

# Interfaces Procédurales

Guy A. Boy

Institut Européen de l'Ingénierie et des Sciences Cognitives (EURISCO)  
4, Avenue Edouard Belin, 31400 Toulouse, France  
+33.5.62.17.38.38 - boy@onecert.fr

## RESUME

Une interface procédurale permet aux utilisateurs de se concentrer sur la tâche qu'ils sont en train d'accomplir et d'éviter la surcharge inutile causée par l'utilisation des interfaces bureautiques (dont les procédures implicites relèvent précisément de tâches bureautiques). Elle se reconfigure automatiquement en fonction du contexte d'utilisation, prenant en compte trois types de fonctions cognitives : anticipation, interaction et récupération. Une interface procédurale intègre plusieurs concepts : simplicité, redondance, stabilité cognitive et l'appui cognitif sous-jacent. Un exemple d'application est donné dans le domaine des réunions assistées par ordinateur.

## ABSTRACT

A procedural interface enables users to concentrate on the task they are performing and to avoid any inappropriate overload coming from the use of desktop interfaces (where procedures are implicitly office-oriented). It automatically reconfigures itself depending its context of use, taking into account three types of cognitive functions: anticipation, interaction and recovery. A procedural interface involves several concepts such as simplicity, redundancy, cognitive stability and cognitive support. An example is provided in the domain of computer-supported meetings.

## Mots-clefs

Interface procédurale, redondance, stabilité cognitive, automatisation centrée sur l'homme, anticipation, interaction, récupération d'erreurs, fonctions cognitives.

## INTRODUCTION

Les interfaces bureautiques (*desktop*) ont commencé à être développées en 1981 avec la station de travail Xerox Star (Smith, 1982). En 1984, le Macintosh d'Apple a révélé au grand public ce type d'interaction (Apple, 1987). Cette évolution a été un grand tournant dans l'utilisation des ordinateurs et l'industrie informatique.

Cependant, dans le contrôle de processus industriels, l'aéronautique, la médecine et les autres domaines dans lesquels les utilisateurs ont besoin de manipuler des systèmes dynamiques, complexes et souvent à risques, la métaphore du bureau est très limitée et inadéquate lorsque les utilisateurs doivent exécuter des tâches complexes, en temps réel et à forte demande attentionnelle. Les gens n'ont pas le temps de chercher les commandes cachées d'un menu lorsqu'ils doivent répondre en urgence. Ils devraient pouvoir manipuler le bon objet d'interface à la bonne pace et au bon moment. La machine doit être automatisée en accord avec ce principe. Cette exigence est reconnue depuis longtemps dans la conception de systèmes à risques. Ces systèmes et environnements ont motivé le développement de l'automatique en tant que discipline. Automatiser consiste à transférer une fonction précédemment accomplie par un être humain à une machine. J'ai déjà proposé une méthode d'allocation de fonctions cognitives parmi des agents humains et machines (1998ab). Aujourd'hui, avec le développement des ustensiles électroniques, de nouveaux besoins émergent. Pour que l'utilisation de ces ustensiles soit efficace et la plus naturelle possible, l'utilisateur ne doit pas percevoir les couches informatiques qu'ils incluent; l'ordinateur doit être invisible (Norman, 1998). À la place, la tâche doit être suggérée par l'interface (et donc visible à tout instant). C'est précisément la raison d'être des interfaces procédurales.

Cet article introduit les principes de base de la conception d'une interface procédurale : simplicité, redondance, observabilité et commandabilité, et stabilité cognitive. Elle sert de support à l'exécution de tâche et met en jeu trois types de fonctions cognitives : l'anticipation, l'interaction et la récupération. Un langage de spécification est proposé pour le développement de ces interfaces. Même si plusieurs cas d'interfaces procédurales ont été développés dans le domaine aéronautique, je présente ici l'interface procédurale d'un environnement pour des réunions assistées par ordinateur.

## PRINCIPES DE BASE

En aéronautique, les pilotes doivent suivre des procédures opérationnelles. Nous avons déjà établi des catégories de facteurs reliant le suivi de procédures à la conscience de la situation (Boy & de Brito, 2000). Les concepts présentés dans le présent article sont dérivés de ces catégories pour être utilisés comme support à la conception simultanée

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
IHM 2002, November 26-29, 2002, Poitiers, FRANCE.  
Copyright 2002 ACM 1-58113-615-3/02/0011...\$5.00.

d'une interface et des procédures qui lui sont associées, et, en poussant plus loin, en intégrant les procédures au sein même de l'interface.

