

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the text '2015/2016'.

2015/2016

Notes de cours

Module : Interface Homme Machine

EDITION N°1

Several thin, dark blue wavy lines that originate from the bottom left and curve upwards and to the right.

Enseignant : Samir BOULDJADJ
UNIVESITE FERHAT ABBAS- SETIF 1

Chapitre 1 – Qualité du logiciel et facteurs humains	1
1.1 Apprenabilité :	1
1.1.1 Observabilité	1
1.1.2 Causalité / Guidage	1
1.1.3 Consistance / cohérence	3
1.1.4 Familiarité, généricité	3
1.2 FLEXCIBILITE	3
1.2.1 Adaptation au contexte'	3
1.2.2 Adaptation à la diversité des utilisateurs.	4
1.2.3 Contrôle explicite	4
1.3 ROBUSTESSE	5
1.3.1 Gestion des erreurs	5
1.3.2 Charge de travail	6
Chapitre 2 – Conception des IHMs	8
1. Prise en compte de l'utilisateur	8
2. Modélisation d'un humain	8
3. Méthodes de conception centrées sur l'utilisateur	8
3.1 Étapes de conception	8
3.1.1 Phase d'analyse	9
3.1.2 Phase de conception	10
3.1.3 Phase d'évaluation	12
Chapitre 3 – Agencement de l'interface & Disposition des éléments	13
1. Organisation de l'espace	13
1.1 Parcours du regard	13
1.2 Visibilité / Accessibilité	13
1.3 Standardisation -Cohérence	14
1.4 Charte graphique	14
1.5 Zones de manipulation	14
2. Techniques de mise en évidence	15
3. Adaptation à la taille de l'écran	16
3.1 Mise en page adaptative	16
3.2 Mise en page responsive	17
4. Disposition dans les interfaces de type "formulaire"	17



Sommaire

Chapitre 4 – Composants de l'interface Éléments d'interaction	18
1. Graphisme	18
2. Les icônes	18
2.1 Interprétation des icônes.....	18
2.2 Conception des icônes	19
2.2.1 Quelques Recommandations	19
3. Les Couleurs	19
3.1 Utilisation de la couleur	19
3.2 Codage de la couleur	20
3.2.1 Quelques Recommandations	20
4. Typographie	20
4.1 Police de caractères.....	20
4.2 Recommandations typographiques.....	21
Chapitre 5 – IHM pour les interfaces mobiles.....	22
1. Introduction	22
2. Spécificités des mobiles	22
2.1 Des écrans réduits.....	22
2.2 Pas de survol	22
2.3 Le doigt et la main	22
2.4 Les comportements utilisateur.....	22
3. Critères d'ergonomie des mobiles	23
3.1 Clarté, lisibilité et information	23
3.1.1 Police et couleurs :	23
3.1.2 Codes et dénomination.....	23
3.1.3 Gestion du doigt.....	23
3.1.4 Feedback	23
3.2 Accélérateurs d'interaction	24
3.3 Faciliter la navigation	24
3.3.1 Boutons et onglets	24
3.3.2 Limiter l'effort de l'utilisateur	24
3.3.3 Satisfaction	24
Chapitre 6 – Accessibilité.....	25
1. Définition de l'accessibilité	25
2. But de l'accessibilité	25
3. Pourquoi l'accessibilité.....	25



Sommaire

4. Solutions.....	26
5. Conclusion	26

Chapitre 1 – Qualité du logiciel et facteurs humains

~ CRITERES DE QUALITE D'UN LOGICIEL ~

L'Interface Homme-Machine est domaine de l'informatique qui cherche à assurer que les systèmes informatiques sont **bien adaptés** aux **besoins des utilisateurs**.

Un logiciel est de bonne qualité quand il possède les 3 critères suivants :

- **Apprenabilité** : facilité avec laquelle l'utilisateur peut prendre en main le logiciel et découvrir ses fonctionnalités, elle est vérifiée par les propriétés suivantes : observabilité, causalité/guidage, consistance / cohérence, familiarité, généricité.
- **Flexibilité** : capacité du système à offrir des modes d'interactions multiples pour répondre aux besoins, préférences et expérience de l'utilisation (flexibilité) et à s'adapter au contexte (adaptabilité), elle est vérifiée par les propriétés suivantes : adaptation, styles d'utilisation, contrôle explicite.
- **Robustesse** : niveau de satisfaction dans la réalisation des tâches permises par le système (ne se limite pas à la fiabilité), elle est vérifiée par les propriétés suivantes : gestion des erreurs, charge de travail.

1.1 Apprenabilité :

1.1.1 Observabilité

C'est la facilité offerte à l'utilisateur de vérifier (voir) les effets de ses actions. Il existe 2 Types d'observabilité : directe où l'utilisateur voit le résultat de l'action immédiatement (ex. : supprimer un fichier sous Windows). Indirecte : le résultat de l'action nécessite une intervention de l'utilisateur (ex. ; supprimer un fichier sous Unix ('rm' suivie de 'ls'))

1.1.2 Causalité / Guidage

Causalité : c'est la capacité du système à aider l'utilisateur à prédire le comportement du système au vu d'interactions précédentes.

Guidage : c'est la capacité du système à conseiller, orienter, informer l'utilisateur lors de ses interactions, On distingue. Le critère de guidage est décomposé en quatre sous-critères : Incitation, Groupement / Distinction, Feedback immédiat, Lisibilité.

1.1.2.1 Incitation

Le critère d'incitation réunit les moyens visant à conduire l'utilisateur à effectuer des actions spécifiques. L'incitation aide l'utilisateur dans son interaction avec le logiciel en lui fournissant les éléments nécessaires pour l'utiliser correctement.

Recommandations :

- Griser les fonctions non disponibles (options de menu, boutons, ...)
- Fournir la liste des saisies attendues (listes déroulantes, codes à utiliser, ...)



- Donner le format de saisie des données (dates, dimensions, ...)
- Modifier la forme du curseur (pointeur de la souris) pour donner des indications sur l'opération à effectuer
- Indiquer clairement les champs obligatoires (* ou autre indicateur)
- Montrer clairement comment aller en avant et en arrière
- Afficher des bulles d'aides (Tooltips) sur les éléments non-triviaux (icônes des barres d'outils par exemple)

1.1.2.2 Groupements et distinctions :

Le critère de groupement consiste à guider l'utilisateur en groupant les informations et les fonctions de même type. Un utilisateur considère généralement que ce qui se ressemble va fonctionner de la même manière (principe de similarité). Deux attributs graphiques sont utilisés dans ce but : Le format de présentation (couleur, forme, syntaxe, ...) et La position dans l'interface (localisation, bordure, ...)

A l'inverse, le critère de distinction consiste à séparer clairement ou à attribuer des propriétés distinctes à des objets qui sont différents ou qui se comportent de manière différente.

Recommandations :

- Regrouper les informations de même type par le format ou par la position (regroupement par menus, entourer avec une bordure, ...)
- Distinguer par une présentation différente ou un emplacement différent, les informations distinctes pour éviter les risques de confusions

1.1.2.3 Feedback immédiat

Le critère de feedback immédiat réunit tous les éléments qui servent à montrer à l'utilisateur ce qu'est en train de faire le système et en lui offrant une rétroaction en réponse à chacune de ses actions. Le respect de ce critère, appelé parfois "retour utilisateur", contribue grandement à accroître la confiance de l'utilisateur.

Recommandations :

- Le logiciel doit répondre à toute action de l'utilisateur par un changement dans la présentation de l'interface (visuel, sonore, ...)
- Indiquer les modes de fonctionnement du système (état)
- Signaler les traitements longs par une indication d'attente (sablier, barre de progression, animation, message, ...)
- Toujours faire apparaître les saisies utilisateur
- Rendre visible les traitements réalisés par le logiciel et indiquer clairement lorsqu'ils ont échoué

1.1.2.4 Lisibilité

Le critère de lisibilité consiste à faciliter la perception des informations textuelles et iconographiques par un choix judicieux de leurs propriétés et de leur disposition. Une attention particulière doit être apportée aux : **polices, couleurs, disposition et icônes**

Recommandations :

- Utiliser une police de caractère lisible (à l'écran, éviter les italiques en petite taille, ...)
- Soigner les espacements (interligne) et les alignements (justification)
- Adapter la taille des libellés et des icônes afin de garantir leur interprétation
- Soigner les contrastes (choix des couleurs pour le fond et le premier plan)
- Prendre en compte la configuration de la machine cible (résolution, ...)

