

Modèles pour l'Interaction Homme-Machine

Tronc commun RICM3
2006-2007

Renaud Blanch
IIHM - CLIPS-IMAG - UJF
mailto:renaud.blanch@imag.fr
<http://iihm.imag.fr/blanch>

Plan du cours

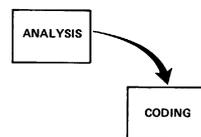


Figure 1. Implementation steps to deliver a small computer program for internal operations.

1. Psychologie cognitive
2. Méthodes de conception
3. Architectures logicielles

2. Méthodes de conception

- 2.0 Introduction
- 2.1 Modèle de l'utilisateur
- 2.2 Modèle de tâche
- 2.3 Du modèle de tâche à l'IHM abstraite
- 2.4 De l'IHM abstraite à l'IHM concrète

2.0 Introduction

L'objectif de l'IHM est de réaliser des systèmes :

- **utiles** (adaptés aux besoins) ; et
- **utilisables** (adaptés aux utilisateurs).

2.0 Introduction

En 1990, Mitch Kapor disait dans son *Software Design Manifesto* :

"Despite the enormous outward success of personal computers, the daily experience of using computers far too often is still fraught with difficulty, pain, and barriers for most people, which means that the revolution, measured by its original goals, has not as yet succeeded.

*There is a **conspiracy of silence** on this issue. It's not splashed all over the front pages of the industry trade press, but we all know it's true. Users are largely silent about this. There is no uproar, no outrage. But scratch the surface and you'll find that **people are embarrassed to say they find these things hard to use**. They think the fault is their own. So users learn a bare minimum to get by. **They under-use the products** we work so hard to make and so don't help themselves or us as much as we would like. They're afraid to try anything else. In sum, everyone I know (including me) feels the urge to throw that infuriating machine through the window at least once a week. (And now, thanks to recent advances in miniaturization, this is now possible.)*

.../...

2.0 Introduction

.../...

*The lack of usability of software and poor design of programs is the secret shame of the industry. By training and inclination people who develop programs haven't been oriented to design issues. This is not to fault the vital work of programmers. It is simply to say that **the perspective and skills which are critical to good design are typically absent from the development process**, or, if present, exist only in an underground fashion.*

We need to take a fresh look at the entire process of creating software - what I call the software design viewpoint. A rethinking of the fundamentals of the process of making software.

2.0 Introduction

Quelques **problèmes** courants :

- trop ou pas assez de fonctions (utilité) ; ou
- existence de fonctions inaccessibles (utilisabilité).

On peut citer :

- des problèmes de **séquencement** de l'interaction ;
- des problèmes **syntaxiques** ou **articulatoires** ;
- des **retours inappropriés** ou **inexistants** ...

2.0 Introduction

Quelques **causes** à ces problèmes :

- l'ignorance des principes issus des théories de la cognition ;
- l'ignorance du contexte d'utilisation ;
- une mauvaise compréhension des tâches de l'utilisateur cible ;
- une mauvaise intégration des préoccupations humaines dans le processus de développement du génie logiciel et notamment à l'étape d'expression des requis.

Le problème est

le processus de développement !

2.0 Introduction

Il faut identifier :

- la **raison d'être** du système ;
- un **modèle de l'utilisateur** qui caractérise les compétences de l'utilisateur cible ;
- un **modèle des tâches** qui spécifie l'enchaînement des activités ;
- une **IHM abstraite** qui spécifie la structure du système en espaces de travail ;
- une **IHM concrète** qui spécifie les techniques d'interaction ;

puis enfin :

- l'**IHM finale** livrée à l'utilisateur.

2.0 Introduction

La raison d'être du système

- doit résumer en quelques phrases **la motivation principale** pour laquelle le travail va être entrepris ;
- doit détailler **les faiblesses** de l'existant ;
- doit fixer **des objectifs** (si possibles mesurables).

C'est ce qu'il ne faut **jamais perdre de vue** tout au long du cycle de conception.

