

Les caractéristiques de l'informatique pédagogique

Ce chapitre décrit les principales caractéristiques de tous les logiciels, et notamment des logiciels éducatifs, pour en faire comprendre les propriétés communicationnelles et informationnelles. On retient huit caractéristiques principales : 1- l'interface, 2- l'information, 3- l'interactivité, 4- l'icône et le curseur, 5- la convivialité, 6- le contrôle, 7- la courbe d'apprentissage et 8- les médias. Ces caractéristiques sont tellement communes qu'il n'y a pas lieu de dire que telle caractéristique est présente dans tel logiciel, c'est une évidence. Cependant, selon le style ou la manière d'être, telle caractéristique devient une qualité remarquable dans tel logiciel. Comme logiciel éducatif, un document doit aussi appartenir à un modèle, comme on l'a vu dans un chapitre antérieur, et posséder les caractéristiques et qualités relatives à ce modèle.

1. L'interface avant tout

En informatique, l'information n'est pas présentée en surface comme dans des livres ou sur des tableaux, elle est plutôt présentée en interface, c'est-à-dire au moyen d'un appareil de représentation de l'information comme l'écran ou le haut-parleur. C'est sur l'écran de visualisation (terme français) appelé souvent moniteur (terme calqué de l'anglais mais accepté en français) que réside l'essentiel de l'interface. L'ordinateur est un appareil principalement visuel. Donc pour lire, naviguer ou écrire à l'ordinateur, il faut passer par l'interface qui constitue les sens de l'ordinateur, si on emploie une analogie humaine. Et sans jeu de mot, tout le sens de l'information passe totalement et uniquement par l'interface.

Les fonctions de l'interface

L'interface a plusieurs fonctions. Elle sert d'abord d'aide-mémoire en permettant à l'utilisateur de retrouver les éléments et fonctions disponibles sur un poste. Par exemple, elle permet de savoir quels sont les logiciels disponibles et lorsqu'un logiciel en particulier est choisi, elle permet de savoir quelles sont les fonctions que ce logiciel autorise. Elle permet aussi l'échange ou l'interactivité entre l'utilisateur et cette machine. Bref, elle est la vitrine unique de toute information et communication.

Les conventions

Pour rappeler ce qui est disponible dans un poste d'ordinateur, l'interface utilise un certain nombre de conventions que l'utilisateur doit reconnaître et savoir utiliser. On retrouve principalement les menus, les icônes souvent sous-titrées, les formulaires à remplir. Si l'utilisateur connaît le langage informatique ou langage de programmation, on retrouve alors dans ces cas l'écriture de la commande désirée. Enfin pour les habitués, on retrouve souvent un raccourci au clavier des commandes par ailleurs disponibles par l'intermédiaire d'un menu ou d'une icône. Lorsque les fonctions les plus usuelles sont regroupées et présentées dans l'espace-écran, on parle de barre d'outils.

L'espace sur l'écran

Les moniteurs ou écrans ont beau être toujours plus grands, il reste que l'espace d'écran utilisé pour gérer le logiciel et l'ordinateur n'est pas disponible pour représenter de l'information spécifique. Une métaphore peut aider à saisir la nuance entre l'information spécifique et l'information de gestion. L'information spécifique est la véritable nouvelle information qui en contexte de l'automobile pourrait être appelée l'information « pare-brise » ou l'information spécifique qui se déroule dans le paysage en opposition à l'information de gestion qui pourrait être appelée l'information « tableau de bord » qui rapporte l'information de la machine.

Une contestation de territoire surgit entre la gestion du logiciel et de l'ordinateur, d'une part, et l'information spécifique, d'autre part. Les menus se déroulent ou apparaissent au besoin, ce qui occupe peu d'espace. Par ailleurs, les icônes et leur regroupement sous forme de barres d'outils sont plus faciles à retrouver mais elles utilisent une plus grande surface, ce qui diminue d'autant l'espace disponible pour l'information. Enfin les formulaires à remplir occupent beaucoup d'espace sur l'écran mais ils permettent de structurer toutes les caractéristiques désirées pour accomplir une tâche. La proportion de surface occupée par les formulaires est tellement grande qu'il faut les faire disparaître lorsqu'ils ne servent pas car tout le territoire est envahi.