1.1.3 Consistance / cohérence

C'est l'homogénéité globale de l'interface homme-machine. L'objectif est de respecter une logique cohérente pour la présentation (graphisme, localisation, vocabulaire, format, syntaxe, ...), le comportement (réaction du système, messages, retours sonores, ...).

L'homogénéité rend le système stable, donc prévisible aux yeux de l'utilisateur. Le temps de recherche de l'information est diminué.

Recommandations :

- Utiliser le même schéma d'agencement pour toutes les fenêtres (gabarit d'écran appelé aussi tracé régulateur)
- La sémantique des boutons de la souris doit être constante
- Le même vocabulaire doit être utilisé pour désigner les mêmes fonctions
- Utiliser une organisation et une syntaxe cohérente pour les menus
- Utiliser de manière cohérente les symboles graphiques (icônes, couleurs, ...)

1.1.4 Familiarité, généricité

Le critère de Familiarité concerne la cohérence avec nos connaissances antérieures : monde réel, et l'expérience acquise. Il est réalisé par l'utilisation des métaphores (ex. Touche entrée des machine à écrire), l'affordance (Capacité qu'a un objet, par ses propriétés intrinsèques (forme...) à suggérer à l'utilisateur l'action qu'il doit faire avec lui) et la standardisation (Certaines interfaces, certaines notions sont devenues des standards de facto : on passe d'une généricité à une vraie familiarité)

1.2 FLEXIBILITE

1.2.1 Adaptation au contexte'

Le premier sous critère 'adaptation au contexte' concerne les moyens mis à la disposition des utilisateurs pour personnaliser l'interface afin de tenir compte des préférences, des aptitudes, des habitudes, du contexte d'utilisation et de la manière de travailler de chacun d'eux. Il vise également à offrir à l'utilisateur la possibilité d'effectuer une tâche ou activer une fonction de différentes manières, par exemple : Par un (menu déroulant, menu contextuel, icône dans une barre d'outils ou raccourci clavier)

Recommandations :

- Permettre d'effectuer les tâches (fonctions) à la fois au clavier et à la souris
- Autoriser le déclenchement d'une commande fréquente depuis plusieurs endroits dans l'application.
- Permettre à l'utilisateur de paramétrer le logiciel selon ses préférences

- Fournir un moyen rapide d'accéder aux commandes des menus (raccourcis)

1.2.2 Adaptation à la diversité des utilisateurs

Le deuxième sous-critère de la flexibilité est la prise en compte de l'expérience de l'utilisateur qui concerne les moyens mis en œuvre pour s'adapter aux différents niveaux d'expérience des utilisateurs.

Pour satisfaire ce critère, des moyens différenciés doivent être prévus pour tenir compte des différences d'expérience qui existent entre les utilisateurs selon leur profil. Il est à noter que l'expérience des utilisateurs peut varier dans le temps.

1.2.3 Contrôle explicite

Le critère de contrôle explicite concerne les aspects liés au degré de maîtrise qu'a l'utilisateur sur les traitements réalisés par le système interactif. Il se décompose en deux sous-critères élémentaires : Actions explicites et Contrôle utilisateur.

Quand les opérations du système résultent directement des actions des utilisateurs, on observe moins d'erreurs et la compréhension du fonctionnement de l'application est facilitée (représentation mentale). C'est un facteur important d'acceptation du logiciel : les utilisateurs n'aiment pas être menés par le bout du nez et se sentir asservis à la machine. Si l'utilisateur a, en tout temps, le contrôle du dialogue, les réactions du système sont mieux prévisibles et l'apprentissage s'en trouve facilité

1.2.3.1 Critère d'actions

Le critère d'actions explicites concerne la relation qui existe entre les actions de l'utilisateur et les traitements qui seront effectués par l'application en réaction à ces actions. Seules les opérations explicitement demandées par l'utilisateur doivent être effectuées par le système. Des automatismes peuvent être tolérés s'ils ont été explicitement acceptés par l'utilisateur. Les opérations devraient être effectuées au moment où on les invoque sauf naturellement s'il s'agit d'actions qui sont explicitement différées, par leur nature ou selon les souhaits de l'utilisateur (déclenchement temporisé, synchronisation avec la fin d'un autre processus, etc.).

Recommandations :

- Ne pas déclencher d'opérations sans le consentement explicite de l'utilisateur.
- Déclencher l'opération immédiatement après l'action de l'utilisateur ou, sinon, indiquer clairement que l'opération sera différée (ou qu'elle ne peut pas être effectuée)

1.2.3.2 Contrôle utilisateur

Le critère de contrôle utilisateur concerne le fait que l'utilisateur doit toujours avoir la main sur le système et en contrôler les opérations et leur déroulement (interrompre, reprendre). Il vise à rendre l'utilisateur autonome dans son interaction avec le système en lui donnant la maîtrise du processus, l'utilisateur doit avoir, en permanence le contrôle sur le logiciel. Si possible, ses actions devraient être facilitées et des options de contrôle appropriées devraient lui être fournies en fonction de l'état actuel du système (prise en compte de la situation courante).

Recommandations :

- Offrir à l'utilisateur une validation explicite des commandes importantes ou difficilement réversibles
- Offrir la possibilité d'interrompre les traitements longs
- Autoriser les retours en arrière (Undo)
- Permettre, en tout temps, de quitter la fonction courante ou même, le logiciel

1.3 ROBUSTESSE

1.3.1 Gestion des erreurs

Le critère de gestion des erreurs regroupe les différents moyens visant à éviter ou à réduire les erreurs de l'utilisateur et lui permettre, le cas échéant, de les corriger afin de conserver l'intégrité du système. L'objectif prioritaire est de minimiser les interruptions dues aux erreurs (maximiser la performance, l'efficacité, la productivité). Trois sous-critères sont à prendre en compte :

- Protection contre les erreurs (éviter que l'utilisateur n'en commette)
- Qualité des messages d'erreur (informer clairement l'utilisateur)
- Correction des erreurs (lui permettre de les corriger)

Recommandations :

- Tout mettre en œuvre pour éviter les erreurs (attitude défensive)
- Engager un dialogue lorsque des actions de l'utilisateur peuvent conduire à des situations irréversibles (ou des conséquences fâcheuses)
- Avertir l'utilisateur au plus tôt et le guider vers la résolution du problème
- Faciliter l'exploration et l'apprentissage du système

1.3.1.1 Le critère de protection contre les erreurs

Le critère de protection contre les erreurs vise à éviter que l'utilisateur commette des erreurs et que, si cela se produit, il en soit averti immédiatement (c'est en fait du guidage). L'objectif est d'éviter que les interruptions dues aux erreurs ne nuisent au bon déroulement de la tâche. On prévient les erreurs en guidant l'utilisateur de manière à ce qu'il ne soit pas enclin à en commettre (lien avec critère de guidage/incitation). Si une erreur survient malgré tout, il faut la détecter au plus tôt et avertir l'utilisateur avec un message clair.

Recommandations :

- Mettre en évidence les commandes non disponibles (griser)
- Fournir la liste des valeurs possibles, des unités, ...
- Détecter les erreurs au plus tôt et avertir immédiatement l'utilisateur
- Minimiser les saisies au clavier (si possible, listes à choix)
- Prévenir les risques de perte de données (demander confirmation)

1.3.1.2 Qualité des messages d'erreur

Le critère de qualité des messages d'erreur concerne la pertinence, la facilité de lecture et l'exactitude des informations données aux utilisateurs sur la nature des erreurs commises ainsi que sur les actions à

entreprendre pour les corriger. Pour que la correction de l'erreur soit aisée, le message d'erreur doit indiquer la nature de l'erreur, sa cause, ainsi que les moyens de la corriger (la qualité des messages favorise l'apprentissage du système). Le contenu des messages doit tendre à dédramatiser les erreurs aux yeux de l'utilisateur (diminue son sentiment de culpabilité et le stress induit).