2.1 Modèle de l'utilisateur

Le rôle du modèle est d'identifier les caractéristiques des **utilisateurs typiques** et de guider les **choix de conception**.

Il a pour but de **réduire les distances d'exécution et d'interprétation** et d'identifier les **besoins sémantiques et syntaxiques**.

Pour l'établir, il faut connaître ses utilisateurs :
"Early and continual focus on the user." [Schneiderman, 1987]
"Talking to users is not a luxury, it s a necessity." [Gould, 1988]

L'accès aux utilisateurs et le choix d'utilisateurs représentatifs est rarement simple.

2.1 Modèle de l'utilisateur

Le modèle de l'utilisateur comporte des **caractéristiques générales** :

- biométriques (taille, âge, sexe, déficiences ...) ; et
- sociales (formation, usages culturels ...)

et des **caractéristiques liées au produit** :

- connaissances dans le domaine (concepts et tâches) ;
- connaissances en informatique (générales, des outils connexes ...)

2.1 Modèle de l'utilisateur

Le modèle de Rasmussen permet de classer les **niveaux d'expertise**.

S'il existe **plusieurs niveaux d'expertise** parmi les utilisateurs cibles, **les hiérarchiser** pour **permettre les choix** en cas d'incompatibilités, ou pour **identifier des utilisateurs types** pour les scénarios et les tests d'utilisabilité.

2.1 Modèle de l'utilisateur

Il existe **différentes techniques** pour établir le modèle de l'utilisateur :

- le **sondage** par questionnaire ;
- les **entretiens** ;
- l'**observation in situ** ;
- le **think aloud** ;
- le **magicien d'Oz** ;
- ...

Quand l'utilisateur n'est pas accessible, il faut se contenter de classifications générales (novice, expert, ...) et de scénarios.

2.1 Modèle de l'utilisateur

Pour les **sondages**, il faut disposer d'**utilisateurs représentatifs**.

Les questions doivent être **précises**, les sondés ne sont pas concepteurs.

Demander de rapporter des **incidents critiques**.

Poser une **question ouverte** :
"Quelles sont vos suggestions ?"

2.1 Modèle de l'utilisateur

Pour les **entretiens**,
il faut qu'un **modérateur** anime la session et fixe les règles
à un petit groupe de concepteurs et d'utilisateurs.

Il n'y a pas de **mauvaise idée**.

Les sessions doivent être **enregistrées**, éventuellement filmées.

2.2 Modèle de tâche

La connaissance de l'utilisateur permet d'analyser son **activité**
(ce qu'il fait indépendamment du système).

Faire rédiger des **scénarios** par des utilisateurs représentatifs
(ou à défaut soi-même) permet d'identifier ces activités.

Il faut ensuite les organiser en un **modèle de tâches**.

2.2 Modèle de tâche

Une **tâche** consiste en :

- un **but** (état souhaité) ; et
- une **procédure** pour atteindre ce but.

Une **procédure** est un ensemble de sous-tâches liées par :

- des relations de **composition** ; et
- des relations **temporelles**.

Une **tâche élémentaire** est

une tâche décomposable en **actions physiques**.

Une **action physique** est

une opération sur un dispositif d'entrée/sortie qui provoque un
changement d'état du dispositif (clic, mouvement, affichage, etc.)

2.2 Modèle de tâche

Les modèles de tâche sont des **structures arborescentes** dont **les nœuds sont les buts** et **les sous-arbres sont les procédures** pour atteindre ces buts.

Les nœuds peuvent être **décorés** par :

- les **concepts du domaines** (objets référencés, i.e. paramètres, variables globales ...)
- les **préconditions** et les **postconditions** ;
- la **fréquence** ;
- la **complexité** ;
- la **criticité** (niveau de danger, caractère irrévocable) ;
- les **contraintes temporelles** (durée maximale) ;
- l'**acteur** responsable de l'exécution de la tâche (utilisateur et/ou système) ;
- toute autre information pertinente (selon le domaine).

