

conception participative

génération de solutions
conception détaillée

stephane.conversy@enac.fr - bureau C107



1

1. ___
2. ___
3. ___

conception participative: plan général

1. intro et phase de définition du problème (3h)
 - interview in situ (tp)
 - scénario de travail (travail personnel)
2. phase de génération d'idées et de conception détaillée: paradigmes d'interaction (2h)
 - identification google map et sncf.com (tp)
- 3. phase de génération d'idées et de conception détaillée (2h cours+TP)**
 - **brainstorming oral + brainstorming video (tp),**
 - **proto papier et video, scénario de conception (personnel)**
4. phase d'évaluation: design walkthrough et tests (3h)
5. pratique en entreprise (intuilab+seditec 2 x 1h30)

2

conception participative: moyens

- Définition du problème
 - Moyens: Interview In Situ
 - Résultats: diagrammes, scénarios de travail
- Génération de solutions
 - Moyens: brainstorming oral et video (BO et BV)
 - Résultats: un ensemble d'idées de solution à explorer
- Conception détaillée
 - Moyens: Ateliers de conception, construction de prototypes, keystroke, interaction instrumentale, dimensions cognitives
 - Résultats: prototypes papier, video, interactif, scénario de conception, story board
- Evaluation
 - Moyens: Design Walk-through et Test d'utilisabilité
 - Résultats: avis et faits sur l'adéquation et l'efficacité du système

3



génération de solutions: objectifs pédagogiques

savoir **citer, décrire, justifier** et **mettre en oeuvre** cette phase et les techniques **brainstorming oral** et **brainstorming video**

les **mettre en oeuvre** durant vos chef d'oeuvre/stages

évaluation en tant que membre du jury de CO et stage:
utilisation, place dans le processus, correction, complétude

4

1. phase de génération d'idées

- cours + video
- brainstorming oral + brainstorming video (tp),

2. phase de conception détaillée

- cours sur les méthodes de design
- prototype papier/video + scénario de conception (personnel)

5

justification de la phase génération de solutions

problèmes:

- trop souvent, on développe tout de suite une solution « évidente »
- la phase d'exploration de solutions est inexistante
 - ...elle se fait souvent devant la machine quand on réalise, l'informaticien fait des choix pour tout le monde...
- parfois plusieurs personnes ont des solutions en tête, comment les confronter ?

but de la phase de génération de solutions: trouver **plusieurs** solutions aux problèmes soulevés durant la phase de « définition du problème »

il faut « élargir l'espace de design »...

moyen: *brainstorming oral* et *brainstorming video*

6

brainstorming oral

but du BO: s'appuyer sur la **créativité collective**, afin de produire de **nombreuses solutions alternatives** ou complémentaires

principe: séance de 20-30 min avec plusieurs personnes autour d'une table.
un modérateur **présente les problèmes** à résoudre

tout le monde **réfléchit à voix haute** (pas en même temps !) sur des **solutions** à ces problèmes, avec l'aide éventuelle de rapides croquis et gestes

on **écoute** les autres et on **rebondit** sur leurs idées

il faut que ça **fuse** ! de la **créativité** !

(video example)

7

justification et règles du BO

- rappel: la CP doit être efficace (rapport qualité/coût)
 - qualité du BO = **nombre, diversité/originalité, ciblant** le(s) problème(s), **construction à plusieurs** « par rebond »
 - coût du BO = **temps, mobilisation des participants**
- donc:
 - (nombre) il faut s'**affranchir des contraintes techniques du brainstorming** lui-même: **oral et papier-crayon**, pas de système d'enregistrement complexe
 - (diversité/originalité) il faut s'**affranchir contraintes implicites** du système à définir, qu'elles soient **techniques** ou organisationnelles
 - (originalité) il faut s'**affranchir de la peur d'être ridicule**: chacun prépare une idée « stupide » et doit la dire pendant la séance
 - (nombre, originalité) il ne faut **pas d'évaluation**, donc pas de critiques ni de discussions sur les mérites d'une solution : ça prend trop de temps et ce n'est pas le but (vient plus tard en phase de conception détaillée...)
 - (diversité/originalité) ne pas avoir peur de mal comprendre les autres, **libre interprétation**
 - (cibler, qualité, « rebond ») il faut **explicitier**: papier/crayon et gestes (préparation au BV)