Interface générale, interface du logiciel

À l'arrivée sur un poste, un usager fidèle reconnaît facilement à quel système d'exploitation il a affaire, c'est la partie générale de l'interface. En éducation, on retrouve surtout le Macintosh et le PC sous Windows. En plus, chaque logiciel apporte son interface particulière. Même si les designers s'efforcent d'adopter des standards pour ne pas trop désorienter les usagers passant d'un logiciel à l'autre, il reste que les fonctionnalités propres à un logiciel se présentent par des menus, des icônes et des formulaires spécifiques à chacun des logiciels.

Interface d'information

Lorsque la gérance du système d'exploitation et du logiciel est couverte, le reste de l'espace d'interface revient à l'information spécifique. Même dans cet espace, ce n'est pas seulement du texte et des images qui sont disponibles. On retrouve en effet des zones sensibles qui permettent d'interagir même à partir de l'espace d'information. Les liens sur Internet en sont précisément des exemples.

Compétences relatives à l'interface

En abordant un document ou un logiciel-outil, l'utilisateur doit bien distinguer ce qui relève de l'appareil et du logiciel et ce qui revient à l'information spécifique. C'est une constatation facile avec un traitement de textes, par exemple, où l'espace pour écrire est facilement identifiable même par les néophytes mais cette constatation est plus difficile lors de l'examen d'un CD-ROM ou d'un site Internet. Le traitement de textes demeure un bon exemple si on le compare au papier. En média de surface, la feuille blanche suffit. En média d'interface, on trouve beaucoup de menus, des barres d'outils et même des modes de visualisation variés (mode page, déroulement ou pleine page, etc.). Il est donc facile dans le cas du traitement de textes de bien partager les compétences relatives à l'interface : celle de l'écriture qui s'exprime dans l'écran blanc et celle du fonctionnement par l'interaction initiée à partir des menus, des icônes, des barres d'outils et des formulaires à remplir.

2. L'information

Ce n'est pas le lieu d'expliquer en quoi consiste l'information spécifique. La présence de ce terme dans cette série vise seulement à bien préciser que l'information textuelle et iconique occupe et doit occuper la plus grande partie de l'espace-interface.

Deux aspects relatifs à l'information méritent considération : **la mise en écran** qui est l'équivalent informatique de la mise en page et **la structuration de l'information**. La mise en écran doit veiller à respecter certains paramètres comme la lisibilité; le regroupement de l'information en des zones spécifiques, par exemple pour l'aide ou pour des précisions d'appoint. La mise en écran doit aussi compter sur les éléments iconiques, les couleurs et toutes sortes de moyens pour attirer l'attention de l'utilisateur. On pense alors aux clignotements, aux animations, aux banderoles déroulantes, etc. Cette mise en écran s'éloigne encore de la mise en page par le fait que l'écran est souvent déroulant et que les écrans ne sont pas spécifiquement liés les uns aux autres comme les pages numérotées d'un livre.

La structuration de l'information est l'art ou la technique d'organisation de la matière en tenant compte du peu d'espace réservé au texte sur un écran et des possibilités de faire rapidement des liens vers d'autres écrans. Cette structuration doit respecter les ressources de la mémoire de l'utilisateur qui ne peut toujours se souvenir des écrans antérieurs et qui ne peut apprécier un écran trop touffu et mal organisé. Mais ces considérations relatives à l'information touchent autant l'utilisateur que le logiciel proprement dit. C'est pourquoi il faut distinguer entre un logiciel-outil, où l'information potentielle relève de l'agent utilisateur et le logiciel-média où l'information fait partie intégrante du produit.