### **Simplicité et affordances perçues**

L'évolution progressive conduisant à remplacer les ordinateurs personnels par des ustensiles électroniques dédiés résulte de la compilation de tâches spécifiques. Par exemple, les tâches de coopération qui ont émergé d'un souci d'amélioration du travail ont conduit à rationaliser des procédures spécifiques. Les gens suivent implicitement ou explicitement de telles procédures. Avec le temps, on observe une co-adaptation des outils de travail et de ceux qui les utilisent. Des routines s'installent. Progressivement, les procédures peuvent être intégrées au sein des interfaces, des interfaces procédurales peuvent donc être développées. Plus spécifiquement, une réunion ne peut pas conduire à de bons résultats sans de telles procédures, par exemple attendre que celui qui parle ait fini d'exprimer ses points de vue, poursuivre en exprimant un autre point de vue, répondre à une question, et ainsi de suite. Toutes ces tâches sont simples en elles-mêmes. Un environnement pour des réunions assistées par ordinateur ne doit pas introduire de complications inutiles, mais suggérer ces tâches simples en contexte chaque fois que possible. Nielsen a déjà suggéré la simplicité comme un facteur clef pour la création d'interfaces Internet utilisables (Nielsen, 1999). Une interface procédurale ne sert qu'à une classe limitée de tâches spécifiques, en aucun cas à une grande variété de tâches. Pour cette raison, elle comporte un certain nombre d'objets d'interface qui permettent à l'utilisateur d'exécuter les tâches du domaine concerné. Ces objets doivent être visibles et saillants afin de suggérer les bonnes actions. Bien sûr, si la tâche est complexe, l'utilisation d'une interface procédurale peut demander un apprentissage. Gibson a fait l'hypothèse que la perception humaine est directe et n'est pas médiatisée par un processus de traitement de l'information (Gibson, 1977). Il a défini les *affordances* comme des relations innées entre les gens et les objets de leur environnement. Les *affordances* perçues (Norman, 1999) sont acquises, c'est-à-dire qu'elles sont des relations construites. Nous parlerons de *capacités suggestives d'action* (Boy, 1998b). Dans cet article, les *affordances* sont des réponses automatiques construites par un apprentissage. Elles sont liées aux concepts d'observabilité et de commandabilité qui vont être décrit ci-après.

### **Observabilité et commandabilité**

Lorsqu'un être humain contrôle un système, deux questions se posent :

- Le système est-il observable, c'est-à-dire les sorties disponibles sont-elles nécessaires et suffisantes pour comprendre ce que le système est en train de faire ?
- Le système est-il commandable, c'est-à-dire les entrées disponibles sont-elles nécessaires et suffisantes pour agir correctement sur l'état global du système ?

L'utilisateur développe habituellement un modèle cognitif pour contrôler le système dont il a la charge, associant les états observables aux états commandables (Norman, 1986, Rasmussen, 1986). Il existe toujours un compromis entre contrôler le système en utilisant un grand ensemble d'états observables indépendants et un petit ensemble d'états observables intégrés. "... plus le nombre de degrés de liberté d'un système est grand, plus il est difficile de le manipuler. Même si le simple décompte des degrés de liberté simplifie considérablement le problème, c'est la manière dont les degrés de liberté interagissent entre eux qui détermine la difficulté de contrôler le système. Par exemple, si les  $n$  degrés de liberté d'un système sont indépendants les uns des autres, alors le système contrôlant doit seulement mettre en œuvre un algorithme adapté au contrôle d'un seul degré de liberté ; l'algorithme peut être dupliqué  $n$  fois pour contrôler l'ensemble. Au contraire, si les degrés de liberté ne sont pas indépendants (c'est-à-dire que lorsque l'affectation d'une valeur à un degré de liberté dépend des valeurs des autres degrés de liberté), alors une équipe de contrôleurs indépendants n'est plus pertinente et des algorithmes de contrôle plus complexes doivent être envisagés." (Jordan & Rosenbaum, 1989).

### **Redondance**

Le mathématicien Claude E. Shannon, travaillant pour la société de télécommunication Bell, a publié en 1949 une théorie mathématique de la communication. Cette théorie permet de mesurer l'information, et en particulier la redondance de l'information. La question était de trouver l'information minimale qui permet de comprendre correctement les messages transférés. En effet, une information est redondante lorsque le récepteur d'un message peut comprendre son sens sans attendre le reste du message. Shannon a relié l'intérêt de l'information à sa probabilité, c'est-à-dire plus l'information est improbable, plus elle est intéressante. Contrairement au but de Shannon qui consiste à éliminer l'information redondante pour accroître la productivité de la communication entre l'Europe et les Etats-unis par l'intermédiaire d'un câble transatlantique, les systèmes à risques nécessitent des informations redondantes permettant de mieux les contrôler. Les signes apparemment superflus dans la compréhension d'un message sont dits redondants. La phrase suivante est redondante (trois fois) par rapport au pluriel : "Les dauphins sont de bons nageurs". Si par erreur vous omettez un "s" alors le lecteur comprendra toujours que la phrase est au pluriel. En répétant un mot, nous augmentons l'effet de ce mot comme : "C'est très très bon !" Si vous effacez l'un des "très" alors la phrase aura toujours le même sens. De la même façon, la navette spatiale a cinq ordinateurs, trois systèmes inertiels et deux systèmes indépendants qui calculent sa position par rapport aux étoiles. Si l'un des ordinateurs ou systèmes est défaillant, les autres prendront le relais pour maintenir la navette fonctionnelle. Les êtres humains ont leurs propres redondances pour survivre. Plus communément, lorsque

vous utilisez un ordinateur, vous avez besoin de redondance. En d'autres termes, vous avez besoin de sauvegarde de vos fichiers pour travailler en toute sécurité.