8

BO: organisation

- qui
 - des concepteurs (de discipline différente), des utilisateurs, donneur d'ordre (si utilisateur)
 - anticipez les problèmes relationnels (ne convoquez pas d'ennemis déclarés, identifiez les personnes à « personnalité » pour être attentif lors de la modération)
- préparation
 - en amont: sélectionnez les problèmes à résoudre
 - réserver une salle adéquate
 - autour d'une table, places proches
 - pas de salle en U, tout le monde doit être concerné et participer
 - préparer une mallette à brainstorming, du papier A3 + crayon pour chacun, un paper board sur lequel le scribe notera les idées
- convocation:
 - donner le thème et les problèmes à résoudre lors de la séance, et dire que ça durera 1h (tenir les promesses, donc préparer !)
 - joindre les scénarios de travail, et demander leur **lecture**
 - « pour l'intérêt et l'efficacité de la séance de travail, il est très important que vous lisiez les scénarios de travail ci-joints... »

9

BO: déroulement

- phase 0 (début):
 - vérifier la position des gens, et que tout le monde est concerné
 - désigner un modérateur qui vérifie le temps, modère les critiques, gère la digression
 - désigner un scribe, qui notera les idées sur un paper board
 - rappeler le thème, s'assurer que tout le monde a lu les ST
 - commencez
- phase 1: générer une quantité d'idées
 - enregistrer toutes les idées, y compris les idées « stupides »
 - le scribe s'assure auprès de l'émetteur de l'idée que sa transcription est fidèle
 - modérer: soyez attentif...
 - aux prises de parole continues, à la timidité (« schtroumpf timide, tu voulais dire quelque chose... »), aux problèmes d'autorité...
 - aux critiques: ne pas évaluer les idées
 - aux digressions: recadrer subtilement sur les problèmes spécifiques

10

- on ne peut pas tout approfondir (rapport qualité/coût)
- phase 2 **optionnelle**: classer en fonction de la qualité si nécessaire
 - chacun choisit ses idées préférées
 - classer les idées par nombre de votes
 - ne pas oublier les idées insolites

11

justification du brainstorming video

- pb: il faut pouvoir **utiliser** les résultats de la phase de génération de solutions
 - il faut donc les mettre sous forme de « rapport » pour se rappeler
 - il faut communiquer aux autres
 - mais il faut s'affranchir de toutes les barrières à la créativité...
 - ... et on va jeter plein de trucs de toute façon !
- but: produire des solutions **compréhensibles**, donc **explicites**
- solutions possibles:
 - textes du BO: pas suffisant (pas assez de détails, description incomplète, statique)
 - outil graphique+animation type flash
 - trop coûteux à mettre en œuvre
 - filtre de l'expert
 - ne permet pas de tout exprimer
 - **Brainstorming Video (BV)**

12

brainstorming video: justification et principes

principe: reprendre les idées émises pendant le BO, et les expliciter avec des *croquis* (ou *MockUp* ou *Sketch*) papier des interfaces + une video des interactions + des commentaires audio:

- dynamique (**interaction**, pas *interface*)
- explicite
- facile à mettre en œuvre
- facile à récupérer
 - avec appareil photo: 15 fps, à transformer en mp4 plus tard

ceci n'est pas du prototypage



il s'agit toujours d'un brainstorming:

- pas d'évaluation ni de critique
- il faut rebondir et proposer des alternatives
- préparer/faire chaque video à deux, pour vérifier l'explicitation...
- il faut faire **vite et sale** (« *quick and dirty* »)
 - inutile de faire de **beaux dessins**
 - stylo **épais**: il faut que ce soit visible sur la video
 - **pas de montage sophistiqué** et **pas plus de 2 min**: découper en plusieurs videos
 - premier écran avec une identification de l'idée: facilite la manipulation des videos