3. L'interactivité

Si l'interface de l'ordinateur était passive comme celle de la télévision classique, l'outil et le média-ordinateur perdraient leur sens. En effet, tout l'intérêt de l'ordinateur tient à ses capacités interactives. Essentiellement un acte interactif consiste à constater un objet dans l'interface, par exemple une icône sur un écran, et à susciter une action sur cet objet désiré par une sélection souvent accomplie par la souris ou par un clavier quelconque. Dans ce texte, on explique d'abord le concept d'interactivité, on énumère ensuite les cinq styles d'interactivité exprimés du côté de l'ordinateur et de son logiciel et on énumère enfin les quatre sortes d'interactivité relatives à l'utilisateur.

Description de l'interactivité

L'interactivité est la capacité de réaction de l'ordinateur aux demandes de l'utilisateur, sa capacité de rétroaction ou de « feedback ». Pour comprendre l'interaction homme-machine, on a recours au langage d'échanges commerciaux utilisé entre des gens qui ne partagent pas une même langue. L'étranger désigne l'objet, indique l'objet d'échange, c'est l'action, et la transaction s'effectue. Très tôt, chaque interlocuteur s'entend sur un « programme » ou des mots partagés pour effectuer les transactions. Ce qui valait pour les grands voyageurs et découvreurs, s'applique de nos jours à l'informatique interactive. Le logiciel est donc programmé analogiquement à cette sorte de langage primitif appelé « Pidgin ». Ce langage simpliste comprend l'objet et l'action. Dans l'écran, l'objet est souvent représenté par une icône et le clic de la souris constitue la demande d'action. Selon sa programmation, l'ordinateur comprend cette demande et exécute l'action correspondante en réponse ou « feedback ».

Les styles d'interaction

Dans l'état actuel de l'informatique, on retrouve cinq styles d'interaction : la sélection par menu, la formule à remplir, le langage de commande, le langage naturel et la manipulation directe (Shneidermann, 1992).

a) L'interaction par **sélection dans un menu** raccourcit l'apprentissage, diminue le nombre de touches à taper, structure la prise de décision, permet l'usage d'outils gérés par dialogue et assure une bonne gestion des erreurs.

b) L'interaction par **formule à remplir** simplifie l'entrée des données, demande peu d'apprentissage, rend l'aide facile et permet l'usage d'outils gérés par la forme comme les cases à remplir.

c) L'interaction par **le langage de commande**. C'est l'approche la plus flexible qui est excellente pour les experts. Elle supporte bien l'initiative de l'utilisateur. Elle est cependant longue à apprendre. L'utilisateur doit connaître toutes les commandes par cœur, ce qui n'offre pas une excellente courbe d'apprentissage.

d) L'interaction par **le langage naturel** ne demande pas de syntaxe particulière à respecter comme avec les commandes mais l'interprétation reste difficile pour la machine. Ce n'est pas encore à point. Le mythe de l'ordinateur intelligent qui comprend la langue de tous les jours directement n'est pas encore accompli! Actuellement, c'est attrayant mais imprécis. En pratique, il n'est pas souvent utilisé.

e) L'interaction par **la manipulation directe**. Des icônes ou autres objets du monde virtuel comme des images ou des cartes permettent de visualiser les concepts. Elle est d'apprentissage et de rétention facile. Les erreurs sont évitées et l'exploration est encouragée. Elle procure un haut degré de satisfaction subjective.

L'interactivité et les usagers

L'interactivité étant l'échange entre un ordinateur avec son logiciel, d'une part, et un agent humain comme usager, d'autre part, on peut donc décrire cette interactivité du côté de ce qui est disponible à l'interface, ce qui vient d'être fait avec les cinq styles précédents et on peut aussi décrire cette interactivité du côté de l'appropriation par l'utilisateur. Quatre ordres ou degrés d'interactivité subjective sont alors énoncés.