Recommandations :

- Placer les messages d'erreur là où l'utilisateur est censée regarder
- Afficher des messages d'erreur explicites (utiliser le langage de l'utilisateur)
- Éviter les textes trop longs (rester bref, utiliser des liens, des références, ...)
- Éviter les textes réprobateurs
- Faire en sorte, si possible, que les textes soient auto-suffisants. Trouver le bon compromis entre concision et exhaustivité

1.3.1.3 Correction des erreurs

Le critère de correction des erreurs inclut tous les moyens mis à la disposition des utilisateurs pour leur permettre de corriger leurs erreurs. Offrir à l'utilisateur les moyens de corriger le plus facilement et le plus rapidement possible ses erreurs en lui permettant, par exemple, de ne corriger que la portion des données qui est erronée et ne pas le contraindre à ressaisir l'ensemble des informations. Plusieurs tactiques peuvent être utilisées selon le type et la gravité de l'erreur :

- Bloquer l'utilisateur tant que l'erreur subsiste (erreurs graves)
- Lui permettre de continuer après une mise en garde (message, signal sonore)
- Ne pas répondre à la commande erronée (un message est nécessaire)
- Corriger automatiquement l'erreur (dans quelques rares cas)

Recommandations :

- Mettre en évidence le champ ou l'élément erroné
- Permettre d'annuler une action ou une série d'actions (par ex. supprimer des articles dans le caddie virtuel d'une application e-commerce).
- Proposer des alternatives (par ex. "Related topics" pour une recherche infructueuse)

1.3.2 Charge de travail

Le critère de charge de travail regroupe l'ensemble des moyens visant à réduire la charge perceptive, mnésique et physique de l'utilisateur. L'objectif est d'allouer le maximum de ressource au système cognitif (raisonnement) et de minimiser les risques d'erreur en réduisant le nombre de stimuli du système sensoriel et le nombre d'activités motrices. On doit donc viser à minimiser à la fois la quantité d'informations que l'utilisateur doit prendre en compte ainsi que le nombre d'actions élémentaires qu'il doit réaliser pour accomplir une tâche donnée.

Le critère de la charge de travail se décompose en deux sous-critères élémentaires : la Brièveté et Densité informationnelle

1.3.2.1 Brièveté

Le critère de brièveté regroupe l'ensemble des moyens visant à réduire la charge perceptive et mnésique de l'utilisateur dans ses interactions avec les composants d'entrée ou de sortie de l'interface

utilisateur (champs d'un formulaire, menus, widgets, ...). Il s'agit donc notamment de limiter autant que possible le travail de lecture et de saisie de l'utilisateur (on parle de concision). Ce critère inclut également la notion d'action minimale qui vise à minimiser le nombre d'actions nécessaires pour atteindre un but ou accomplir une tâche.

Recommandations :

- Limiter le nombre d'options dans un menu ou dans une liste déroulante
- Éviter les libellés trop longs
- Réduire le nombre d'actions élémentaires pour atteindre un objectif donné
- Éviter à l'utilisateur d'avoir à se souvenir d'informations d'une fenêtre à l'autre, d'avoir à faire des calculs ou de saisir des informations qui peuvent être déduites par le système.
- Pas plus de 5 à 7 items à mémoriser sur un display : décomposer la tâche sinon

1.3.2.2 Densité informationnelle

Le critère de densité informationnelle concerne la charge perceptive et mnésique pour l'ensemble des éléments d'une interface utilisateur. La performance des utilisateurs est influencée négativement quand la charge informationnelle est trop élevée ou (c'est plus rare) trop faible. Il faut donc supprimer tous les éléments sans lien direct avec la tâche en cours et qui pourraient distraire inutilement les utilisateurs. Les bannières publicitaires (ou fenêtres pop-up) sont des exemples d'augmentation (inutile ?) de la densité informationnelle de l'interface.

Recommandations :

- N'afficher que les informations pertinentes pour effectuer la tâche (boîtes de dialogue simples, représentations graphiques, ...)
- Éviter les écrans trop chargés (décomposer si nécessaire)
- Éviter les liens trop nombreux dans un texte affiché sur une page web
- Éviter les textes trop verbeux (dialogue simple, phrases courtes)
- Privilégier la reconnaissance (symboles, icônes)

Chapitre 2 – Conception des IHMs

1. Prise en compte de l'utilisateur

Une des clés pour réussir à concevoir une interface ergonomique est de connaître et comprendre ses utilisateurs. L'humain doit constituer l'élément central à considérer dans la conception d'une interface homme-machine. La prise en compte des utilisateurs doit intervenir assez tôt, dès la phase d'analyse d'un projet logiciel. Ils doivent être impliqués dans toutes les phases du cycle de développement.

Même si la prise en compte de l'utilisateur ne facilite pas la tâche du développeur, c'est indispensable de l'impliquer si l'on veut créer une expérience utilisateur de qualité. En plus de compétences techniques, le développeur doit donc, en plus, posséder des compétences humaines et relationnelles, notamment :

- une bonne capacité d'écoute et de communication (savoir argumenter)
- des talents de négociateur (souplesse)
- la faculté de pouvoir extraire les informations importantes parmi des données peu structurées
- la capacité de se mettre à la place de l'autre (empathie)
- une bonne dose de créativité et le sens de l'initiative

Pour éviter de commettre des erreurs grossières de conception, il est aussi indispensable de connaître quelques caractéristiques générales des humains (perception, comportement, ...).

2. Modélisation d'un humain

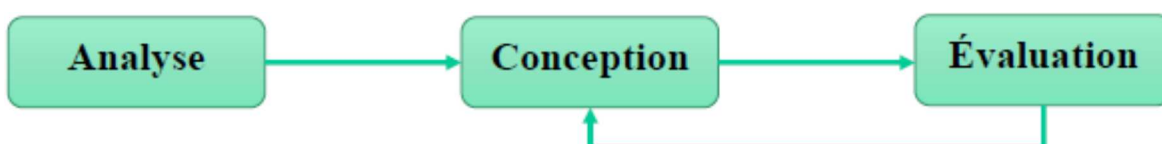
De nombreuses tentatives ont été entreprises pour modéliser le comportement des humains (et les recherches continuent). Malheureusement, il n'y a pas d'unification dans ces modèles qui relèvent, pour la plupart, de la psychologie cognitive. Même si ces modèles ne peuvent pas servir de preuve (le problème général est trop difficile), ils permettent malgré tout, par leur pouvoir expressif, d'offrir un support formalisé pour le raisonnement, de travailler sur des bases rationnelles et d'effectuer des validations expérimentales.

On distingue différents niveaux d'abstraction : Théorie de l'Action, Modèle de Rasmussen, Modèle du processeur humain (et bien d'autres : GOMS, ICS, CCT, Keystroke, ...)

3. Méthodes de conception centrées sur l'utilisateur

3.1 Étapes de conception

La démarche de conception orientée utilisateur consiste à mettre en place un processus itératif s'appuyant sur l'analyse de l'expérience utilisateur. Dans une démarche de conception centrée sur l'utilisateur, on distingue généralement trois phases dans le processus itératif de développement :



Chapitre2 : Conception des IHMs

Phase d'analyse : elle vise à :

- préciser les attentes et le besoin des utilisateurs finaux
- prendre connaissance de la tâche réelle des utilisateurs
- prendre connaissance du contexte d'utilisation
- permet de préciser l'utilité recherchée par les utilisateurs de l'application

Phase de conception : on se basant sur les résultats de l'analyse :

- Concevoir une première maquette de l'interface (prenant compte l'analyse de la tâche des utilisateurs et les principes d'ergonomie des interfaces)
- •Évoluer cette maquette en réalisant des maquettes de plus en plus précises à chaque cycle (corrigées et affinées en fonction des retours des évaluations). consiste à :
- •Évaluer, mesurer l'utilisabilité de l'interface
 - Effectuer une évaluation analytique
 - Présenter le résultat de la conception à l'utilisateur, observer les difficultés rencontrées, prendre son avis, ...
- •Identifier les points à améliorer dans la version suivante (prochain cycle)

3.1.1 Phase d'analyse

Durant la phase d'analyse, plusieurs techniques peuvent être utilisées pour cerner le problème et préciser les attentes des utilisateurs :