13

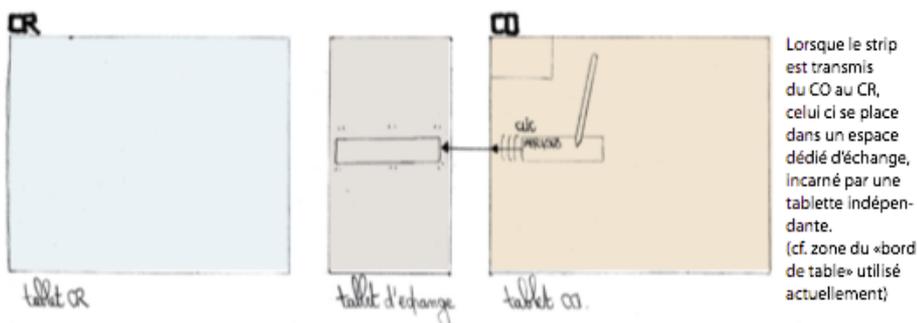
résultats de la phase de génération d'idées

pour la consultation par les membres du projet:

- site wiki avec video+résumé textuel adéquat, **online** si possible (à la *youtube*)

pour les rapports écrits:

- *storyboard*: bande dessinée avec étapes importantes de l'interaction...
 - temps -> espace: consultation plus efficace



14

phase de génération de solutions: synthèse

- but: **trouver plusieurs solutions** aux problèmes soulevés durant la phase de « définition du problème »
- moyens: à partir des ST, brainstorming oral (génération) et video (génération+explicitation)
- piège pour vous (concepteurs): phase de la CP qui paraît la plus facile à faire
 - ne pas s'arrêter à ça: la CP **ne se résume pas** au brainstorming
 - « *j'ai appliqué une méthode de conception participative, car j'ai fait la **définition du problème** avec une **séance de brainstorming entre mon maître de stage et moi** »: le dernier qui a dit ça sèche dans le grenier du bâtiment O, celui que personne ne trouve...*
 - **bien le préparer et bien le gérer** pour bien produire...

15

1. ____
2. ____
3. ____

plan du jour

1. phase de génération d'idées

- cours + video
- brainstorming oral + brainstorming video (tp),

2. phase de conception détaillée

- **prototype papier/video + scénario de conception**
- cours sur les méthodes de design

16

phase de conception détaillée: moyens

- prototype, scénarios de conception
- le prototype seul ne suffit pas !
 - on ne fabrique pas un prototype sans penser à son utilisation
 - revenir au scénarios de travail pour savoir quelle direction prendre, et faire des protos qui servent à quelque chose !
- scénario de conception:
 - même forme/structure qu'un scénario de travail
 - illustration de l'utilisation du prototype

17

17

Scénario de conception

But

- Créer une description réaliste de l'utilisation du nouveau système
- Aider les utilisateurs à imaginer l'usage de l'interface, afin qu'ils choisissent en connaissance de cause et en fonction du contexte

Procédure

- Prendre un scénario de travail existant
- Utiliser les idées générées pendant le brainstorming
- Modifier le scénario de travail pour inclure le nouveau système en cours de conception
- Faire un storyboard, avec texte et images

18

But: faire des choix de conception en simulant l'interaction avec le système avec des prototypes

Pour qui ?

- Les concepteurs
- Les utilisateurs
- La direction

Faire des choix:

- Pas trop tôt: considérer différentes alternatives de conception
- S'assurer de l'utilisabilité dans différentes conditions
- Se concentrer sur les parties problématiques de l'interface
 - Un système peut être bon en théorie mais mauvais en pratique à cause de détails, même petits
- Utiliser des méthodes descriptives, prédictives et génératives

19

Prototypes Papiers

Paper Prototyping, Carolyn Snyder, Morgan-Kaufman

Le plus rapide à mettre en œuvre...