i- L'interactivité-réflexe intervient lorsque le programme se comporte comme un jeu ou une scène artificielle (réalité virtuelle) où l'utilisateur réagit spontanément à une situation. Les jeux vidéo font largement appel à cette interactivité de premier niveau. Cette interactivité est exigeante et demande de l'habileté, ce qui s'accroît avec l'usage et la pratique. Par contre, comme elle est spontanée, elle ne sollicite pas tout l'intellect de l'utilisateur. Donc un jeune qui est bon avec des jeux vidéos ne développe pas des connaissances transférables en contexte d'apprentissage formel, si ce n'est l'appropriation à l'environnement technologique.

ii- L'interactivité fonctionnelle permet à l'utilisateur de se servir d'un outil informatique ou de lire un média informatique. Elle comprend divers degrés selon l'approfondissement du logiciel que l'utilisateur possède. Contrairement à l'interactivité réflexive, celle-ci n'a pas besoin d'être grande pour être efficace. Dès qu'un utilisateur est capable d'accomplir une tâche à l'ordinateur, cela signifie que son interactivité fonctionnelle est adéquate. L'interactivité fonctionnelle ne touche pas au talent cependant. On s'explique. Par exemple, ce n'est pas tout de connaître le traitement de textes, il faut aussi du talent pour écrire avec une machine interactive. Dans le jargon populaire, le temps d'apprentissage pour atteindre une interactivité fonctionnelle s'exprime comme sa courbe d'apprentissage. Un logiciel est aussi qualifié de convivial (user-friendly) si cette courbe est bonne, ce qui revient à un apprentissage rapide tout en permettant de bons résultats dans l'accomplissement d'une tâche.

À ce niveau l'apprentissage est cependant difficile car l'attention est presque toute centrée sur le fonctionnement et le déroulement. C'est le lot des jeunes internautes de tous âges qui n'ont pas atteint l'interactivité cognitive.

iii- L'interactivité cognitive est plus subtile. Elle relève de l'intelligence de l'utilisateur qui non seulement se sert d'interactivité fonctionnelle mais il s'en sert à des fins cognitives pour rechercher une information pertinente, pour structurer des données, pour écrire et lire, pour répondre à des questions, pour tirer profit d'une simulation, etc. À la différence de la précédente, cette interactivité se remarque aux tâches qui sont accomplies et aux effets qui sont obtenus. C'est celle qui est désirée dans un premier temps en éducation. Les autres formes d'interactivité ne sont que des préalables. À ce niveau, l'apprentissage est possible mais à des degrés divers.

iv- L'interactivité intégrale est une variante de l'interactivité cognitive et elle relève de l'expertise. Cette fois, l'utilisateur connaît tellement bien l'environnement informatique qu'il utilise qu'on peut dire qu'il ne s'agit plus de relation entre un humain et une machine mais que c'est plutôt un humain avec un machine qui réalise une tâche, un peu comme un ouvrier-expert utilise son marteau pour planter un clou : il est compétent lorsque son marteau plante le clou sans qu'il pense au marteau. Habituellement, l'interactivité intégrale n'est acquise qu'avec un nombre restreint de logiciels et lorsque l'utilisateur aborde un nouvel outil, il retombe souvent à une interactivité fonctionnelle. L'interactivité intégrale répond à la définition de la vertu, c'est une disposition stable pour bien agir dans un domaine.

À ce dernier niveau, l'apprentissage et la créativité sont rendus faciles, du moins tant que les paramètres technologiques ne sont pas trop changés.

4. L'icône et le curseur

L'icône est cette petite image souvent dessinée à gros traits qui surmonte une zone sensible à l'écran, une zone qui est aussi appelée bouton. Elle est un dépôt sur l'écran d'une fonctionnalité courante. Le problème principal de l'icône c'est qu'elle est conventionnelle et qu'elle doit être apprise parce que l'élément pictural qu'on y retrouve n'est pas forcément explicite pour le nouvel usager. (L'icône, nom masculin ou féminin inspiré des théories de Pierce est un symbole qui a une ressemblance avec la chose représentée, ce n'est pas le cas ici, c'est plutôt l'icône, nom féminin comme l'icône religieuse qui curieusement convient ici.) En ce sens, il arrive souvent, par redondance ou explicitation, qu'on sous-titre l'icône, ce qui la transforme alors en menu.