- **Enquête/interviews des utilisateurs**
 - Panel d'utilisateurs représentatifs
 - Interviewés généralement individuellement (en leur posant des questions concernant leurs attentes, souhaits, informations qu'ils aimeraient trouver sur le site,...)
- **Focus group** méthode d'enquête qui vise à recueillir l'opinion des utilisateurs sur un produit ou un concept d'un nouveau produit, logiciel, site web, ... (méthode employée en marketing)
 - Discussion de groupe (avec utilisateurs ou futurs utilisateurs potentiels)
 - Recueillir les opinions et les motivations des participants (éléments subjectifs)
- Analyse de la tâche des utilisateurs
- Analyse de la situation (contexte)

3.1.1.1 Analyse de la tâche

L'analyse de la tâche a pour but de recueillir des informations sur la manière dont les utilisateurs effectuent l'activité pour laquelle le système est développé. On procède généralement en deux étapes :

- Des interviews permettent de se faire une première idée
- L'observation ensuite des utilisateurs sur le lieu de travail de manière à déterminer l'activité effectivement réalisée

La modélisation de la tâche comportera notamment :

- Les buts des utilisateurs et manière de les atteindre
- Les informations nécessaires pour accomplir la tâche
- Le traitement des cas exceptionnels (traitement des urgences pour les applications critiques)

Chapitre2 : Conception des IHMs

L'analyse de la tâche servira à structurer l'interface utilisateur :

- Découpage en fenêtres, onglets, ...
- Structuration des fonctions, organisation des menus, ...
- Informations à afficher dans l'interface

3.1.1.2 Analyse de la situation

L'analyse de la situation consiste à prendre connaissance du contexte dans lequel les utilisateurs vont se servir du logiciel. On procède généralement par observation, interview ou questionnaire dans le contexte d'utilisation (sur le lieu de travail). Les informations recueillies permettent d'ajuster le logiciel à la population ciblée. On déterminera notamment :

- Les connaissances informatiques des utilisateurs
- Leurs connaissances du domaine applicatif
- L'environnement général d'utilisation (éclairage, bruit, gants, ...)
- Fréquence et durée d'utilisation du système
- Encadrement, formation (prévue) des utilisateurs

Ces éléments seront importants pour déterminer :

- Le type de guidage que devra offrir le logiciel
- Le mode de dialogue le mieux adapté
- Arbitrer le compromis : facilité d'apprentissage et rapidité d'utilisation
- Charge mentale supportable, types d'interactions, de stimuli, etc.

3.1.2 Phase de conception

Trois méthodes sont généralement employées durant la phase de conception : Les personas, Le tri de cartes, Le maquettage (prototypage).

Durant la phase de conception, il est important de garder en tête la diversité des utilisateurs (il n'existe pas de conception "prêt à porter"). Dans l'absolu, il n'y a pas de bons logiciels ou de bons sites web. Cela dépend des utilisateurs et du contexte dans lequel ils s'en servent. Dans cette population, il faut plutôt s'intéresser à caractériser la dispersion et identifier des familles d'utilisateurs (cluster).

3.1.2.1 Persona

La méthode des persona permet de construire des archétypes des utilisateurs de l'application à la manière des personnages d'un film ou d'une pièce de théâtre auxquels les concepteurs pourront se référer lors du design de l'interface. Elles permettent au team de développement de donner un visage humain au groupe cible. Les persona permettent, au sein d'un team de développement, de partager une certaine vision des utilisateurs cibles et d'orienter certains choix de conception en se mettant à leur place.

La construction des persona se fait en deux étapes :

• **Phase de recherche** : série d'interviews auprès d'un panel d'utilisateurs six (6 à 8 utilisateurs) pour collecter un ensemble de données constitutives des futurs personas :

- âge, genre, profession, catégorie socioprofessionnelle
- Leurs besoins ou leur motivation vis-à-vis de l'application.

Chapitre2 : Conception des IHMs

- Les critères de choix.
- La façon de se servir de l'application.
- L'expertise du domaine.

●**Phase d'agrégation** : ces données permettent de construire les personas proprement dits.

Ensuite Identifier les objectifs d'utilisation de chaque persona depuis les caractéristiques comportementales. En effet, l'interface sera facile à utiliser dès lors qu'elle répond aux objectifs de l'utilisateur.

3.1.2.2 Tri par carte

Le tri par carte est une méthode conçue pour construire l'architecture de l'information d'une application informatique : arborescence des rubriques d'un site web, structure des menus ou organisation des fonctionnalités d'un logiciel.

Le tri par carte est mis en œuvre au début de la phase de conception car il permet de construire la structure de base sur laquelle s'appuie l'interface à la fois en termes d'organisation et de terminologie (vocabulaire). L'exercice peut se pratiquer avec des cartes physiques ou avec des cartes virtuelles (il existe des logiciels spécialisés).

Chaque carte a un titre (un ou 2 mots-clés) qui peut être modifié par l'utilisateur(s), sous le titre, une ou deux phrases décrivent plus en détail l'information en question. + éventuellement une image (par exemple celle du produit concerné lorsqu'il s'agit d'un site de commerce).

Le tri par carte est réalisé en trois étapes :

- validation des libellés : on demande aux participants de vérifier la conformité des titres de chaque carte avec le texte descriptif situé en dessous. Si ce n'est pas le cas, les participants peuvent modifier le titre de la carte en employant un libellé qui leur semble mieux adapté.
- groupement : on demande aux participants de regrouper les cartes « qui se ressemblent » en construisant des familles.
- dénomination des groupes : on demande aux participants de donner un nom à chacun des groupes qu'ils viennent de construire. (Éventuellement, ils peuvent regrouper certains groupes pour former un « groupe de groupes » auquel il donne également un nom).

Il y a deux grandes familles de tris de cartes ; Le tri ouvert ou tri montant où Les participants définissent leurs propres groupes (auxquels ils donnent des noms) par contre dans le tri fermé, le nom des groupes est donné aux participants, Ils doivent placer les cartes dans les groupes imposés, il est employé pour valider le tri ouvert.

Les utilisateurs trient généralement les cartes individuellement (pour limiter les effets de groupe) mais des tris effectués en groupes sont aussi pratiqués. Des études expérimentales ont montré qu'avec 15 participants on obtient un bon résultat.

3.1.2.3 Maquettage

Le maquettage ou prototypage consiste à concevoir des versions préliminaires et intermédiaires de l'interface avant de finaliser les spécifications conduisant au développement du produit final.

Chapitre2 : Conception des IHMs

Les maquettes constituent des versions jetables de l'interface alors que les prototypes sont généralement développés sur la même base technologique que le produit final. Dans le domaine des interfaces ces deux termes sont cependant assez interchangeables.

La fidélité de la maquette désigne sa ressemblance par rapport à l'interface finale en termes de graphisme et d'interactivité. Les maquettes statiques n'autorisent aucune interaction, elles sont parfois réalisées sur papier (on peut simuler les enchaînements en préparant des story-boards : séquences de vues représentant une fonction). Dans une maquette fil de fer (*wireframe*), le graphisme n'est volontairement pas travaillé et on n'utilise généralement pas la couleur.

Le prototypage horizontal consiste à établir une maquette statique de toutes les vues de l'application. Il n'y a aucune interaction possible par contre Le prototypage vertical consiste à se concentrer sur certaines vues et à coder les fonctionnalités et l'interaction. L'utilisateur peut ainsi dérouler un scénario typique et voir l'enchaînement des différentes vues et le comportement de l'interface.

3.1.3 Phase d'évaluation

Les méthodes d'évaluation vont permettre de mesurer l'ergonomie de l'application afin d'identifier les points à améliorer en particulier entre deux itérations de la maquette. Les deux méthodes les plus fréquemment employées sont :

- l'audit ergonomique qui consiste à analyser l'interface au regard d'un ensemble de critères ergonomiques reconnus.
- test utilisateur où l'utilisateur est placé en situation réaliste d'utilisation afin d'identifier les difficultés qu'il rencontre.

Lors de la phase d'évaluation, plusieurs métriques peuvent être utilisées, par exemples : tâches réussies, temps d'exécution, erreurs commises, efficacité, facilité d'apprentissage, problèmes rencontrés, éléments subjectifs, métriques comportementales et physiologiques,...

Chapitre3 –Agencement de l'interface & Disposition des éléments

1. Organisation de l'espace

L'agencement des informations et des éléments d'interaction joue un rôle important dans l'utilisabilité d'une interface. La vision constitue le système perceptif le plus développé chez l'humain. Il faut donc accorder une attention particulière à la manière de présenter les éléments dans l'interface.

