CE TEXTE COMPREND 1857 MOTS (BIBLIOGRAPHIE INCLUSE)

Article pour le Courrier du CNRS

Contribution : III.3.2. Interfaces Logicielles

Joëlle Coutaz, Professeur à l'Université Joseph Fourier de Grenoble,
LGI/IMAG URA CNRS 398
BP 53 X, 38041 Grenoble Cedex
tel. 76 51 48 54, email : joelle@imag.fr

Définition

Au sens large, une interface est un dispositif servant de limite commune à plusieurs entités communicantes. Dans le domaine de la communication homme-machine, une *interface logicielle* (ou interface homme-machine) est un ensemble de composants logiciels permettant à un ou plusieurs individus de communiquer avec un système informatique en vue d'accomplir une tâche. Un tel système, capable de communiquer en temps réel avec des utilisateurs, est dit *interactif*.

L'organisation des composants d'une interface homme-machine (IHM) obéit à des règles prescriptives. L'expérience montre que ces éléments directeurs ne suffisent pas à cerner la construction d'une interface. Des outils, véhicules de ces principes, sont nécessaires au concepteur et au réalisateur de logiciels interactifs. Parallèlement à la mise en place de techniques pratiques, la définition de concepts adaptés à l'évolution du domaine se poursuit dans les laboratoires de recherche. Eléments directeurs, outils et thèmes de recherche sont les trois thèmes présentés ici en liaison avec les interfaces logicielles.

Les éléments directeurs

La conception logicielle des systèmes interactifs et leur réalisation se heurtent à deux sources de difficulté : la diversité de l'environnement et l'expression des échanges. L'environnement inclut des applications aux besoins les plus divers, une grande variété de matériels et des utilisateurs qui ne cessent d'évoluer. Dans un système interactif, le caractère immédiat et informatif des échanges met à contribution tous les composants logiciels avec pour conséquence, le risque (ou la tentation)

de constituer un amalgame confus entre l'expression du comportement perceptible du système et la production des informations qui relèvent du domaine. Cet amalgame rend difficile, voire impossible, l'ajustement du système aux variations de l'environnement.

En génie logiciel, la complexité trouve son remède dans l'usage de la modularité. Si les principes de cette technique sont connus, leur application est difficile à maîtriser en l'absence de modèle. Jusqu'à ces dernières années, on ne disposait pas de modèle de référence pour la construction des systèmes interactifs. Sans cadre directeur pour appliquer la modularité, les réalisateurs ont créé des logiciels modulaires, certes, mais inadaptés à l'évolution itérative des interfaces.

Comme le montre la figure 1, les modèles d'architecture proposés actuellement pour les systèmes interactifs reposent tous sur la distinction entre les fonctions spécifiques à un domaine (les services) et les mécanismes de présentation (l'IHM). Nous appelerons désormais :

- *noyau fonctionnel* l'ensemble des services logiciels spécifiques à un domaine (par exemple, une base de données mémorisant les horaires d'avion),

- *interface homme-machine* le logiciel chargé de la communication d'informations entre le noyau fonctionnel et l'utilisateur (par exemple, la présentation des horaires sous forme de formulaires),

- *système interactif*, l'ensemble constitué d'un noyau fonctionnel relié à une interface (par exemple, un système de consultation d'horaires).

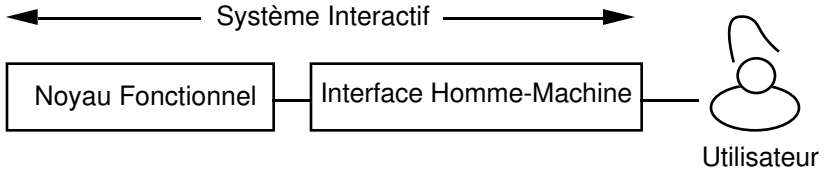

Figure 1 : Organisation schématique d'un système interactif

La distinction *noyau fonctionnel, interface homme-machine* sert de fondement à l'ajustement itératif du système : en théorie, la modification de la partie IHM peut être effectuée sans mettre en cause les services du noyau fonctionnel et réciproquement. Cette propriété est importante : en matière de conception, il faut pouvoir réviser des choix antérieurs. Par exemple, il faut être capable de remplacer facilement un formulaire par un menu.