Lorsque plusieurs icônes accomplissent des fonctionnalités de même ordre, elles sont maintenant de plus en plus regroupées sous forme de barre d'outils. Le deuxième problème avec l'icône c'est qu'elle est attrayante, provocante. L'utilisateur qui broute ou navigue est facilement tenté de cliquer sur cette icône pour voir l'effet qu'elle provoque. Cette facilité dans le choix des actions tend à créer de trop rapides rafraîchissements d'écrans d'informations, ce qui désoriente l'utilisateur brouteur, navigateur ou explorateur, notamment sur Internet.

Le curseur est une variante de l'icône. Il indique le lieu de l'action à l'écran. Lorsqu'il est lié à la souris il spécifie par sa forme le genre d'action. Par exemple, la montre ou le sablier indiquent qu'il faut patienter, la petite main avec l'index sert précisément à indiquer, selon la métaphore du doigt qui appuie sur un bouton, tandis que le long I sert de point d'insertion dans un traitement de textes, etc.

5. La convivialité « user friendliness »

La convivialité représente le degré de facilité et de performance d'un environnement-logiciel. Le terme « user friendly » vient de ce que les ordinateurs étaient appelés des « amis », dans une perspective anthropomorphique. L'équivalent français ne laisse pas voir la même origine mais désigne les mêmes caractéristiques. Le degré de convivialité dépend de 5 facteurs :

1- La **durée d'apprentissage** du logiciel correspond au temps qu'un agent typique met pour apprendre les commandes pertinentes à une tâche. On peut croire que plus l'apprentissage est approfondi, meilleure est la performance.

2- La **vitesse de performance** exprime le temps nécessaire pour accomplir telle tâche.

3- Le **taux d'erreurs** s'exprime de diverses manières. Un logiciel est moins convivial si l'utilisateur commet souvent et facilement des erreurs, oublie souvent des paramètres, etc. La gravité des erreurs intervient aussi. Par exemple, le logiciel qui n'avertit pas qu'un programme est sauvegardé entraîne de graves erreurs. D'habitude, si le taux d'erreurs doit être bas, alors la vitesse de performance est réduite. Enfin, l'usage occasionnel tend à faire augmenter le taux d'erreurs.

4- La **réention** à long terme se manifeste comme la propriété intuitive qui fait qu'un usager occasionnel s'y retrouve facilement dans le logiciel sans grande perte de performance.

5- Pour obtenir une bonne **satisfaction subjective**, il faut obtenir des résultats rapidement. On peut trouver des cas où des compromis de convivialité sont acceptables, ce qui est rare en éducation. On pense alors à des logiciels dont l'incidence peut être critique pour la vie comme le contrôle aérien. Dans les cas de cette espèce, on accepte mieux un long apprentissage pour que, par la suite, les actions soient rapides et sans erreurs. En éducation, la convivialité revient à une question d'équilibre entre la satisfaction obtenue par la performance et un taux d'erreurs acceptable.

Encore avec Shneidermann (1992), on peut enfin énoncer quelques règles d'or relatives au design de logiciels conviviaux. Ces règles peuvent aussi servir pour apprécier ou évaluer des logiciels.

1- **L'uniformité des séquences** d'action facilite les usages répétitifs. En effet, contrairement à la croyance des nouveaux venus dans le domaine, il est préférable de toujours miser sur des commandes ou actions déjà connues, l'apprentissage et l'usage sont alors simplifiés.

2- Les habitués qui travaillent longtemps préfèrent utiliser **des raccourcis au clavier** plutôt que des menus, surtout pour les actions répétées comme les copier et les coller.

3- La **rétroaction** ou le feedback doit être approprié à l'action. Un gros bravo à l'écran est exagéré si une simple bonne touche a été activée. Par contre avant de tout effacer, il faut un avertissement plus musclé de la part du logiciel.

