logiciel [Palanque 1992]. [Balbo 1992] fait une comparaison entre un graphe de tâches de l'utilisateur et ses actions physiques. Cette comparaison permet de mettre en évidence des anomalies de conception dans l'interface, par exemple, une fenêtre qui serait ouverte à l'initiative du système, puis immédiatement fermée par l'utilisateur.

Notre approche d'évaluation hybride "a priori et a posteriori" a l'avantage d'intégrer des facteurs humains dans tous les niveaux d'abstraction d'un système interactif :

i) dans les niveaux syntaxiques et lexicaux, nous utilisons les règles ergonomiques de présentation qui permettent la définition de l'interface réelle;

ii) pour effectuer la correspondance entre les niveaux sémantique et syntaxique, nous utilisons les règles ergonomiques de présentation qui permettent la définition de l'interface conceptuelle;

iii) dans le niveau conceptuel, où les spécifications conceptuelles sont définies, le modèle de tâche initial assure que les spécifications conceptuelles soient centrées sur l'utilisateur et ses tâches.

6. Prototype

Actuellement, un prototype de notre outil est déjà implémenté. C'est un contrôleur de dialogue responsable de la gestion dynamique de l'interface au travers d'un contrôle hiérarchique des agents d'interfaces d'un système interactif. Le contrôleur du dialogue a été développé en C++ sous WINDOWS. Il est composé d'un ensemble de classes dont les fonctionnalités représentent la structure interne des *agents* et *tâches*. Les méthodes de ces classes consistent d'une part à gérer les enchaînements du dialogue, c'est-à-dire, à vérifier si les tâches peuvent être déclenchées à la suite d'une interaction de l'opérateur ou de tout autre événement survenant lors d'une session. D'autre part, ces méthodes consistent à gérer la mise en œuvre de la structure de l'interface. Cette gestion concerne l'initialisation, l'instanciation et la terminaison des objets interactifs qui font partie de la structure de l'interface.

Ce contrôleur du dialogue gère la dynamique d'interfaces d'une application particulière à partir de la hiérarchie des agents d'interface et des tâches de cette application, décrits sous forme d'objets C++.

7. Conclusion

Nous avons montré qu'en utilisant l'outil proposé et en suivant la méthodologie élaborée, on propose une assistance aux concepteurs durant le développement d'un système interactif. Cette méthodologie conduira à l'amélioration de la qualité du logiciel, à la limitation des erreurs et, en conséquence, à l'augmentation de la productivité.

Nous constatons que les fondements des sciences cognitives constituent les outils essentiels au développement de méthodes d'interaction homme-machine.

Notre travail de recherche apporte une double contribution : dans le domaine des systèmes homme-machine il s'agit d'un outil qui permet le passage d'un modèle de spécification d'interface conceptuelle vers un modèle d'architecture d'interface. Il contribue ainsi à la recherche sur la transition entre spécification et réalisation des interfaces. Dans le domaine de la supervision il s'agit d'un outil d'aide à la conception d'interfaces adaptatives à partir d'un modèle de tâche décrit en fonction du comportement de l'opérateur.

8. Limites et perspectives de recherche

Les perspectives de recherche concernent l'affinement de certains éléments intervenant dans les processus de spécification et d'adaptation d'interfaces.

Au niveau du travail de spécification réalisé, nous avons insisté sur le fait qu'un certain nombre de recommandations ergonomiques ont été choisies pour définir le MIC. Des règles ergonomiques ont donc été décrites. Notre intention, lors de la définition de ces règles était d'effectuer le mieux possible le groupement des informations manipulées dans une interface (ces groupements pouvant être incomplets, ...). Ces regroupements devraient être faits en considérant toute la hiérarchie et non seulement des branches d'arbre liées entre elles.

Au niveau du travail d'adaptation réalisé, nous avons proposé un mécanisme de détermination concernant les assistances à afficher et la manière de les afficher. Ce mécanisme repose aussi bien sur la notion de stéréotype de l'opérateur que sur un ensemble de règles adaptatives. Nous avons décidé que les valeurs qui déterminent les facteurs d'un stéréotype sont paramétrables par le concepteur, et qu'il peut aussi modifier les règles adaptatives. Si, d'une part, cette décision contribue à ce que le processus d'adaptation soit ajustable à un domaine d'application particulier, d'autre part, elle rajoute au concepteur le rôle d'évaluer l'utilisation du système par l'opérateur. Le réajustement automatique des valeurs et des règles nécessiterait pour être mis en œuvre l'implémentation d'un module spécifique.

Remerciements

Je remercie messieurs les professeur et maître de conférence Philippe François et Charles Santoni pour m'avoir encadré et aidé dans le développement de ce travail.

Cette recherche a été partiellement financée par le gouvernement brésilien : UNIFOR (Université de Fortaleza), UFCe (Université Fédéral du Ceara) et CAPES (Coordenacao de Aperfeicoamento de Pessoal de niveau superieur).