Bien que valide en toute circonstance, le schéma de la figure 1 définit une organisation trop générale pour être directement exploitable par le concepteur de logiciel interactif. Des modèles multi-agents raffinent cette image en unités à la fois mieux adaptées aux techniques du génie logiciel mais aussi plus sensibles aux impératifs cognitifs de la communication homme-machine [Coutaz 90, Bass 91].

Les outils

Les outils pour la construction d'IHM se répartissent en trois catégories : les boîtes à outils, les squelettes d'application et les générateurs d'interfaces. La figure 2 en illustre la classification.

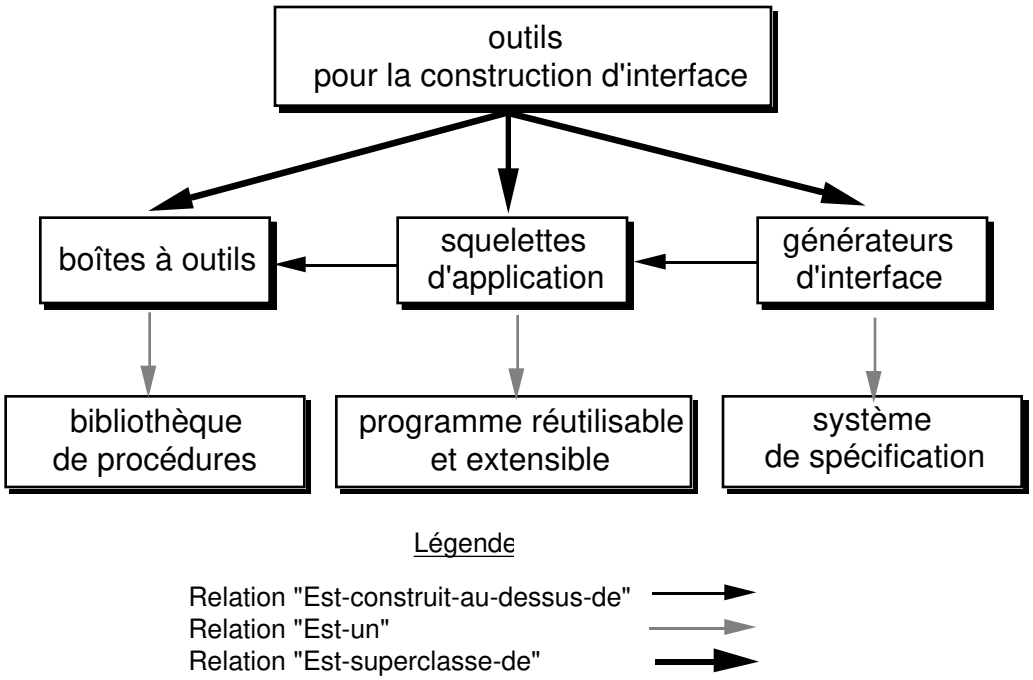

Figure 2 : Les classes d'outils pour la construction d'interfaces logicielles.

Une boîte à outils est une bibliothèque de procédures. L'éventail des fonctions offertes inclut la gestion des fenêtres, des événements tels que les clics souris ou la frappe de caractères sur le clavier, la production de figures géométriques simples comme les polygones et les cercles, la gestion des entités de dialogue tels les menus, les boutons poussoirs et les formulaires. Parmi les avantages d'une boîte à outils, nous retenons la définition d'un style d'interaction "normalisé" que l'utilisateur retrouve de système en système. Le style Macintosh est un bon exemple de ce phénomène. Parmi les inconvénients, nous notons l'absence de guide pour la conception logicielle et un apprentissage difficile (le programmeur doit découvrir le mode d'emploi).

Un squelette d'application réalise les fonctions usuelles de l'interface homme-machine sous la forme d'un logiciel réutilisable et extensible. La tâche du programmeur consiste à greffer sur le squelette les composants spécifiques à son système. Un squelette est construit au-dessus d'une boîte à outils dont il assemble les composants utiles à l'IHM d'une large classe de noyaux fonctionnels. Il a l'avantage de fournir au programmeur une architecture prête à l'emploi mais il ne masque pas la boîte à outils sous-jacente.