... donc le plus facilement jetable

- Explorer, explorer, explorer

objectif des premières phases de prototypage:

expliciter la compréhension et l'adéquation des solutions aux problèmes

susciter les échanges et la confrontation des solutions

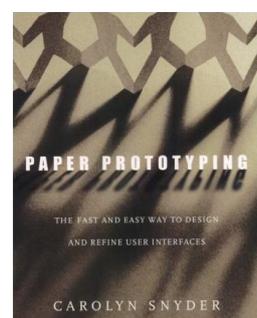
pas de valider des choix de concepteur !

pour les utilisateurs: compréhensible, modifiable, manipulable

permet la construction à plusieurs: meilleure confrontation et partage de la connaissance du prototype

Interviennent à plusieurs étapes :

- Design
- Tests
 - Design walkthrough, test d'utilisabilité



20

But

- Produire un prototype testable d'une nouvelle interface
- Produire un document pour archive et démonstration

Procédure

- Utiliser le scénario de conception
- Revoir la liste des idées du brainstorming
- Créer les « props »
 - matériaux : papier, marqueurs, post-its, transparents
- Exécuter un scénario et essayer des variantes

21

Exemple: Xerox Star

400 pages de spécification de l'interface avant la moindre ligne de code

Comprenait tous les écrans possibles

Prototypage tout le long du développement de la spécification

Test

- « *Accurately time-stamped videos that superimposed a view of the screen, a view of the user, and a view of the user's hands.* »
- « *Verplank and his crew did 600 or 700 hours of video, looking at every single feature and function. From all these video recordings, we were able to identify and eliminate many problems.* »

1. phase de génération d'idées

- cours + video
- brainstorming oral + brainstorming video (tp),

2. phase de conception détaillée

- prototype papier/video + scénario de conception
- **cours sur les méthodes de design**

23



conception détaillée: objectifs pédagogiques

savoir **citer, décrire, justifier** et **mettre en oeuvre** cette phase et les techniques de **prototypage papier** et de **scénario de conception**

savoir citer et décrire les trois types de méthode (**descriptives, prédictives, génératives**), pour guider la conception

savoir citer, décrire et mettre en oeuvre les principes de **réification, polymorphisme, réutilisation**, la notion de **modèle conceptuel**, la notion de **dimensions cognitives** (hidden dependencies, premature commitment, secondary notation, viscosité), et **keystroke**

savoir dire **quand** ces méthodes et techniques interviennent dans le processus, et les **mettre en oeuvre** durant vos chef d'oeuvre/stages

être capable de **justifier** les choix de conception

24

1. intro
2. méthodes descriptives
 1. quelques méthodes possibles
 2. cognitive dimensions of notation
3. méthodes prédictives
 1. propriétés des instruments
 2. keystroke-level model
4. méthodes génératives
 1. réification, polymorphisme, reuse
 2. modèle conceptuel

25

phase de conception détaillée

- but: construire de manière **itérative** et **réfléchi**e des prototypes (papier ou logiciel)
- finalité (entre autres [cf cours de yannick jestin sur le prototypage](#)):
 - expliciter les solutions, pour faire ressortir les vrais problèmes, et donc évaluer la justesse de l'identification des problèmes
 - évaluer *a priori* l'intérêt des solutions
 - expliciter les solutions, pour faire ressortir les « détails » importants
 - être évalué par les utilisateurs lors d'un *design walkthrough* ou d'un *test d'utilisabilité*

26

- séance de travail seul-e ou à plusieurs comprenant:
 - réflexions sur ce qu'il faut faire
 - est-on sûr de résoudre un problème réel identifié ?
 - quel est le but du prototype ? [cours de yannick jestin sur le prototypage](#)
 - **génération et sélection de solutions**
 - **résout-on le problème avec de bonnes solutions ?**
 - construction de prototype (ça prend du temps !)