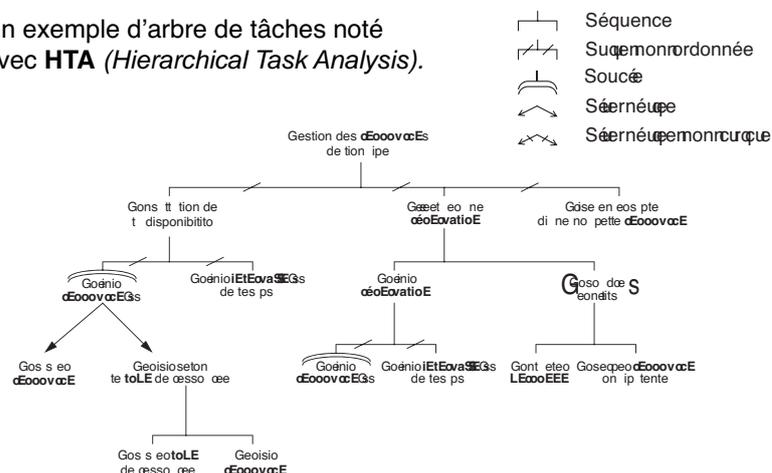
2.2 Modèle de tâche

Il existe de nombreuses notations pour les modèles de tâche, par exemple **CTT**, **HTA**, **MAD** ...

Les *use cases* et les diagrammes de séquences d'UML peuvent aussi être utilisés.

2.2 Modèle de tâche

Un exemple d'arbre de tâches noté avec **HTA (Hierarchical Task Analysis)**.



2.2 Modèle de tâche

Il n'y a pas de notation standard,

CTT (*Concurrent Task Tree*) propose les opérateurs suivants :

- T1 >> T2 : séquence ;
- T1 [data]>> T2 : séquence & transmission d'information ;
- T1 [] T2 : disjonction ;
- T* : fermeture (répétition de zéro à autant de fois qu'on veut) ;
- Tⁿ : répétition ;
- [T] : option ;
- T1 [> T2 : interruption définitive ;
- T1 |> T2 : interruption avec reprise ;
- T1 || T2 : parallélisme.

Il existe un éditeur qui permet l'édition de modèles de tâche : **CTTE**.
<http://giovane.cnuce.cnr.it/ctte.html>

2.2 Modèle de tâche

Les *use case models* d'UML spécifient les requis fonctionnels du point de vue des acteurs.

Use Case Name

- 1 Brief Description
- 2 Actors*
- 3 Flows of Events
 - 3.1 Main (basic) Flow
 - 3.1.1 Step 1
 - 3.1.2 Step 2
 - 3.1.3 Step ...
 - 3.2 Alternative Flows
 - 3.2.1 Alternative flow 1
 - 3.2.1.1 Step 1
 - 3.2.1.2 Step 2
 - 3.2.1.3 Step ...
 - 3.2.2 Alternative flow 2
 - 3.2.3 Alternative flow ...
- 4 Special Requirements
 - 4.1 Usability requirements
 - 4.2 Business rules
 - 4.3 Other non-functional requirements...
- 5 Pre-conditions
- 6 Post-conditions
- 7 Extension Points

2.2 Modèle de tâche

3. Flow of Events

3.1 Basic Flow

- 3.1.1 BID**
This use case starts when the Buyer bids on a displayed item currently available for auction.
- 3.1.2 ENTER AMOUNT**
The Buyer enters the bid amount. The system validates the bid amount. The entered bid must be greater than the current (i.e., greatest) bid by an amount greater than the minimum bid increment specified for the auction.
- 3.1.3 BUYER CONFIRMS BID**
The system provides information that tells the legal obligations of placing a bid. The Buyer confirms that the bid should be placed.
- 3.1.4 POST BID**
The system posts the bid for the auction. The entered bid becomes the current (i.e., greatest) bid.
- 3.1.5 SEND EMAIL**
The system sends an email confirmation to the Buyer, including the bid amount for the auction item, as well as when the auction will close.
- 3.1.6 SYSTEM CONFIRM BID**
The system notifies the Buyer that the bid has been accepted and displays the Buyer's name and email address, as well as the item name and the bid amount. The use case ends.