L'interface d'un système est caractérisée par un ensemble de  $n$  états observables ou sorties  $\{S_1, S_2, \dots S_n\}$  et un ensemble de  $m$  états commandables ou entrées  $\{E_1, E_2, \dots E_m\}$ . L'interface est redondante s'il y a  $p$  sorties ( $p < n$ ) et  $q$  entrées ( $q < m$ ) nécessaires et suffisantes pour utiliser correctement le système. Les  $(n-q)$  sorties et  $(m-q)$  entrées restantes sont des états redondants de l'interface lorsqu'elles sont associées à des sous-systèmes indépendants du système global. Ces états redondants doivent être choisis de façon à assister l'utilisateur en situation normales, anormales et d'urgence. Dans les cockpits d'avion, par exemple, plusieurs instruments sont dupliqués, un pour le commandant de bord, un autre pour le copilote. De plus, des états observables présentés sur des instruments digitaux sont aussi présentés sur des instruments traditionnels redondants. Contrôler un système état-par-état en utilisant des informations redondantes appropriées est très différent de déléguer l'activité de contrôle à un automate. De nouvelles formes de redondance émergent de l'utilisation des systèmes hautement automatisés. L'observabilité et la commandabilité des systèmes est habituellement centrée sur le *Quoi* des états du système à contrôler. La supervision des systèmes hautement automatisés nécessite des informations redondantes sur le "pourquoi", le "comment", le "avec quoi" et le "quand" afin d'augmenter l'*insight*, la confiance et la fiabilité : *Pourquoi* le système fait ce qu'il fait ? *Comment* agir sur le système pour le mettre dans un certain état en utilisant les commandes disponibles ? *Avec quelles autres présentations* et commandes les entrées/sorties courantes devraient être associées ? *Quand* une information devrait être envoyée ou acquise ?

### **Stabilité cognitive**

La stabilité cognitive est analysée en utilisant la métaphore de la stabilité en physique. La stabilité peut être statique ou dynamique. La stabilité statique est liée aux degrés de liberté ; en d'autres termes un objet dans un monde tridimensionnel est habituellement défini par trois degrés de liberté. Une chaise est stable lorsqu'elle a (au moins) trois pieds. Les êtres humains sont stables sur deux jambes, mais c'est à cause de la stabilité dynamique parce qu'ils ont appris à compenser, souvent inconsciemment, leur instabilité. Lorsqu'un objet est perturbé par un événement externe, il y a habituellement deux cas : l'objet revient à sa position initiale, nous disons que l'objet est dans un état stable ; et l'objet diverge de sa position initiale, nous disons que l'objet est (ou était) dans un état instable. Beaucoup de chercheurs ont étudié les erreurs humaines de manière extensive pendant les deux dernières décennies et en ont déduit plusieurs catégories (Norman, 1981; Reason, 1990; Hollnagel, 1993). Lorsqu'un utilisateur agit de façon erronée, deux cas se présentent : l'utilisateur récupère de

son action erronée, nous disons que l'utilisateur est dans un état stable ; et l'utilisateur ne récupère pas de son action erronée, nous disons que l'utilisateur est (ou était) dans un état instable.

Il existe des actions humaines erronées qui peuvent être tolérées, et d'autres qui doivent être bloquées. Les systèmes de tolérance et de résistance à l'erreur sont habituellement des redondances utiles. La tolérance à l'erreur est toujours associée à la récupération de l'erreur. Il y a des erreurs bonnes à faire parce qu'elles encouragent la prise de conscience de la situation et la récupération. Cependant, la récupération est souvent difficile, et souvent impossible, lorsque les ressources nécessaires ne sont pas disponibles. Je propose le concept de réversibilité de l'action chaque fois qu'un utilisateur peut revenir en arrière à partir d'une action erronée, et agir correctement. La fonction UNDO (défaire) disponible sur la plupart des applications logicielles aujourd'hui offre une redondance aux utilisateurs qui détectent les fautes d'orthographe et décider de les corriger. Alors, faire des fautes d'orthographe est toléré, et une ressource de récupération est disponible. La résistance à l'erreur est, ou devrait être, associée au risque. Les ressources de résistance à l'erreur sont utiles dans les systèmes à risques lorsque des hauts risques sont possibles. Ils ne peuvent pas être pertinents dans les environnements à bas risques parce qu'ils perturbent habituellement l'exécution de la tâche. Par exemple, les outils de traitement de texte qui offrent la correction automatique d'orthographe peuvent perturber la tâche principale de génération d'idées. Un apprentissage et une formation inadéquats, une faible vigilance, une fatigue et une grande charge de travail constituent les influences principales adverses à la stabilité cognitive. La stabilité cognitive est favorisée par la simplicité, la redondance ainsi qu'une observabilité et des commandabilité appropriées de l'interface utilisateur. Les utilisateurs tendent à utiliser des sources redondantes d'information qu'elles soient personnellement construites ou délibérément données pour maintenir une stabilité cognitive raisonnable. De telles sources redondantes d'information sont appelées fonctions cognitives stabilisantes. "Que se passera-t-il si je fais cela?" Toute ressource redondante qui contribue à répondre à une telle question de l'utilisateur est susceptible de contribuer à la stabilité cognitive d'un système donné.

### **ASSISTANCE A L'UTILISATEUR**

#### **Anticipation, interaction et récupération**

Tout utilisateur emploie au moins les trois fonctions cognitives suivantes pour maintenir sa propre stabilité cognitive dans l'exécution d'une tâche donnée :

1. *L'anticipation* est une fonction qui met en jeu la prise en compte d'événements en avance et la préparation d'une "réaction" en focalisant la mémoire de travail sur un ensemble approprié de formes situationnelles (ensemble de conditions sur la situation corrélées ou non entre elles) (Boy, 1998b). Ces formes sont

acquises pas à pas par la formation et la pratique, articulées et accumulées dans la mémoire à long terme.