27

génération et sélection de solutions

- génération = créativité ?
 - il faut rester ouvert et créatif, mais la créativité n'est pas une excuse à l'ignorance (dixit Philippe Palanque)
 - il faut pouvoir **argumenter et justifier**
- utiliser des méthodes **descriptives, prédictive, génératives**
- méthode descriptive
 - décrire les phénomènes, en les identifiant s'ils sont connus (et pouvoir en parler avec les autres), réutiliser les connaissances, analyser les causes d'un problème (mais pas forcément les prédire cf gestalt illusions visuelles)
- méthode prédictive
 - avec des *modèles*, évaluer a priori l'intérêt (ou *prédire* l'efficacité) d'une solution par rapport à une autre
- méthode générative
 - utiliser des moyens de réflexion pour générer des solutions

28

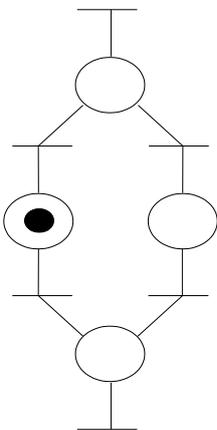
- identifier, reconnaître, réutiliser, déconstruire...
 - un paradigme d'interaction
 - un type de système connu
 - interface builder, window manager, collecticiel, un processus de workflow, un langage de programmation (textuel ou visuel)
 - le dernier qui n'a pas reconnu un interface builder sèche etc.*
 - un problème avec des solutions connues
 - par ex de visualisation (cours de visualisation)
 - une interaction déjà vue, et qui marche
 - login, base de données, etc...
 - *cognitive dimensions of notation*

29

une méthode descriptive: *cognitive dimensions of notation*

thomas green, 1989

- ou plutôt, *cognitive dimensions of interaction...*
 - représentation (notation) + possibilités d'interaction



```
void main() {
    while(1) {
        ....
    }
}
```



30

- pour répondre à ces questions:
 - est-ce que les activités prévues sont bien instrumentées par le système ?
 - si non, expliciter le problème en utilisant des phénomènes déjà identifiés (*cognitive dimensions*)
- repose sur un ensemble d'**abstraction des activités** d'un utilisateur de système interactif

31

abstraction des activités

- **incrementation**: ajouter des fiches dans un fichier; des formules dans un tableur; des instructions dans un programme
- **transcription**: recopier un extrait de livre; convertir une formule mathématique en formule dans le tableur; ou en code dans un programme
- **modification**: changer les mots de l'index d'un catalogue; changer l'organisation spatiale d'une feuille de tableur; modifier un algorithme pour tenir compte de nouvelles contraintes
- **exploratory design**: faire des croquis; concevoir une fonte typographique. On n'a pas le résultat final en tête, il faut le découvrir...
- **searching**: rechercher une cible connue; par exemple les endroits où une fonction est appelée dans un programme
- **exploratory understanding**: découvrir une structure ou un algorithme; la base d'une classification

32

dimensions cognitive: dimensions d'analyse des systèmes

- Abstraction gradient
- Closeness of mapping
- Consistency
- Diffuseness / terseness
- Error-proneness
- Hard mental operations
- **Hidden dependencies**
- Juxtaposability
- **Premature commitment**
- Progressive evaluation
- Role-expressiveness
- **Secondary notation and escape from formalism**
- **Viscosity**
- Visibility

33

hidden dependencies

- les dépendances entre entités sont-elles visibles ? si oui le sont-elles dans les deux directions ? est-ce qu'un changement à un endroit conduit à un effet de bord inattendu ?
- lien direct ou indirect: une entité utilise une autre entité qui à son tour utilise une autre entité. Si la 3^e entité change, quelles sont les répercussions sur la première ?
- exemple:
 - les cellules dans un tableur
 - les styles dans un traitement de texte
 - une hiérarchie de classe complexe

34

- existe-t-il des contraintes sur l'ordre des actions à effectuer ? est-ce que des décisions doivent être prises alors qu'on ne possède pas encore toutes les informations nécessaires pour les prendre ? ces décisions peuvent-elles être annulées ou corrigées plus tard ?
- exemple:
 - être obligé de déclarer une variable avant de l'utiliser
 - être obligé de donner des noms ou des renseignements par l'intermédiaire de boîte de dialogue à chaque création d'entité...
 - nombre de fps dans flash/director, ou taille des diapo dans powerpoint d'il y a longtemps...