1.1 Parcours du regard

L'utilisateur ne parcourt pas toujours l'écran de la même manière. Des études expérimentales ont montré que l'utilisateur adopte deux stratégies différentes :

- Lors de la première vision de l'écran : Exploration rapide, l'utilisateur adopte un parcours en 'Z' (le regard part du coin supérieur gauche de l'image, parcourt systématiquement la zone centrale et se termine dans le coin inférieur droit).
- Lors des consultations ultérieures : Recherche sélective, l'utilisateur connaît l'image et il positionne son regard sur des emplacements qui lui semblent pertinents (là où il s'attend à trouver l'information qu'il recherche).

Remarque : Sur certains types de pages web (moteurs de recherche, blogs, portails, e-zines, ...) des études ont montré que le regard d'une majorité d'internautes suit un tracé en forme de 'F' (F-Pattern). L'utilisateur lit d'abord horizontalement les premières lignes puis parcourt ensuite les lignes suivantes en allant toujours moins loin sur la droite, finalement, l'utilisateur parcourt verticalement la partie gauche de l'écran.

1.2 Visibilité / Accessibilité

La visibilité et l'accessibilité des zones de l'écran sont également des points à prendre en compte lors de la conception des interfaces. Pour les interfaces utilisées avec une souris (ou autre dispositif de pointage), on considère que les zones proches des coins sont plus difficile à atteindre (le mouvement est plus contraint).

Très visible peu accessible	Très visible Bien accessible	Assez bien visible peu accessible
Très visible Bien accessible	Zone la plus visible Zone la plus Bien accessible	Très visible Bien accessible
Peu visible Peu accessible	Peu visible Bien accessible	Peu visible Peu accessible

Chapitre 3 : Agencement de l'interface & Disposition des éléments

1.3 Standardisation -Cohérence

Plus la charge de travail augmente plus l'utilisateur standardise ses procédures d'exploration et s'achemine vers la recherche sélective en se focalisant sur certaines zones de l'écran. Un point important à considérer dans la phase de conception est la cohérence de l'interface :

- Positionnement des éléments
- Graphisme (couleurs, taille, polices, symboles, ...)
- Comportement

L'homogénéité de l'ensemble des interfaces d'une application ou d'une suite logicielle renforce le sentiment de confiance, accélère l'apprentissage et accroît la fidélisation des utilisateurs. Pour assurer cette cohérence, on utilise des gabarits d'écran qui précisent l'agencement général des contenus et des commandes pour les différents types de fenêtres ou pages de l'application.

Si l'interface doit s'adapter à différents types de périphériques (avec des tailles d'écran différentes et des orientations variables), un design adaptatif (responsive design) doit être envisagé.

Des gabarits d'écran seront conçus pour les différentes déclinaisons de la présentation des interfaces :

- Adaptation du nombre de colonnes
- Abandon de certains contenus
- ...



1.4 Charte graphique

Les gabarits d'écran font souvent partie d'une charte graphique qui définit un ensemble d'éléments visant à assurer la cohérence de l'ensemble des vues d'une application, des pages web d'un site, etc. La charte graphique définira en outre :

- Les polices de caractères à utiliser
- La taille des éléments
- La palette de couleurs à utiliser (codage couleur)
- Le style de conception des icônes (ou une palette d'icônes)
- Les espacements et les marges (entre et autour des éléments)
- La justification des textes
- Les textures, les images d'arrière-plan
- Les ombrages
- Les bordures
- ...

1.5 Zones de manipulation

Dans les zones de manipulation des interfaces, on conseille de disposer, dans le sens de la lecture : Les éléments importants en premier → Les commandes fréquemment utilisées ensuite → Les autres éléments à la fin.

2. Techniques de mise en évidence

Un des nombreux buts du concepteur d'interface est d'attirer l'attention de l'utilisateur sur certains éléments importants (du point de vue de l'utilisateur, de ses objectifs, de ses tâches). Il existe de nombreux moyens de mise en évidence de certains éléments de l'interface mais toutes ces techniques ne sont pas équivalentes. Tout l'art consiste à les utiliser à bon escient, de façon cohérente et surtout avec parcimonie. Si l'on abuse de ces techniques, on obtient l'effet contraire en dispersant l'attention de l'utilisateur et en surchargeant son système perceptif (charge mentale).

Quelques règles générales :

- Limiter le nombre total de moyens de mise en évidence utilisés (la signification des différents modes sera plus facile à apprendre et à mémoriser pour l'utilisateur)
- Limiter le nombre de moyens de mise en évidence sur le même objet (l'efficacité n'est pas liée au nombre de moyens utilisés)
- La mise en évidence d'un objet ne devrait pas affecter la perception de son état (p. ex inversion vidéo fait perdre la couleur d'un texte)
- Si possible, la mise en évidence d'un objet doit s'arrêter dès la disparition de l'état correspondant ou dès la fin de l'action de l'utilisateur qui a déclenché sa mise en œuvre

Parmi les principales techniques de mise en évidence, on peut mentionner :

- Le clignotement / Les animations :
 - Attire l'attention même en zone périphérique de la vision
 - Forte charge perceptive (dérangeant)
 - A réserver à des stimulations importantes (urgences)
- L'inversion vidéo / Le surlignement
 - Intversion des couleurs de premier-plan et d'arrière-plan
 - Classique pour indiquer la ligne sélectionnée dans une liste
 - Peut réduire la lisibilité du texte
 - Un surlignement est souvent préférable à une inversion pure
- Les attributs du texte : **gras**, *italique*, souligné, ombré
 - Pour le gras et l'italique, l'effet dépend fortement de la police utilisée
 - Le souligné est généralement à éviter (sauf év. pour liens hypertextes)
- La police de caractères (texte, texte, texte, texte, ...)
 - Utile pour des textes longs (paragraphe)
 - Impact dépend du choix des polices
- La couleur
 - Être cohérent dans son utilisation (codage couleur)
 - Limiter le nombre de couleurs utilisées
 - Utiliser des couleurs distinguables (nommables)
 - Pour des mises en évidence importantes, à combiner avec une autre technique (penser aux Daltoniens)
- La taille des éléments (texte, icône, image, ...)
 - Structuration : titres, sous-titres, paragraphes, légendes, ...
 - Icônes : à réserver à des situations particulières
 - Risque de donner un aspect chaotique (menace la cohérence)

Chapitre 3 : Agencement de l'interface & Disposition des éléments

- L'encadrement (bordures)
 - But : regrouper les informations et focaliser le regard
 - Utiliser des lignes fines ou des couleurs pastel (éviter l'effet "annonce mortuaire")
 - Limiter les imbrications
- Les puces (bullets)
 - Mettre en évidence des lignes de texte (celle que vous lisez par exemple)
 - Attirer le regard sur des 'points d'entrée' visuels
 - Permettent une numérotation des points (facilitant le référencement)
- Le son
 - Moyen d'alerte extrêmement efficace (surtout si continu ou répétitif)
 - A réserver à des mises en évidence importantes (systèmes critiques)
 - Un texte parlé peut contenir davantage d'information
 - A utiliser avec parcimonie
- La proximité
 - Permet de mettre en évidence les relations (les liens) entre les éléments (libellé -champ associé par exemple)
 - Une mauvaise gestion peut conduire à des erreurs d'interprétation

3. Adaptation à la taille de l'écran

La disposition des éléments dans une interface doit prendre en compte la taille (et l'orientation) des écrans (fenêtres). Dans le domaine du web, l'adaptation aux différents terminaux (smartphones, tablettes, phablets) est devenue incontournable, compte tenu de la diversité des dispositifs utilisés.

On nomme conception responsive (responsive design) l'ensemble des techniques utilisées pour faire en sorte que l'interface s'adapte à la taille de l'écran ou de la fenêtre dont elle dispose pour s'afficher. Pour les applications autonomes (standalone) on a généralement recours à des conteneurs associés à des gestionnaires de disposition (layout-managers) qui se chargent d'adapter la taille et la position des composants en fonction de l'espace à disposition, tout en respectant certaines contraintes (règles de disposition).