2. *L'interaction* est une fonction qui met en jeu des réactions directes à des événements autant que des actions articulées à partir d'une intention. L'interaction est typiquement opportuniste dans le sens où un être humain peut commencer à développer un ensemble d'actions intentionnelles, être perturbé par un événement non prévu, réagir à celui-ci, et éventuellement oublier les actions initialement planifiées. On peut donc dire que toute interaction est une réaction à un stimulus externe ou à une intention de l'utilisateur.
3. *La récupération* est une fonction qui met en jeu des ensembles d'actions consciemment construites et articulées (plans d'actions) suite au diagnostic d'une situation. Habituellement une telle situation est provoquée par un événement qui n'était pas attendu dans l'ensemble initial des actions à exécuter. La récupération est basée sur la reconstruction causale de sens afin de construire un plan d'actions.

Pour chacune de ces fonctions cognitives, les utilisateurs développent des ensembles appropriés d'actions pour résoudre un problème. La nature de l'expertise se traduit par la façon dont les gens anticipent, interagissent ou récupèrent. Les gens anticipent en faisant des prédictions du *futur* afin de sélectionner les bonnes formes situationnelles d'actions prêtes à utiliser le moment venu. Lorsque les gens interagissent, la nature de l'interface est cruciale puisque les gens répondent à des stimuli qui leur sont *immédiatement* perceptibles et font sens par rapport à leurs intentions et leurs plans. Lorsque les gens récupèrent d'une situation non anticipée ou anormale, le diagnostic est crucial puisque les plans d'actions subséquents sont basés sur une conscience appropriée de la situation et des événements *passés* significatifs.

#### **Appui cognitif**

Je transpose ici la notion d'appui en physique dans le domaine de l'ingénierie cognitive. Les gens s'appuient sur des artefacts cognitifs comme ils s'appuient sur des objets physiques. En particulier, les gens améliorent leurs fonctions cognitives d'anticipation, d'interaction et de récupération par l'utilisation de diverses catégories d'appuis cognitifs. Ces catégories peuvent être liées à l'outil physique utilisé, les capacités de l'utilisateur, la tâche à accomplir, et leur environnement organisationnel. Par exemple, la redondance est un appui cognitif. L'utilisation simultanée de couleur, de formes et de groupements d'états reliés est un exemple classique de redondance de l'interface utilisateur. Les gens perçoivent à la fois les contrastes et les similarités. L'information improbable devrait être soulignée afin d'anticiper les surprises possibles. Le retour redondant d'information sur les actions de l'utilisateur est susceptible d'améliorer l'interaction. Une assistance

additionnelle à l'action devrait améliorer la récupération. Dans tous les cas, l'observabilité et la commandabilité redondantes, telles que définies plus haut, sont susceptibles d'améliorer la stabilité cognitive. Le développement de formes situationnelles et d'habitudes de traitement comme la vérification croisée systématique est susceptible d'améliorer l'anticipation, l'interaction ou la récupération. La formation à la récupération d'erreur est susceptible d'améliorer la stabilité cognitive. Les gens qui agissent (disons les acteurs) cherchent habituellement des appuis cognitifs que ce soit consciemment ou inconsciemment. Lorsqu'un tel appui n'est pas disponible, les gens le construisent soit comme une bonne pratique soit comme des outils externes servant de support à leurs activités. Les gens qui utilisent un environnement pour des réunions assistées par ordinateur ont besoin de règles (souvent appelées procédures) pour communiquer efficacement avec les autres. Plus généralement, toute coopération efficace nécessite des formes d'interaction procéduralisées. Ces formes d'interaction procéduralisées incluent des informations redondantes servant d'appui cognitif à leurs utilisateurs. Les tâches de communication peuvent être représentées comme des algorithmes d'actions. Puisque de tels algorithmes sont répartis parmi les gens et la technologie. Les gens ont besoin d'utiliser des cartes cognitives servant de support à leurs activités. Les fonctions cognitives d'anticipation, d'interaction et de récupération peuvent être améliorées lorsque ces cartes cognitives sont disponibles au sein de l'interface utilisateur.

#### **UN LANGAGE DE SPECIFICATION : LES BLOCS**

En pratique, l'analyse des fonctions cognitives (Boy, 1998) est une méthodologie qui permet d'identifier la congruence d'un ensemble de fonctions cognitives ; les fonctions cognitives redondantes en particulier. Ceci conduit au développement et à l'utilisation d'un formalisme de description appelé blocs d'interaction permettant de capturer des formes émergentes d'interactions ou procédures.

#### **Blocs d'interaction = procédures**

Un bloc d'interaction (i-Bloc) est défini par les attributs suivants (Figure 1) : un algorithme d'actions ; une forme situationnelle qui inclue des *pré-conditions de déclenchement* et une *forme contextuelle* ; des post-conditions qui incluent un *but* (post-conditions normales) et des (post)conditions anormales. La normalité est (faiblement) défini comme ce qui est préféré de faire dans un contexte donné, et est associée à ce qui est fréquemment fait.