35

secondary notation and escape from formalism

- informations supplémentaires apportées par d'autre moyen que la notation/représentation officielle
- exemple:
 - commentaires dans un programme
 - commentaires dans la marge d'un document

36

- résistance au changement, à la modification
- exemples:
 - passer tous les titres de section en majuscules: faut-il une action par titre de section ? (*repetition viscosity*)
 - insérer une nouvelle figure dans un document oblige à changer tous les numéros de figure suivant, plus leurs références dans le texte (*knock-on viscosity*: notation inconsistante)

37

Ex téléphone: Making a call



- 'Lift the phone' by pressing the top left radio button;
- Then press the number keys in the appropriate order.
- That's all you have to do.
- 'Hang up' by pressing the top right-hand radio button.

38

Ex téléphone: store and recall a number



- To store a number:
 - pick up the handset
 - press STORE
 - key in the telephone number you wish to store
 - press STORE again
 - press one of the keypad buttons (0 - 9) (depending on where you wish to store it)
 - the number is now stored
- To dial a stored number:
 - pick up the handset
 - press RECALL
 - press one of the keypad buttons for the number you want to dial

39

Ex téléphone: subdevice memory label

Digit	Who
0	<i>Aunt Mary</i>
1	<i>Barack</i>

- Carton pour noter les numéros à la main
- Comment modifier un label ?
- Comment réarranger l'ordre des labels ?

40

relation entre *activités* et *dimensions cognitives*

	<i>transcription</i>	<i>incrementation</i>	<i>modification</i>	<i>exploration</i>
viscosity	acceptable	acceptable	harmful	harmful
hidden dependencies	acceptable	acceptable	harmful	acceptable for small tasks
premature commitment	harmful	harmful	harmful	harmful
abstraction barrier	harmful	harmful	harmful	harmful
abstraction hunger	useful	useful (?)	useful	harmful
secondary notation	useful (?)	–	v. useful	v. harmful
visibility / juxtaposability	not vital	not vital	important	important

41

design manoeuvre: comment corriger les problèmes

AIM	MANOEUVRE	AT THIS COST
to reduce viscosity:	add abstractions (so that one 'power command' can change many instances)	increases need for lookahead (to get the right abstractions); raises the abstraction barrier; may also increase hidden dependencies among the abstractions
to improve comprehensibility:	allow secondary notation – let users choose where to place, things, how to use white space, what font and colour to use; allow commenting	increases viscosity (because layout, colour etc are not usually well catered for by the environment)
to make premature commitment less expensive:	reduce viscosity (so that users can easily correct their first guess)	see above, re viscosity
to remove need for lookahead:	remove internal dependencies in the notation; or allow users to choose an easier order to make decisions	may make notation diffuse, or increase errors allow free order needs a cleverer system
to improve visibility:	add abstractions (so that the notation becomes less diffuse)	see above re abstractions

42

Ex téléphone: subdevice memory label

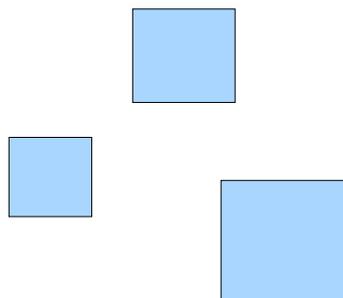
Digit	Who
0	<i>Aunt Mary</i>
1	<i>Barack</i>

- Carton pour noter les numéros à la main
- Comment modifier un label ?
- Comment réarranger l'ordre des labels ?