Pour les sites et applications web, on adopte généralement des stratégies d'adaptation basées sur deux grands principes :

- Mise en page adaptative (adaptive layout)
- Mise en page responsive (responsive layout)

Dans une majorité d'interfaces, la disposition des éléments est basée sur une grille (un découpage du contenu en lignes et en colonnes) ou sur un assemblage de plusieurs grilles. La taille ainsi que le nombre de lignes et de colonnes de ces grilles pourront s'adapter en fonction de la taille de l'écran (ou de la fenêtre).

3.1 Mise en page adaptative

Dans les pages web on joue généralement sur la largeur et le nombre de colonnes des grilles. La longueur des pages s'adaptera en conséquence (avec un scrolling vertical). Avec une technique de mise en page adaptative, la largeur de la grille s'adapte par à coup. La largeur de chaque colonne est généralement fixe et le nombre de colonnes affichées change en fonction de certains points de rupture qui dépendent de la

Chapitre 3 : Agencement de l'interface & Disposition des éléments

taille de la fenêtre. Techniquement, on utilise généralement des media-queries (CSS 3) pour définir ces points de rupture (points de bascule) qui feront changer dynamiquement les feuilles de styles associées à la page affichée.

3.2 Mise en page responsive

La technique de mise en page responsive (responsive layout) est basée sur la notion de grille fluide (liquid layout) dans laquelle les colonnes sont définies en pourcentage de la largeur totale (ou autre unité relative à la taille de la grille). Ainsi les colonnes s'adaptent en continu à la taille de la grille.

Remarque : Pour éviter de placer le contenu dans des colonnes trop étroites (avec un contenu qui devient difficile à lire), on définit cependant des points de ruptures (comme pour la mise en page adaptative) qui permettent d'adapter malgré tout la disposition générale (le nombre et la disposition des colonnes par exemple) en fonction de la place à disposition.

4. Disposition dans les interfaces de type "formulaire"

Les applications classiques comportent souvent des interfaces de type "formulaire" qui sont composées de libellés, de champs texte, de listes déroulantes, de cases à cocher, de zones de texte, etc. Une des questions récurrentes à propos de ce type d'interfaces concerne la disposition des libellés qui sont associés aux éléments d'interaction.

Les modèles classiques de positionnement des libellés par rapport au champ sont : Au-dessus, A gauche, aligné à gauche, A gauche, aligné à droite, A l'intérieur du champ, Comme bulle d'aide (*tooltip*).

Chacune de ces dispositions présente des avantages et des inconvénients selon la situation (nombre de champs, taille de l'écran, longueur des libellés et des champs, etc.). En règle générale, on évitera les libellés à l'intérieur des champs et comme bulles d'aide (*tooltip*). Ces dispositions présentent, en moyenne, plus d'inconvénients que d'avantages. Sur les petits écrans, la meilleure disposition est souvent de placer le libellé au-dessus du champ (il limite le scrolling horizontal et reste visible lorsqu'on zoome). Si la longueur des libellés est pratiquement identique pour tous les champs, un placement au-dessus ou à gauche avec alignement à gauche est préférable. Si la longueur des libellés est très variable, un placement au-dessus ou à gauche avec alignement à droite est préférable.

Chapitre 4 – Composants de l'interface

Éléments d'interaction

1. Graphisme

La qualité graphique des interfaces joue un rôle important car elle contribue à mettre l'utilisateur dans une attitude positive (différentes études ont montré qu'une majorité d'utilisateurs y sont sensibles ; il ne faut donc pas négliger l'aspect émotionnel de l'interaction). La conception graphique des interfaces est souvent déléguée à des professionnels mais il est cependant important que les développeurs en comprennent les aspects principaux. Les éléments qui influencent les aspects graphiques sont nombreux, ils concernent par exemple :

- La base de conception (skeuomorphisme, flat design, pseudo-3D, ...)
- Le design des éléments graphiques (icônes, images, bordures, ...)
- Le choix et l'harmonie des couleurs (palette de couleurs, codage couleur)
- La typographie (polices de caractères, taille, style)
- La disposition des éléments (position, alignement, ...)

2. Les icônes

Les icônes sont des représentations symboliques plus ou moins abstraites (des graphismes) auxquelles on associe une signification dans l'interface utilisateur (c'est une notion très ancienne). L'utilisation d'icônes présente différents avantages :

- Les icônes sont indépendantes de la langue
- Sur une même surface, les icônes peuvent contenir plus d'informations que du texte ("une image vaut mille mots")
- Les icônes sont interprétées plus rapidement que du texte (pas de lecture, reconnaissance globale)

Lors du choix (ou de la conception) des icônes, il faut être attentif à la connotation culturelle associée à certains graphismes (afin d'assurer la compréhension auprès de tous les utilisateurs cibles)

2.1 Interprétation des icônes















L'utilisation d'icônes peut également présenter des inconvénients, notamment sur le plan de l'apprentissage du logiciel. Le principal problème rencontré concerne la compréhension de l'icône qui est parfois source d'ambiguïté.

Différentes enquêtes montrent que les utilisateurs préfèrent les interfaces comportant des icônes. Elles leur semblent plus faciles à utiliser, plus conviviales et plus ludiques (attitude positive). Cependant, plusieurs expériences ont prouvé qu'avec les icônes, les erreurs sont plus nombreuses qu'avec l'utilisation de commandes identifiées par des textes (notamment durant la phase d'apprentissage et chez les utilisateurs occasionnels), il est recommandé de combiner les icônes avec texte simple ou des info-bulles.

Chapitre 4 — Composants de l'interface -Éléments d'interaction

2.2 Conception des icônes

La conception des icônes peut être basée sur différents principes (qui peuvent être combinés) :

Règle	Description	Exemple
Ressemblance	L'icône représente une image de l'objet ou de l'action	 
Description Exemple	Décrit l'effet de l'action ou l'objet avant et après l'action	  
Analogie	Représente un lien sémantique avec l'objet (fonctionnement semblable, métaphore)	   
Symbolique	Utilisation d'un symbole usuel, connu dans un contexte donné (pictogramme familier)	  
Arbitraire	Symbole arbitraire (doit être appris)	 

Plus les icônes sont descriptives, plus elles sont faciles à interpréter par l'utilisateur et l'inverse (plus le graphisme est abstrait, plus l'utilisateur a de la peine à interpréter l'icône {il est contraint à apprendre et à mémoriser la signification}). Les icônes qui réussissent à représenter les objets concernés ainsi que les actions sont celles qui sont le plus facilement interprétables.

2.2.1 Quelques Recommandations

- Assurer une cohérence graphique dans les icônes (même style)
- Privilégier les représentations traditionnelles des objets usuels (téléphone, horloge, ...)
 - Pictogrammes familiers même s'ils sont éloignés des objets du monde réel.
- Utiliser des icônes standards
- Ajouter des bulles d'aides aux icônes non triviales
- Limiter le nombre d'icônes (les réserver aux commandes fréquemment utilisées)
- Concevoir les icônes pour des résolutions limitées (16x16 ... 128x128)
 - Ne pas réduire la taille d'images en haute résolution (photo)
- Valider la compréhension des icônes lors de tests utilisateurs
- Composer les icônes en combinant des sous-éléments connus

3. Les Couleurs

3.1 Utilisation de la couleur

La couleur constitue un élément important dans la signalétique des interfaces. Elle permet notamment :

- de mettre en évidence des objets
- d'informer sur l'état courant d'un élément
- d'établir des liens sémantiques entre éléments
- de contribuer à l'esthétique de l'interface
- ...

Dans l'utilisation des couleurs, il faut prendre en compte un élément important : tout le monde ne perçoit pas les couleurs de la même manière. Le daltonisme (dyschromatopsie) touche une proportion non négligeable de personnes (env. 8% des hommes et 0.5% des femmes).

Chapitre 4 — Composants de l'interface -Éléments d'interaction

3.2 Codage de la couleur

L'idée du codage couleur est d'associer à chaque couleur une signification concrète et uniforme pour l'ensemble des interfaces du système.

Limiter le nombre de couleurs pour favoriser leur discrimination : 4 au maximum dans un display et 7 ± 2 sur l'ensemble de l'interface. Les règles de codage doivent se baser sur un principe directeur (peu importe lequel, l'important est d'en avoir un), par exemple :

- Codage à la norme en vigueur dans le contexte applicatif
- Distance sémantique :
 - Même type d'information → même couleur
 - Types d'information similaires/proches → couleurs peu contrastées
 - Types d'information différents → couleurs contrastées

Dans certains domaines, certaines professions, il existe des codages couleur plus ou moins explicites qu'il faut respecter (parfois, des normes sont impératives).