3.2 Alternative Flows

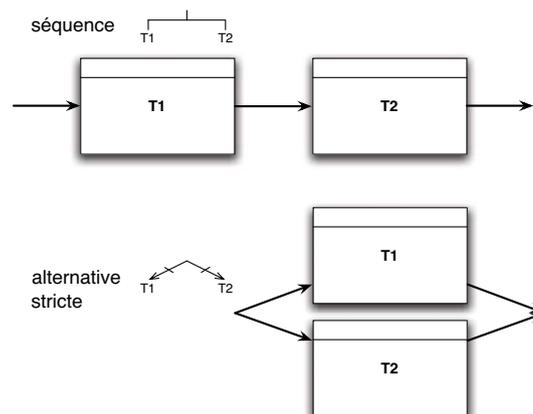
- 3.2.1 AUCTION IS CLOSED**
At BF BID, the auction for the item has been closed, a message is displayed to the Buyer stating that the auction has been closed, and the bid is not accepted. The use case ends.
- 3.2.2 BUYER HAS PENDING PAYMENTS**
At BF BID, the Buyer has pending payments, the system displays a message to the Buyer, reminding him/her that he/she has pending payments that are due and that until those payments are made, the Buyer cannot participate in an auction, either as a Buyer or a Seller. The use case ends.
- 3.2.3 ENTERED BID IS INVALID**
At BF ENTER AMOUNT, the User enters an invalid bid, the system indicates to the User the reason the bid is invalid (e.g., the entered bid was not greater than the greatest bid by an amount greater than the minimum bid increment specified for the auction). Resume at BF ENTER AMOUNT.
- 3.2.4 BID NOT CONFIRMED**
At BF CONFIRM BID, the Buyer does not confirm that the bid should be placed. The use case resumes at BF ENTER AMOUNT.

2.3 Du modèle de tâche à l'IHM abstraite

L'arbre des tâches est segmenté en **espaces de travail**.

Les relations entre les tâches permettent de déduire les **enchaînements** entre les espaces de travail.

2.3 Du modèle de tâche à l'IHM abstraite



2.3 Du modèle de tâche à l'IHM abstraite

Il faut vérifier la **conformité des enchaînements** avec le **modèle de tâche** (distance sémantique).

Il faut **annoter** les espaces de travail avec les **concepts du domaine** pertinents.

On peut **regrouper** les espaces de travail pour **raccourcir** les trajectoires d'interactions.

Il faut vérifier que les **tâches fréquentes** sont **accessibles rapidement**.

On peut (doit) ajouter des **services d'utilité publique** comme le couper/coller, l'*undo/redo*, des valeurs par défaut pertinentes, de l'aide ...

2.4 De l'IHM abstraite à l'IHM concrète

Il faut garder en tête que le but est de **réduire les distances d'exécution et d'évaluation.**

Pour cela il faut :

- vérifier la **conformité à l'IHM abstraite** ;
- assurer la visibilité des concepts pertinents ;
- mettre en évidence les changements d'état ;
- regrouper les informations ayant un lien logique ;

2.4 De l'IHM abstraite à l'IHM concrète

Il faut garder en tête **les principes théoriques de la première partie.**

Le modèle du **processeur humain** précise que la mémoire à court terme est petite, donne des indications sur comment organiser les menus ...

Le modèle ***Interacting Cognitive Subsystems*** donne des indications pour organiser les écrans ...

Les modèles ***GOMS*** et ***Keystroke*** permettent de quantifier des trajectoires d'interaction et de faire des choix.

Critères ergonomiques

Les **critères ergonomiques** [Scapin et Bastien, 1993] fournissent une "grille de lecture" des interfaces.