Un générateur d'interfaces crée une large part de l'IHM d'un système interactif à partir d'une spécification. Cette interface est ensuite reliée à un noyau d'exécution qui s'apparente à un squelette d'application. La spécification s'exprime dans un langage spécialisé comme UIL [OSF 89], ou bien s'effectue par manipulation directe au moyen d'un éditeur interactif comme Interface Builder de NeXT [Webster 89]. L'objectif premier du générateur est le prototypage rapide : permettre la création d'IHM à moindre coût, puis effectuer des tests et ajuster les choix. Contrairement aux boîtes à outils et aux squelettes destinés aux informaticiens, certains générateurs sont utilisables par des non informaticiens : le concepteur dessine son interface et le générateur produit le logiciel automatiquement. En réalité, les interfaces ainsi créées sont limitées par la base d'objets de présentation intégrée au générateur (typiquement les fenêtres, les menus, les boutons, les icônes et les boîtes de dialogue). Malheureusement, la généralité et l'extensibilité passent inévitablement par la programmation.

Les outils cités ci-dessus sont le résultat de travaux menés en recherche au cours de cette dernière décennie.

La recherche

La recherche actuelle en interface logicielle vise, pour l'essentiel, *l'intégration du savoir pluridisciplinaire*. On observe un fort rapprochement entre disciplines disjointes comme la psychologie cognitive, la linguistique et l'informatique. Ce rapprochement s'effectue dans le cadre de projets européens (et notamment le projet ESPRIT Basic Research Action AMODEUS) ou nationaux tels le programme cogniscience du CNRS et le PRC Communication Homme-Machine. L'objectif est d'intégrer aux concepts et techniques de l'informatique les résultats exploitables de l'ergonomie cognitive.

On observe également un rapprochement entre disciplines de l'informatique telles la communication en langue naturelle parlée et écrite (en génération comme en reconnaissance), la vision par ordinateur et la synthèse d'image, la kinesthésie (geste et retour d'effort). La mise en

commun de toutes ces modalités de la communication ouvre un champ nouveau dont on ne peut encore cerner les retombées socio-économiques. En particulier, on observe aux USA la percée fulgurante des réalités virtuelles et artificielles rendue possible par l'amélioration des performances du matériel [Krueger 90]. Ces systèmes reproduisent électroniquement un monde réel ou créent des environnements imaginaires en y intégrant le sujet humain. Ils s'appuient sur une communication multimédia et/ou multimodale qui combinent l'audition, la parole, la vision et le geste. La distinction entre le multimédia et le multimodal mérite d'être clarifiée.

Au sens général, un média tel l'ordinateur, est un dispositif technique servant de véhicule à l'information. En psychologie, la notion de modalité réfère aux catégories sensorielles humaines. Ainsi l'information transmise ou acquise par un média fait-elle appel à une modalité de la communication humaine. Un dispositif sera multimédia s'il sollicite simultanément plusieurs canaux sensoriels. Dans le cas des systèmes informatiques, il convient de raffiner ces définitions.

Comme l'indique la figure 3, il est proposé de situer les systèmes informatiques multimédia et multimodaux dans un espace à trois dimensions : pouvoir d'abstraction, fusion, et contrainte temporelle [Coutaz 92].