3.2.1 Quelques Recommandations

- Concevoir en premier l'interface sans couleurs la couleur ne doit être qu'un plus
- Classifier les types d'objets/données à afficher : analyse sémantique
- Regrouper ces types par niveaux d'importance ... donc de visibilité
- Choisir les composantes couleurs pour chaque niveau d'importance :
 - teinte : sémantique de l'objet
 - saturation et luminance : visibilité
- Prototypage

4. Typographie

Les parties textuelles des interfaces constituent souvent des éléments importants de la communication avec l'utilisateur. C'est notamment vrai pour les sites web, les bornes interactives, etc.

4.1 Police de caractères

Une police de caractères (ou fonte ; angl. font) rassemble les lettres minuscules, les lettres majuscules, les chiffres ainsi que les signes de ponctuation et les ligatures (combinaison de caractères comme le 'œ') d'un même style dans un corps (taille) et une graisse donnée.

Le choix des polices de caractères ainsi que le respect des règles typographiques jouent donc un rôle important dans la conception de ce type d'interfaces. La typographie est une science ancienne, avec une très longue tradition et beaucoup de subtilités qui ont un impact sur la qualité graphique et la lisibilité des textes.

Une grande partie de la terminologie qui est encore utilisée actuellement, date de l'époque de la composition au plomb.

Il existe deux grandes familles : polices *avec sérifs* (ex. Garamond, Times New Roman) ou *sans-sérifs* (ex. Arial. Calibri)

4.2 Recommandations typographiques

- Écrire les textes en minuscules (majuscule au début des phrases). Lecture 13% plus rapide qu'un texte entièrement en majuscules.
- Sur écran, éviter les polices avec sérifs, les italiques et les polices 'manuscrites' en petites tailles (tenir compte de la résolution de l'écran).
- Minimiser le nombre de polices utilisées (pas plus de trois).
- Comme pour les couleurs, donner un rôle précis à chaque police (paragraphe standard, titre, légende, ...).
- Utiliser quelques tailles de police distinguables (éviter de varier la taille de la police simplement pour placer plus de texte).
- Placer de préférence quelques lignes longues plutôt que plusieurs lignes courtes de manière à minimiser les mouvements oculaires.
 - Env. 45..65 caractères par ligne, 30..35 en double colonne
- Éviter de souligner de longues portions de texte (év. un mot pour le mettre en évidence) car cela dégrade la lisibilité.
- Utiliser des polices droites, sans sérif, par exemple Arial, Helvetica ou Verdana.
- Aérer le texte en créant des paragraphes
 - Env. 1 espace toutes les 5 lignes (ordre de grandeur)
 - La norme ISO 9241-3 définit certaines contraintes à respecter pour garantir une bonne lisibilité à l'écran
- Favoriser la lisibilité en assurant un contraste suffisant
 - Utiliser de préférence des lettres sombres sur un fond clair
- Prendre en compte la diversité des tailles d'écran et des résolutions lors de la définition de la taille des polices de caractères
 - Choisir les bonnes unités (si c'est possible) en fonction du contexte et des types de périphériques potentiellement utilisés
 - ▶ unités absolues/relatives
 - ▶ tailles minimales/maximales.

Chapitre 5 – IHM pour les interfaces mobiles

1. Introduction

Les critères ergonomiques permettent d'orienter les choix de conception vers des pistes ergonomiquement solides, sans pour autant sacrifier l'aspect mercatique ou esthétique d'un site. Ils contribuent ainsi à éviter les pièges de la subjectivité et des goûts personnels, en offrant un cadre de travail neutre et efficace. Les interfaces mobiles et tactiles ont des particularités qui les différencient des interfaces classiques de la navigation web sur ordinateur. On va présenter ici quelles sont ces spécificités, ainsi que les critères d'ergonomie à respecter pour la conception d'interfaces mobiles et tactiles, illustrés par des exemples de sites mobiles ou d'applications.

2. Spécificités des mobiles

2.1 Des écrans réduits

Le premier point, évident, est que les smartphones et tablettes tactiles disposent d'écrans de taille réduite par rapport aux écrans d'ordinateurs. La lecture y est donc par défaut plus difficile sur ces appareils.

2.2 Pas de survol

Une autre singularité est l'absence de dispositif de pointage : le survol des éléments d'une interface mobile n'existe pas. Or c'est une action à laquelle nous sommes fortement habitués lors de la navigation web sur ordinateur. Il s'agira donc d'assurer la bonne affordance des éléments de l'interface mobile lors de la conception.

2.3 Le doigt et la main

Le dispositif de pointage des mobiles et tablettes tactiles est le doigt. Ce dispositif étant plus grossier que le curseur de souris d'ordinateur, de nouvelles modalités de sélection apparaissent. Par exemple la taille des éléments interactifs doit être repensée pour le doigt humain, et la distance "pouce ↔ élément" cliquable doit être juste. Le "pincer" d'écran pour zoomer (et plus généralement le "multi-touch") est devenu presque naturel pour les utilisateurs d'appareils tactiles.

La gestion de la main est à prendre en compte lors de la conception d'interfaces mobiles. En effet la main masque une partie de l'écran lors de l'utilisation d'un appareil tactile. Le placement des éléments de l'interface mobile sera donc différent de celui sur ordinateur (on privilégiera le centre par exemple)

2.4 Les comportements utilisateur

Enfin le comportement des utilisateurs de mobiles est différent de celui des utilisateurs d'ordinateurs. L'écran mobile se voit consacrer moins de temps que l'écran d'ordinateur. Une application ou un site web mobile qui fonctionne mal cessera donc tout de suite d'être utilisée. En moyenne, un mois après l'achat ou le téléchargement seulement 15% des utilisateurs continuent à utiliser l'application. 6 mois après, ce taux descend à 5%. D'où l'importance de la mise à jour des contenus. Par définition, l'environnement extérieur à l'appareil "mobile" est constamment changeant, et cet environnement peut à tout moment perturber

l'utilisation de l'appareil mobile. La logique de navigation sur une interface mobile doit donc être intuitive et solliciter le moins possible la mémoire de l'utilisateur.

3. Critères d'ergonomie des mobiles

3.1 Clarté, lisibilité et information

3.1.1 Police et couleurs :

Pour faciliter la lecture sur écran mobile :

- Utiliser des polices sans empattement (exemple : Arial) : Les polices à empattement ne sont pas à bannir, mais relèvent plus d'un choix esthétique (pour les titres par exemple).
- Utiliser entre 5 à 7 couleurs différentes : au-delà le lecteur aura des difficultés à comprendre rapidement l'interface proposée.
- Privilégier une page avec un fond clair et une typo foncée pour un contraste optimal.
- Utiliser un arrière-plan uni, plutôt qu'une image de fond, pour ne pas compliquer la lecture.

3.1.2 Codes et dénomination

Parler le langage utilisateur, en employant des termes clairs, et primordial. Il faut éviter toute ambivalence, qui mènerait à de mauvaises interprétations et à des erreurs. Les pictogrammes et les icônes valent souvent mieux que du texte, mais eux aussi doivent être sans ambiguïté, c'est-à-dire lisibles en terme de taille et d'interprétation. Plus généralement, toute image qui ne transmet pas d'information doit être évitée afin de ne pas surcharger une page

3.1.3 Gestion du doigt

Les interfaces tactiles requièrent de nouvelles modalités de sélection. La taille des boutons, des pictogrammes et des textes cliquables doit être adaptée au doigt humain. Les lois de la proximité et de la similarité s'appliquent aux interfaces mobiles. La loi de Fitts également -Concernant la distance entre différents objets : sur smartphone tout semble proche, cependant la jauge de distance est le pouce- Lors de la conception d'interfaces mobiles, il faut donc se poser la question "est-ce que le pouce accède à toutes les informations ?". Concernant la taille des objets : le Fittsizing se voit très souvent appliqué. Il s'agit du principe de compensation de la distance pour aller à un point (ou un objet) par l'augmentation de sa taille. Pour faciliter encore la bonne accessibilité des objets, on privilégiera un placement au centre de l'écran.