Par exemple, une réunion assistée par ordinateur dans le contexte d'une session GEM (Group Elicitation Method) (Boy, 1996) est représentée par un contexte de i-Blocs (Figure 2). Soit l'ensemble des i-Blocs {"Préparation", "Brainwriting", "Debriefing", "Définition des participants"} qui ont les mêmes conditions contextuelles (formes contextuelles). "Préparation" comprend trois i-

Blocs non présentés dans la Figure 2 : “Définition du projet”, “Sélection des participants” et “Briefing”. Les flèches représentent des liens et sont identifiées par des nombres, de 1 à 6 dans la Figure 2.

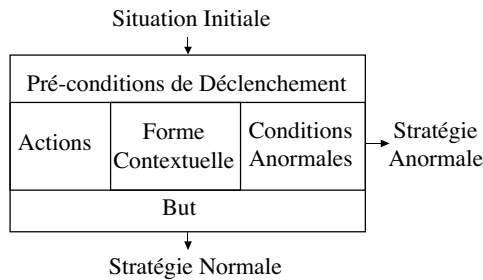


Figure 1. Représentation d'un bloc d'interaction.

Par exemple, le lien 4 relie “Préparation” à “Définition des participants” lorsqu’une condition anormale donnée du i-Bloc “Préparation” est satisfaite, par exemple “pas assez de participants pour que la réunion ait lieu”. Le i-Bloc “Préparation” est appelé la “racine” du contexte de i-Blocs “session GEM”. Un contexte de i-Blocs est un i-Bloc lui-même. Dans notre exemple, les pré-conditions de déclenchement du contexte de i-Blocs sont les pré-conditions de déclenchement du i-Bloc “Préparation”, ses conditions anormales sont les i-Blocs “Définition des participants”, et son but est “Debriefing”. Cette propriété est très utile dans l’analyse des fonctions cognitives : pour étendre un i-Bloc en un contexte de sous-i-Blocs ; pour grouper plusieurs i-Blocs déjà développés en un contexte de i-Blocs.

### Blocs d'interaction et interfaces procédurales

En situation normale, les i-Blocs sont organisés et traités en séquence, par exemple “Préparation” puis “Brainwriting” ... puis “Debriefing” (Figure 2). Le traitement résultant est *linéaire*. En situation anormale, cette séquence est interrompue par la création d’un branchement vers d’autres i-Blocs (conduisant à des actions de récupération). Dans notre exemple GEM, supposons que, lorsque l’on exécute les actions du i-Bloc “Préparation”, le nombre de participants n’est pas suffisant (condition anormale faible 4). L’organisateur de la session GEM doit alors utiliser le i-Bloc “Définition des participants” qui permet de trouver plus de participants sauf si le nombre de participants reste trop faible après un temps donné (condition anormale forte 6). Dans ce cas, l’organisateur de la session GEM doit changer de contexte et arrêter la session GEM. Si le nombre de participants est acceptable, alors la procédure “Définition des participants” aboutit et l’organisateur peut terminer la procédure “Préparation” (lien 5). Le traitement résultant est *non-linéaire*.

Il convient de noter qu’une structure de i-Blocs peut être représentée sous une forme hypertextuelle. Les buts conduisent à des liens linéaires entre i-Blocs. Les

conditions anormales conduisent à des liens non-linéaires entre i-Blocs. Chaque i-Bloc est lié à une représentation graphique externe ayant un sens pour les utilisateurs. Cette propriété hypertextuelle peut être utilisée pour concevoir et développer une interface procédurale.

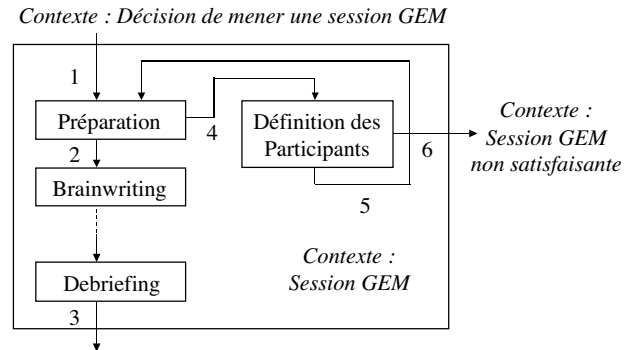


Figure 2. Représentation i-Bloc d'une session GEM.

## APPLICATION AUX REUNIONS ELECTRONIQUES

### Codage couleur

En aéronautique, comme dans beaucoup d’autres domaines industriels, le codage couleur est systématiquement utilisé pour maximiser les *affordances* perçues en suggérant des actions appropriées et communiquer la bonne information au bon moment sous le bon format. Le codage couleur est une information redondante qui améliore l’observabilité et la commandabilité. Dans cet article, focalisons nous sur un exemple simple de codage couleur et une interface dépendant du contexte d’un environnement pour réunions assistées par ordinateur : le système TEAM™ qui instrumentalise les sessions GEM. Bien sûr, les utilisateurs ont besoin d’apprendre le codage couleur pour pouvoir anticiper, interagir et récupérer de leurs propres erreurs ou des pannes du système. Plus le codage couleur est universel, plus les gens sont susceptibles de l’utiliser efficacement et en toute sécurité. C’est une difficulté parce que la signification des couleurs dépend fortement de la culture. Le rouge n’a pas la même signification en France et en Chine par exemple. Cependant, il y a diverses catégories de significations des couleurs liées à la pratique. Une fois que les gens les ont apprises, ils les utilisent correctement. Le nombre de couleurs doit généralement être suffisamment faible pour les gens se souviennent de leurs significations. Il est communément admis que les significations de  $7 \pm 2$  couleurs sont facilement mémorisées. L’interface utilisateur du système TEAM™ comprend des boutons et des champs codés avec des couleurs. Le *vert* signifie que l’utilisateur est invité à fournir une information à l’ordinateur, par exemple sélectionner ou entrer une valeur. *Bleu foncé* indique une sortie à l’utilisateur. *Jaune* indique que le processus est en cours. *Cyan* indique que l’information correspondante est validée. *Gris* signifie inactif, c’est-à-dire lorsque l’utilisateur agit sur une zone grise rien ne se passe.