4- On connaît mieux le **résultat immédiat** comme le « What you see is what you get ». C'est une qualité qui dit qu'il est souhaitable que le résultat puisse être constaté au moment où l'action est posée. Les logiciels obéissent maintenant davantage à cette règle mais on rencontre encore des logiciels qui ne livrent pas de suite le produit final, ce qui désoriente l'utilisateur.

5- Les erreurs sont inévitables mais le **contrôle des erreurs** doit faire en sorte qu'elles ne soient jamais graves, c'est-à-dire qu'elles soient toujours facilement et rapidement réparables, notamment grâce à la règle suivante, celle du renversement. 6- Précisément parce que les erreurs sont humaines, le **retour à un état antérieur** connu comme l'annulation de la frappe ou le « undo » doit toujours être possible. Certains logiciels autorisent même plusieurs étages d'annulation.

7- Un bon logiciel laisse toujours le **contrôle à l'agent**, même pendant des tâches de traitement de l'information par la machine. Ce qui revient au système, c'est de toujours fournir une réponse ou feedback proportionné à l'action en cours.

8- Par respect pour la **mémoire à court terme** de l'utilisateur, le logiciel doit enfin ne jamais obliger l'utilisateur à se souvenir de contenus d'écrans qui sont maintenant passés.

En vérité, le logiciel qui respecte ces règles est généralement convivial mais il faut maintenant expliquer ce qu'on entend par le contrôle.

6. Le contrôle

Le contrôle de la transaction tantôt par l'utilisateur et tantôt par la machine est une conséquence de l'interaction. Cela revient à savoir qui mène entre l'agent, d'une part, et le système informatisé, d'autre part. Par exemple, un traitement de textes laisse un contrôle entier à l'utilisateur. Le correcteur de fautes passe par ailleurs le contrôle au logiciel. Un exercice et un tour guidé laissent le contrôle au logiciel la plupart du temps. Un logiciel hautement interactif serait celui qui passe le contrôle très souvent. On dit alors que le logiciel passe la main comme dans un jeu. Il n'y a pas de mal à ce que le contrôle passe par la machine et le logiciel, l'important c'est plutôt que l'utilisateur puisse en tout temps reprendre ce contrôle sur l'action car c'est lui qui doit demeurer maître du pouvoir de la machine.

7. La courbe d'apprentissage

La courbe d'apprentissage du logiciel est le rapport entre le temps d'apprentissage du logiciel et la capacité d'accomplir des tâches. Un logiciel trop simple qui ne permet pas la réalisation efficace de certains travaux possède une mauvaise courbe. Un logiciel très long à apprendre et compliqué à utiliser, notamment par des interactions trop nombreuses pour la tâche à accomplir aurait aussi une mauvaise courbe. On comprend qu'une tâche raffinée ne peut être accomplie avec un logiciel trop simple mais dans le choix du logiciel, tout est question de proportion car une mauvaise courbe entraîne une mauvaise convivialité comme on l'a vu plus haut.

8. Les médias

Depuis que le commerce s'est mis à utiliser le terme « multimédia » pour désigner un ordinateur performant, le terme a perdu de sa précision et de sa signification comme caractéristique du logiciel. Un multimédia peut vouloir dire qu'il est attrayant, qu'il a des images et des sons mais cela désigne rarement une caractéristique pédagogique. Dans la plupart des cas, l'accent est alors placé sur une fonctionnalité : graphisme, son, animation, communication ou commutation directe. Pour parler des divers sens utilisés par l'interface, on parle maintenant de multimodalité. Pour être significatif, il faudrait ajouter quelque chose au terme multimédia car, de nos jours, tout est multimédia et interactif.

Les documents-logiciels, les CD-ROMS et les sites Internet appartiennent à la catégorie des médias. Ce sont des documents-médias parce qu'ils contiennent de l'information spécifique et qu'ils relèvent de divers genres ou modèles comme on le voit dans un autre chapitre.