Ces critères sont regroupés en 8 catégories :

- guidage ;
- charge de travail ;
- contrôle explicite ;
- adaptabilité ;
- gestion des erreurs ;
- homogénéité / cohérence ;
- signifiante des codes et dénominations ;
- compatibilité

Critères ergonomiques

1. guidage

“ [...] ensemble de moyens mis en œuvre pour [...] conduire l'utilisateur lors de ses interactions [...] ”

“ Un bon guidage facilite l'apprentissage et l'utilisation du système [...] ”

Critères ergonomiques

1.1 incitation

“ [...] recouvre les moyens mis en œuvre pour amener les utilisateurs à effectuer des actions spécifiques [...] ”

1.2 groupement / distinction entre items

“ [...] prend en compte la **localisation** et le **format** afin d'illustrer les relations entre les items affichés [...] ”

1.3 feedback immédiat

“ [...] une réponse aussi immédiate que possible doit être fournie à l'utilisateur [...] ”

1.4 lisibilité

“ [...] concerne les caractéristiques [...] de présentation [...] pouvant entraver ou faciliter la lecture [...] ”

Critères ergonomiques

2. charge de travail

“ [...] ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs [...] ”

“ Plus la charge de travail est élevée, plus grands sont les risques d'erreurs [...] ”

Critères ergonomiques

2.1 brièveté

“ [...] limiter autant que possible le travail de lecture, d’entrée, et les étapes par lesquelles doivent passer les utilisateurs. ”

2.1.1 concision

2.1.2 actions minimales

2.2 densité informationnelle

“ [...] la charge de travail [...] pour des ensembles d’éléments [...] ”

Critères ergonomiques

3. contrôle explicite

“ [...] prise en compte par le système des actions explicites de l'utilisateur [...] ”

“ Quand les entrées des utilisateurs sont explicitement définies par eux-mêmes, [...] les ambiguïtés et les erreurs sont limitées.”

Critères ergonomiques

3.1 actions explicites

“ [...] le système doit exécuter seulement les opérations demandées par l'utilisateur [...] et ce, au moment où il les demande.”

3.2 contrôle de l'utilisateur

“ [...] l'utilisateur doit toujours avoir la main [...] ”

Critères ergonomiques

4. adaptabilité

“ [...] capacité à réagir selon le contexte [...] ”

“ [...] une interface ne peut convenir à la fois à tous ses utilisateurs potentiels [...] ”

4.1 flexibilité

“ [...] capacité de l'interface à s'adapter à des actions variées de utilisateurs.”

4.2 prise en compte de l'expérience de l'utilisateur

“ [...] respecter le niveau d'expérience de l'utilisateur.”

Critères ergonomiques

5. gestion des erreurs

“ [...] moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent. [...] ”

“ Les interruptions provoquées par les erreurs ont des conséquences négatives sur l'activité des utilisateurs.”

Critères ergonomiques

5.1 protection contre les erreurs

“ [...] détecter et prévenir les erreurs d'entrée de données ou de commandes [...] ”

5.2 qualité des messages d'erreur

“ [...] concerne la pertinence, la facilité de lecture et l'exactitude de l'information [...] sur la nature des erreurs commises [...] ”

5.3 correction des erreurs

“ [...] moyens mis à la disposition des utilisateurs pour leur permettre de corriger leurs erreurs.”

Critères ergonomiques

6. homogénéité / cohérence

“ [...] façon dont les choix de conception de l'interface [...] sont conservés pour des contextes identiques [...] ”

“ [...] Le manque d'homogénéité peut augmenter considérablement le temps de recherche [...] ”

Critères ergonomiques

7. signifiante des codes et dénominations

“ [...] adéquation entre l'objet ou l'information affichée ou entrée et son référent [...] ”

“ Lorsque le codage est signifiant, le rappel et la reconnaissance sont meilleurs [...] ”

Critères ergonomiques

8. compatibilité

“ [...] accord [...] entre les caractéristiques des utilisateurs [...] et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue [...] d'autre part [...] ”

“ [...] Les performances sont meilleures lorsque l'information est présentée sous une forme directement utilisable. ”