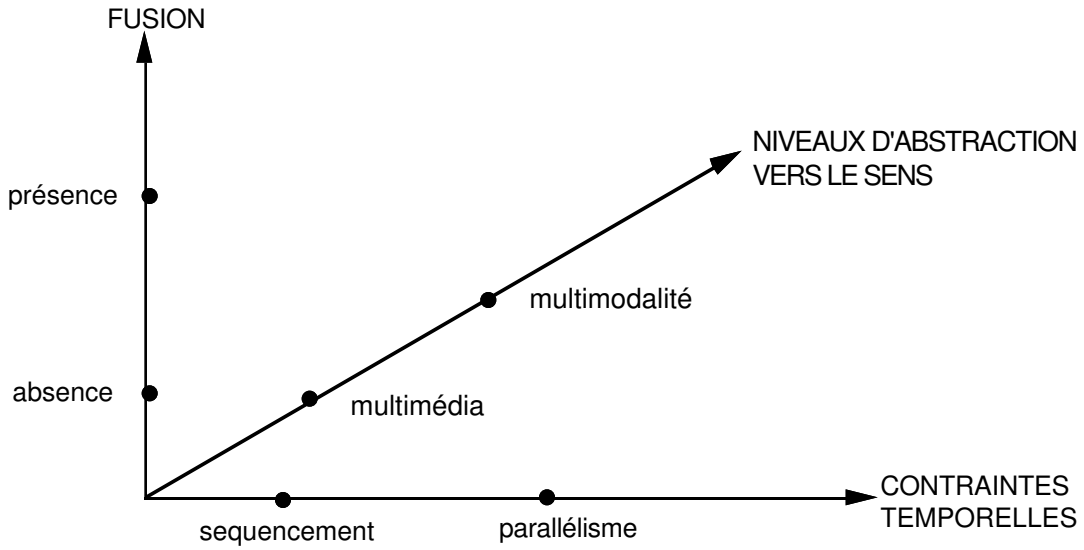

Figure 3 : Un espace de classification des systèmes multimédia et multimodaux.

Le pouvoir d'abstraction traduit la capacité interne du système à découvrir le sens des expressions d'entrée ou bien, dans le sens inverse, à produire des expressions à partir d'une représentation du sens. Le système est donc capable de gérer plusieurs niveaux d'abstractiuon allant du symbolique au physique. Les systèmes multimédias se limitent aux niveaux d'abstraction les plus bas : bien que

les images soient modifées par des algorithmes de compression, le pixel reste le concept de base. Au plus, cette image est-elle encapsulée dans une unité d'information, tel un nœud dans un réseau hypermédia. Dans ce réseau, le sens d'un texte parlé est greffé manuellement au moyen d'outils spécialisés. A l'inverse, les systèmes multimodaux visent à découvrir le sens automatiquement.

La fusion traduit la combinaison de différents types d'informations. Par exemple, l'incrustation d'images de synthèse dans un film vidéo est une fusion. L'expression "mets ça là" accompagnée des gestes déictiques indiquant l'objet à déplacer et le lieu destinataire, requiert la fusion d'informations d'entrée émises sur des canaux moteurs distincts.

Les contraintes temporelles ont trait à la séquentialité ou au parallélisme dans la production ou la réception des informations. Elles peuvent se situer à différents niveaux de granularité : canal de communication, entre unités d'information, au sein d'une unité d'information. Par exemple, le parallélisme au niveau du canal de communication offre la possibilité d'écouter de la musique tout en parlant et en saisissant du texte. Les contraintes temporelles entre unités d'information expriment la synchronisation par exemple entre des images et du texte : lorsque l'utilisateur déroule le texte, la séquence d'images associée est affichée. Au sein d'une unité d'information, le temps exprime les relations entre les entités de cette unité, par exemple les relations temporelles entre les objets d'un dessin animé.

En résumé, un système multimédia et un système multimodal sollicitent l'un et l'autre plusieurs canaux de communication humains. Tous deux, peuvent effectuer de la fusion d'informations et tous deux doivent résoudre les problèmes de synchronisation à divers niveaux de granularité. Alors qu'un système multimodal vise à découvrir le sens des informations automatiquement, un système multimédia requiert une greffe manuelle du sens. On observe aussi dans les applications actuelles que les informations multimédias sont l'objet de la tâche alors que les énoncés multimodaux servent à contrôler le déroulement de la tâche. Il est raisonnable de penser que cette distinction s'estompera avec les progrès de la communication multimodale. Toutefois, il faut noter que nous sommes loin de maîtriser ce nouveau type d'interaction aussi bien sur le plan cognitif que technique. A ce titre, il constitue un champ d'étude prometteur.

Références

[Bass 91]
L. BASS, J. COUTAZ: Developing Software for the User Interface; Addison Wesley, 1991.

[Coutaz 90]
J. COUTAZ : Interfaces Homme-Ordinateurs, Conception et réalisation; Dunod, 1990.

[Coutaz 92]
J. COUTAZ, “Multimedia and Multimodal User Interfaces:A Software Engineering Perspective”, proceedings of the *St Petersburg International Workshop on Human Computer Interaction’92*, St Petersburg, August, 1992.

[Krueger 90]
M. W. KRUEGER: Artificial Reality II; Addison-Wesley Publ., 1990.

[OSF 89]
OSF : OSF/Motif, Programmer's Reference Manual, Revision 1.0; Open Software Foundation, Eleven Cambridge Center, Cambridge, MA 02142, 1989.

[Webster 89]
B.F. WEBSTER : The NeXT Book ; Addison Wesley publisher, 1989.