Références bibliographiques

[Balbo, 1992] S. Balbo, J. Coutaz, (1992), *Un pas vers l'évaluation automatique des interfaces homme-machine*, ERGO-IA. Biarritz.

[Beaudouin, 1991] M. Beaudouin - Lafon, (1991), *Interfaces Homme - machine : Vue d'ensemble et perspectives*. Actes du congrès Génie logiciel & Systèmes experts, Interfaces homme-machine, Maquettage & prototypage. Vol. 24, 4-16.

[Benzian, 1993] Z. Benzian, J.L. Ermine, C. Falinower et B. Bergeon, (1993), *Modélisation des connaissances et spécification de système de supervision de centrales thermiques*, Gr-Automatique, Journées d'étude *Sûreté, surveillance, supervision*.

[Bodard, 1994] F. Bodard, A. Hennebert, J. Leheureux, I. Provot, J. Vanderdonck, (1994), *A model-based approach to presentation: A continuum from task analysis to prototype*. Eurographics. 25-39.

[Coutaz, 1988] J.Coutaz, (1988), *Interface Homme-ordinateur: Conception et Réalisation*, Thèse de doctorat. Université Joseph Fourier.

[Coutaz, 1991] J. Coutaz, (1991), *Architectural design for user interfaces*, Proc of ESEC'91, European software engineering conference. 7-22.

[Coutaz, 1993] J.Coutaz, (1993), *Intelligence artificielle et interfaces homme-machine*. PRCA. 160-186.

[De Baar, 1992] D. de Baar, J. Foley, K.Mullet, (1992) *Coupling application design and user interface design*. In P. Bauersfeld, J. Bennett, G. Lynch. CHI'92, Addison-wesley, 259-266.

[Foley, 1991] J.Foley, W.C.Kim, S.Kovacevic, K.Murray, (1991), *UIDE, An intelligent user interface design environment*. In Sullivan & Tyler (Ed.), *Architectures for intelligent interfaces: elements and prototypes*, Reading MA : Addison-W., 339-385.

[Furtado, 1996] E.S. Furtado, (1996), *Conception, Réalisation et Evaluation d'interfaces à partir des Spécifications Conceptuelles*. Colloque Jeunes Chercheurs en Sciences Cognitives. Hyères, France.

[Kolski, 1993] Ch. Kolski, (1993), *Ingénierie des interfaces homme - machine*, Hermes.

[Le Robert, 1993] *Dictionnaire de la langue française*, (1993), Ed. Dictionnaires Le Robert, France.

[Moussa, 1992] F.Moussa, (1992), *Contribution à la conception ergonomique des interfaces de supervision dans les procédés industriels: application au système Ergo-Conceptor*. Thèse de doctorat. Université de Valenciennes.

[Norman, 1986] D.A. Norman, (1986), *Cognitive Engineering, User Centered Systems Design*. Lawrence Erlbaum Associates.

[Normand, 1992] V. Normand, (1992), *Le modèle SIROCO - de la spécification conceptuelle des interfaces utilisateur à leur réalisation*, Thèse de doctorat. Université de Grenoble I.

[Palanque, 1992] P. Palanque, (1992), *La modélisation d'objets coopératifs et interactifs*, Thèse de doctorat, Université de Toulouse.

[Petoud, 1990] I. Petoud, Y. Pigneur, (1990), *An automatic and visual approach for user interface design*. Engineering for Human-computer interaction. Elsevier science Publishers B.V. (North-holland). 403-421.

[Rasmussen, 1983] J. Rasmussen, (1983), *Skills, rules, knowledge; signals, signs, symbols and other distinctions in human performance models*, IEEE trans on SMC, SMC 13(3), 257-266.

[Santoni, 1995] Ch. Santoni, E. Furtado, Ph. Francois, (1995), *Towards adaptive UIMS for supervision systems*. Proc. IEEE. Vol 3, 2598-2603.

[Scapin, 1990] D.L. Scapin, (1990), *Organizing Human Factors Knowledge for the evaluation and design of interfaces*, Int journal of HCI. Vol 2, n°.3, 203-229.

[Shneiderman, 1992] B. Shneiderman, (1992), *Designing the user interface*. Addison Wesley.

[Tarby, 1993] Tarby J.C., (1993), *Le dialogue homme-machine: des spécification à sa gestion automatique*. Thèse de doctorat, Toulouse.

[Tendjaoui, 1992] M. Tendjaoui, (1992), *Contribution à la conception d'interface intelligente pour le contrôle de procédés industriels : application au module décisionnel d'imagerie*. Thèse de doctorat. Université de Valenciennes.

[Ujita, 1992] H.Ujita, (1992), *Human characteristics of plant operation and man-machine interface*, Reliability engineering and system safety. Vol 38, 119-124.

[Vanderdonckt, 1993] J. Vanderdonckt, F.Bodart, (1993), *Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection*, in Proc. INTERCHI'93, Amsterdam, 424-429.