### Organisation de l'interface dirigée par la tâche

Dans la plupart des ustensiles électroniques comme les applications pour les réunions assistées par ordinateur, il y a très peu de commandes, de présentations et de champs d'entrée dont les utilisateurs ont besoin. Les présentations et les commandes peuvent immédiatement suggérer les bonnes actions à leurs utilisateurs. Dans la mesure où les utilisateurs n'ont pas à se focaliser sur des questions d'interaction telles que chercher la bonne commande cachée quelque part dans un menu pop-up, ils peuvent porter plus d'attention à la tâche qu'il doivent accomplir. Par exemple, l'environnement de réunion TEAM™ permet à plusieurs utilisateurs d'exécuter des tâches telles que générer des points de vue, dériver des concepts et atteindre des consensus. Comme dans tout environnement de réunion, il y a des procédures à suivre pour garantir de bons résultats. Dans TEAM™, ces procédures sont intégrées dans l'interface utilisateur selon une structure d'i-Blocs. À tout instant, les participants à une réunion TEAM™ savent combien de temps il leur reste pour exprimer leurs points de vue par exemple. Ils sont avertis plusieurs fois avant la fin. Autre exemple, une fenêtre de discussion (*chat*) est automatiquement présentée lorsque le facilitateur envoie un message de priorité. Cette fonction a été pensée comme une interruption au sein du déroulement normal de la réunion et pour maintenir une bonne stabilité cognitive.

### Exemples d'utilisation

Dans l'environnement TEAM™, le système dédié au facilitateur de la réunion électronique inclut la procédure suivante :

- définir ou choisir un projet auquel cette réunion appartient ;
- choisir et interagir avec un ensemble de participants (bouton *Project*) ;
- définir une question (bouton *Briefing*) ;
- accomplir un brainstorming écrit (bouton *Brainwriting*) ;
- accomplir un regroupement des points de vue en concepts (bouton *Concept*) ;
- établir des priorités (bouton *Consensus*) ;
- accomplir un *debriefing* (bouton *Debriefing*) ;
- déduire une décision et des actions (bouton *Decision*).

Dans la figure 3, le bouton "Project" est *jaune*, c'est-à-dire le projet courant est en cours et l'utilisateur ne peut pas l'interrompre. Le bouton "Briefing" est *bleu foncé*, c'est-à-dire le briefing est en cours, mais l'utilisateur peut en sortir s'il le désire. Les autres items (boutons) de procédure globale sont *gris*, et donc inactifs. Le bouton "Staff Validated" est *cyan*, c'est-à-dire l'ensemble des participants choisis a été validé. Le bouton "Request" est *vert*, c'est-à-dire l'accès direct aux participants via email est actif et peut être sélectionné. Il en est de même pour le bouton "Help". Les champs "Question" et "Contents" sont *jaunes*, c'est-à-dire l'utilisateur peut y entrer du texte.

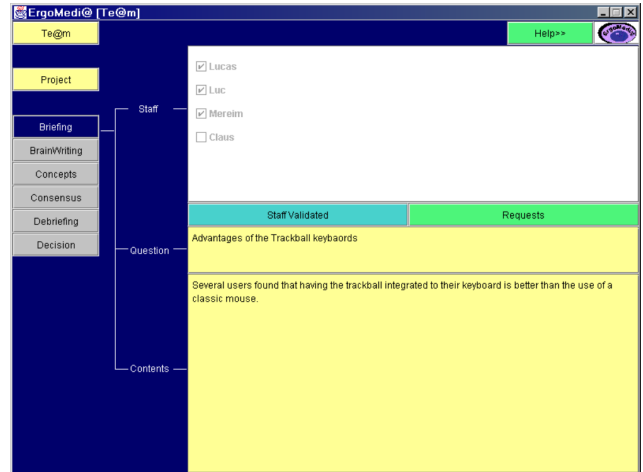


Figure 3. Ecran "Briefing" de l'interface TEAM™.

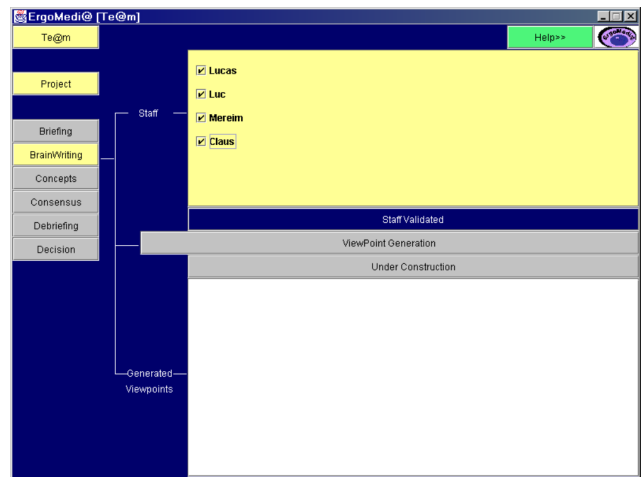


Figure 4. Ecran 1 "Brainwriting" de l'interface TEAM™.