3.1.4 Feedback

Informé en permanence l'utilisateur sur l'état de l'application. Les temps de chargement pouvant être longs sur mobile, il faut faire comprendre à l'utilisateur que l'application n'est pas bloquée. Cela peut passer par l'usage de jauge de chargement ou de pictogrammes animés, accompagnés de texte (court et simple). Toujours dans le souci d'informer l'utilisateur, les services de base du smartphone doivent être intégrés. La barre de statut du téléphone devrait être présente constamment. En effet, bien qu'une application plein-écran soit plus immersive, les utilisateurs s'attendent cependant à voir l'état de leur batterie, ou les notifications qui apparaissent dans cette barre. La caché force à quitter l'application pour accéder à ces informations, ce qui est une perte de temps. Les interruptions possibles (SMS, appels, etc.) doivent également être gérées : l'application doit enregistrer son état avant l'interruption et permettre de reprendre là où elle s'est arrêtée.

3.2 Accélérateurs d'interaction

- Historique d'interactions : Divers types d'éléments font office "d'accélérateurs" d'une application mobile. L'historique d'interaction et les listes d'éléments favoris en font partie :
 - Une application qui conserve l'historique d'interaction d'un utilisateur avec elle en facilitera et en accélérera l'usage. Cela peut passer par garder en mémoire les recherches récentes ou les plus fréquentes pour les mettre en avant lors d'une nouvelle recherche. Plus généralement cela permet d'éviter à l'utilisateur de saisir plusieurs fois les mêmes informations.
 - les listes d'éléments favoris permettent d'épargner à l'utilisateur la répétition d'une action plus ou moins longue et pénible.
- Faciliter l'écriture des données : les "accélérateurs" permettent d'économiser le temps des utilisateurs de mobiles, comme les services d'auto-complétion (affichage de suggestions au fur et à mesure de la saisie), et le pré-remplissage des formulaires. Ces outils permettent notamment de réduire les erreurs de frappes, fréquentes avec les claviers virtuels des smartphones ou des tablettes tactiles.

3.3 Faciliter la navigation

3.3.1 Boutons et onglets

Sur le web, menus et autres fils d'Ariane aident l'utilisateur dans sa navigation. Avec leur écrans réduits, il est plus compliqué (mais pas impossible) d'intégrer ce type de repères visuels aux smartphones et tablettes. Les boutons de navigation ("précédent", "suivant") interviennent à ce niveau-là. Logiquement, leur position à l'écran doit correspondre à leur fonction : le bouton "précédent" sera placé à gauche, et le bouton "suivant" à droite. Les systèmes d'onglets permanents (en haut ou en bas d'écran) commencent à se standardiser et constituent des repères facilement identifiables. Attention cependant à ne pas proposer trop d'onglets différents (4 à 5 au maximum) pour ne pas réduire leur lisibilité.

3.3.2 Limiter l'effort de l'utilisateur

Il s'agit de concevoir un parcours de navigation qui épargne la mémoire de l'utilisateur. Ce dernier ne doit pas avoir besoin de se rappeler de ce qu'il vient de faire pour comprendre à quel endroit de l'application il se trouve et ce qu'il doit y faire. Il faut lui indiquer où il arrivera s'il tapote sur tel objet. A ce titre, l'intitulé des boutons de navigation peut justement être travaillé pour ne pas trop solliciter sa mémoire. Par exemple : Plutôt qu'un libellé "Retour" (retour vers quoi ?), il est préférable d'indiquer le nom de la partie de l'application vers laquelle l'utilisateur va revenir (exemple : "Accueil"). L'application ne devra pas proposer plus de 3 directions de navigations différentes sur une même page (en plus des 4 à 5 onglets permanents vus au point précédent).

3.3.3 Satisfaction

Tous les critères précédents participent évidemment à la satisfaction de l'utilisateur. On peut cependant faire particulièrement attention aux points suivants, afin d'assurer une expérience positive lors de l'utilisation de l'application ou du site mobile conçu :

- Supprimer les éventuels bogues et vérifier le bon déroulement de la navigation : application ne doit pas se fermer brutalement, ni contenir de page "cul-de-sac" (qui ne proposent pas de choix de navigation supplémentaires).
- Le lancement de l'application ou le chargement du site est-il suffisamment rapide ?
- L'application devrait être le moins possible gourmande en ressources, de façon à ne pas vider la batterie de l'appareil trop rapidement (à ce titre les services de géolocalisation doivent pouvoir être désactivés dans les options de l'application).

Chapitre 6 – Accessibilité

1. Définition de l'accessibilité

L'accessibilité numérique : « L'accessibilité numérique consiste en la mise à la disposition de tous les individus, quel que soit leur matériel ou logiciel, leur infrastructure réseau, leur langue maternelle, leur culture, leur localisation géographique, ou leurs aptitudes physiques ou mentales, des ressources numériques. » Wikipédia

L'accessibilité du Web : selon Tim Berner-Lee : « Mettre le Web et ses services à la disposition de tous les individus, quel que soit leur matériel ou logiciel, leur infrastructure réseau, leur langue maternelle, leur culture, leur localisation géographique, ou leurs aptitudes physiques ou mentales. »

L'accessibilité du web est la problématique de l'accès aux services et contenus en ligne non seulement pour les personnes handicapées et les seniors, mais aussi de manière plus générale pour tous les utilisateurs qui ne disposent pas du confort offert par un ordinateur de bureau situé dans une pièce tranquille. En effet, son application concerne également les utilisateurs non handicapés placés dans des situations moins confortables comme avec un téléphone mobile, une tablette... ou placés en situation particulière de bruit, de dimension d'affichage, etc.

2. But de l'accessibilité

L'accessibilité vise à assurer que le logiciel est utilisable par des personnes qui ont des déficiences :

- Visuels : personnes malvoyantes, aveugle, daltonien et mauvaise vision
- Auditives : personnes malentendantes ou sourds
- Motrices : impossibilité/difficulté de bouger un membre, tremblement
- Cognitives : difficulté de compréhension, dyslexie

3. Pourquoi l'accessibilité

Beaucoup de personnes sont concernés, vraiment beaucoup...

De par leur situation :

- Déficiência visuel : soleil / ombre
- Auditives : silence / bruit ambiant / absence de casque / obligation de silence
- Motrices : besoins d'une main pour faire autre chose
- Cognitives : pas de connaissance de la langue, pas de connaissance du domaine, mauvaise mémoire.

Et encore... Dans une période de leur vie.

- Déficiência visuel : vieux / jeunes
- Auditives : vieux / jeunes
- Motrices : vieux / jeunes / enfant
- Cognitives : vieux / jeunes / enfant

Chapitre 6 : Accessibilité

Donc tout le monde, Nous sommes tous, à un moment ou à un autre, en situation d' handicap !

4. Solutions

Le système essaye de prendre en compte au maximum ces problèmes : ajustement automatique de la luminosité, lecteur d'écran, taille par défaut de la police, prévention du rebond...

Solutions aux Problèmes visuels

- Utilisation sans pointeurs (clavier uniquement)
- Pas de texte en image pour qu'il puisse être lue Par la machine ou affiché en braille
- Grande taille de police, loupes d'écran
- Ne pas transmettre de l'information uniquement avec les couleurs ou les différences de police (utiliser les narrateurs de lecture d'écran destinés aux aveugles)

Solutions aux Problèmes moteur

- Saisie audio
- Pointage avec un dispositif qui observe la pupille
- Prédiction des mots, tolérance aux fautes
- Pas de touches simultanées

Solutions aux Problèmes cognitifs

- langue de l'utilisateur
- culture de l'utilisateur
- Simplicité
- Cohérence
- Redondance

Solutions aux Problème de dyslexie

- Éviter les fonds blanc
- Ne pas justifier le texte
- Faire des phrases courtes
- Éviter l'italique
- ...

Bonne nouvelle

- Le système d'exploitation gère déjà beaucoup de choses.
- Ne combattez pas le système !
- Utilisez les préférences du système (taille, couleurs)
- N'entravez pas le bon fonctionnement du système

5. Conclusion

Il y a pas de recommandation spéciales, Appliquez les règles d'ergonomies vues précédemment. Nous sommes tous en situation d'handicap. Il faut y faire attention ! Ce qui est bon pour les cas extrêmes est également bon pour les autres situations.