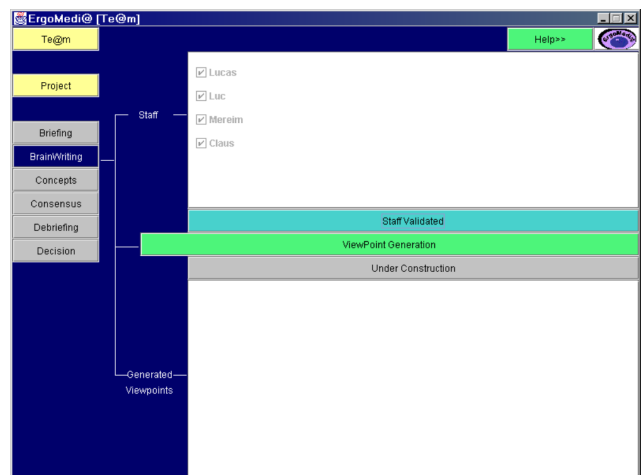


Figure 5. Ecran 2 "Brainwriting" de l'interface TEAM™.

Dans la figure 4, le briefing est terminé et la phase de brainwriting est en cours sans sortie possible parce que le processus de validation des participants est en cours. Seulement le bouton “Staff Validated” est une sortie (*Bleu foncé*). Dans la figure 5, le bouton “Staff Validated” devient *cyan* parce que l'utilisateur a validé la liste choisie des participants, le bouton “Viewpoint Generation” devient *vert* et le bouton “Brainwriting” devient *bleu foncé*. Toutes ces transitions sont gérées par une structure de i-Blocs.

### Résultats sur l'utilisabilité

*Simplicité.* Les testeurs de TEAM™ ont trouvé que l'utilisation du codage couleur et des présentations et commandes dirigées par la tâche était très efficace, et facile à apprendre. Les erreurs observées étaient liées aux problèmes typiques de beta-test ainsi qu'à des erreurs d'étourderie. Ces problèmes ont été corrigés, aucune autre question négative n'a été identifiée par rapport aux erreurs humaines puisque les utilisateurs n'avaient qu'à suivre les procédures mises en œuvre (voir les résultats sur la stabilité cognitive). Il convient de noter que le faible nombre d'alternatives d'actions possibles combiné au codage couleur favorise grandement l'anticipation, l'interaction et la récupération.

*Redondance.* Plusieurs sortes de redondances simples ont été mises en œuvre au sein du système. Tout d'abord, le codage couleur et l'ordre d'apparition des commandes et des présentations présentent typiquement des informations redondantes. Ensuite, tous les boutons sont équipés d'explications explicites (tool tips) fournissant le “quoi”. Le “comment le système fait ce qu'il fait” est donné par le bouton contextuel Help. Ce sont des fonctions habituelles des interfaces bureautiques courantes. Finalement, les redondances “comment”, “avec quoi” et “quand” sont intégrées dans l'interface procédurale par l'intermédiaire de boutons et de présentations selon une logique géographique et chronologique suivant le déroulement de la réunion GEM. Cette sorte de redondance simplifie considérablement l'interaction homme-machine au bénéfice d'une plus grande concentration sur les tâches typiques dévolues à une réunion.

*Stabilité cognitive.* Ce serait trop long de décrire toutes les questions d'interaction posées par les participants à la réunion. Cependant, il convient de rapporter qu'ils ont tous apprécié la simplicité de l'outil pour récupérer des erreurs mineures telles que valider un point de vue qui n'était pas totalement clair et bien écrit et en le réécrivant immédiatement ou plus tard. Comme un principe général, le concept de stabilité cognitive a été utilisé pour tester et améliorer l'interface procédurale afin de permettre aux utilisateurs de récupérer des erreurs mineures et de bloquer des erreurs plus sérieuses.

*Appui cognitif.* Puisque le développement d'un tel outil est centré sur l'utilisateur, à chaque étape la tâche et l'interface ont été développées simultanément, c'est-à-dire les fonctions derrière chaque bouton et présentation ont été créées et modifiées progressivement après chaque

évaluation. L'appui cognitif pour l'utilisation d'un tel environnement de réunion a émergé du processus de conception participative. Par exemple, nous avons découvert qu'il existait seulement six couleurs pertinentes pour ce type d'interface procédurale. De plus, les noms des boutons peuvent changer dans certains cas pour suggérer la bonne action dans le contexte courant. Nous avons retrouvé comme une bonne pratique que l'interprétation chronologique d'une séquence de tâche est soit de haut en bas ou de la gauche vers la droite. Ce sont des exemples très simples de cette sorte d'appui cognitif que nous avons étudiés. Ce travail fondamental est en cours et des études plus approfondies sont menées sur des systèmes aérospatiaux. Les résultats correspondants seront publiés plus tard.

### AUTRES TRAVAUX

Le développement d'une interface procédurale conduit au même type d'avantages que ceux qui sont donnés par la boîte à outils contextuelle développée par Salber et al. (1999), comme la génération de blocs réutilisable pour la prise en compte du contexte. Les interfaces procédurales sont développées à partir d'une analyse à base de blocs d'interaction et testées pas à pas par rapport à l'ensemble des principes décrits plus haut. Des travaux similaires sont actuellement menés sur l'informatique sensible au contexte (*context-aware computing*) (voir l'édition spéciale du journal HCI éditée par Moran & Dourish, 2001). Diverses sortes d'outils informatiques et d'environnements de travail émergent dans des situations où le contexte est crucial. Si la notion de contexte est très récente dans la conception d'interface homme-machine (Moran, 1994), c'est une question clef pour les prochaines décennies dans les domaines où le temps, la complexité et la sécurité sont des facteurs clefs. Aujourd'hui, l'informatique interactive et les systèmes d'information en réseau ont conduit aux mêmes problèmes posés depuis longtemps dans des domaines comme l'aviation ou le contrôle de processus.

### CONCLUSION

Cet article a présenté les concepts de simplicité, redondance, observabilité et commandabilité qui servent de support à la stabilité cognitive des systèmes homme-machine. Les blocs d'interaction ont été présentés comme un langage de spécification pour la conception des interfaces procédurales. Celles-ci ont été appliquées avec succès à la conception d'un environnement pour des réunions assistées par ordinateur. Cette approche et méthode ont été utilisées avec succès à d'autres domaines comme l'aéronautique dans lesquels l'interaction doit être la plus naturelle et intuitive possible. (Boy & Ferro, 2000). Les relations homme-machine sont construites par un apprentissage progressif et l'appropriation d'une culture ainsi qu'une conception participative. Diverses sortes d'appuis cognitifs ont émergé pour améliorer ces relations. Les outils d'appui cognitif peuvent être intrinsèques sous la forme de pratiques redondantes, ou extrinsèques sous la

forme de d'outils externes. Les interfaces procédurales induisent des formes simples d'interaction prenant en compte la complexité des tâches procédurales inutiles à prendre en compte par les utilisateurs.

#### REMERCIEMENTS

Merci à IndySoft et FCL International qui ont donné un support à ce travail. Je remercie spécialement Meriem Chater, André Fort, Luc Haudot, Patrick Martinent, et Lucas Stéphane. René Amalberti et Andrew Gillett ont donné des conseils précieux pour améliorer la qualité de cet article.

#### REFERENCES

1. Apple. *Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley.
2. Boy, G.A. (1996). The Group Elicitation Method for Participatory Design and Usability Testing. Companion Proceedings of CHI'96, ACM Conference on Human Factors in Computing Systems, ACM Press, New York.
3. Boy, G.A. (1998a). Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems in *Proceedings of CHI'98* (May 1998), ACM Press, 265-272.
4. Boy, G.A. (1998b). *Cognitive Function Analysis*. Ablex. Distributed by Greenwood Publishing Group, Westport, CT.
5. Boy, G.A. (1998c). L'interaction homme-machine : une approche de l'ingénierie cognitive pour la conception centrée sur l'homme. *Sécurité et cognition*. Colloque du GIS Sciences de la Cognition, Ministère de la Recherche. HERMES Science, Paris.
6. Boy, G.A. & De Brito, G. (2000). Toward a Categorization of Factors related to Procedure Following and Situation Awareness. *Proceedings of the HCI-Aero 2000 Conference*, In Cooperation with ACM-SIGCHI. Cepadues, Toulouse, France.
7. Boy, G.A. & Ferro, D. (to appear in 2001). Using Cognitive Function Analysis to Prevent Controlled Flight Into Terrain. Chapter for the *Human Factors and Flight Deck Design Book*, Don Harris (Ed.), Ashgate, Aldershot, UK.
8. Gibson, J.J. (1977). The theory of affordances. In R.E. Shaw & J. Bradshaw (Eds.) *Perceiving, Acting and Knowing*. Lawrence Erlbaum Associates, Hillsdale, NJ.
9. Hollnagel, E. (1993). The phenotype of erroneous action. *International Journal of Man-Machine Studies*, 39, pp. 1-32.
10. Jordan, M.I. & Rosenbaum, D.A. (1989). Action. In *Foundations of Cognitive Science*, Michael I. Posner (Ed.). The MIT Press, Cambridge, MA.
11. Moran, T.P. (1994). Special issue on context in design. *HCI Journal*. Volume 9, pp. 1-149.
12. Moran, T.P. & Dourish P. (2001). Introduction to the special issue on Context-Aware Computing. *HCI Journal*. Volume 16.
13. Nielsen, J. (1999). *Designing Web Usability. The Practice of Simplicity*. New Riders Publishing, Indiana.
14. Norman, D.A. (1986). Cognitive Engineering. In *User-Centered System Design.*, D. Norman & S. Draper (Eds.), Hillsdale, NJ: Lawrence Erlbaum Associate, pp. 31-61.
15. Norman, D.A. (1998). *The invisible computer*. Cambridge, MA: MIT Press.
16. Norman, D.A. (1999). Affordances, conventions and design. *Interactions*, 6, 3, pp. 38-43.
17. Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction - An Approach to Cognitive Engineering*. North Holland Series in System Science and Engineering, A.P. Sage, Ed.
18. Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.
19. Salber, D., Dey, A.K. & Abowd, G.D. (1999). The Context Toolkit: Aiding the Development of Context-Enabled Applications. *Proceedings of CHI'99*, Pittsburgh, PA, USA, pp. 434-441.
20. Smith, D. (1982). Designing the Star User Interface. *Byte*, April, pp. 242-282.