43

Exemple alignement d'objets graphiques

- Visqueux
 - Si je bouge une entité, il faut que refasse les commandes
 - Surtout avec des groupes (dégrouper, regrouper)
- Solution: « abstraction » → guide magnétique



44

méthodes descriptives: synthèse

- décrire (seulement...) un phénomène, ou un système
- permettent d'appliquer des connaissances connues et de communiquer en utilisant le même vocabulaire
- permettent d'identifier plus rapidement les problèmes réels
- comment s'en servir ?
 - d'abord des connaissances: mémoriser
 - les identifier pendant les phases de conception détaillée

45

1. ___
2. ___
3. ___

conception détaillée: plan

1. intro
2. méthodes descriptives
 1. quelques méthodes possibles
 2. cognitive dimensions of notation
- 3. méthodes prédictives**
 1. propriétés des instruments
 2. keystroke-level model
4. méthodes génératives
 1. réification, polymorphisme, reuse
 2. modèle conceptuel

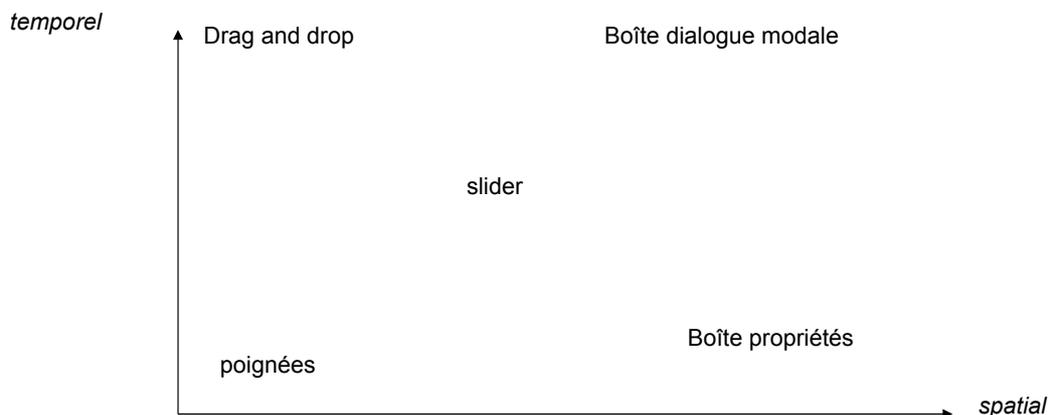
46

- méthode permettant de prédire des résultats (par exemple de performance), parfois en s'appuyant sur un *modèle*
- *modèle*: simplification de la réalité
- exemple:
 - propriété des instruments (interaction instrumentale):
 - degré d'indirection spatiale et temporelle
 - degré de compatibilité
 - degré d'intégration des dimensions
 - keystroke

47

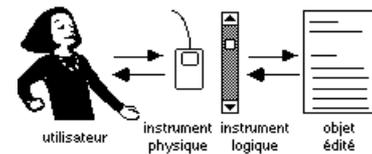
degré d'indirection spatiale et temporelle

- indirection spatiale: distance entre action et objet
- indirection temporelle: temps entre action de l'utilisateur et exécution sur l'objet

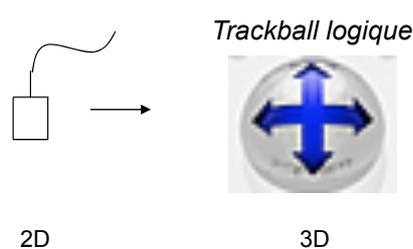
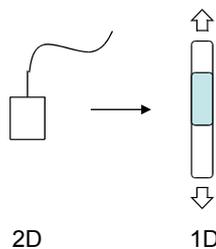


48

degré d'intégration des dimensions et degré de compatibilité



- Degré d'intégration des dimensions



- Degré de compatibilité
 - Scrollbar up = doc down
 - Souris pour déplacer un objet
- aparté: séparabilité/intégrabilité des dimensions
 - contrôler R,G,B avec une souris

49

keystroke-level model (KLM)



Décomposition en tâches élémentaires pour expliciter le nombre de tâche et prédire le temps d'exécution

Opérateurs :

- K = frappe (*keystroking*) 0,2s
- B = clic (*buttonpress* et *buttonrelease*) 0,1s
- H = rapatriement de la main (*homing*) 0,4s
- P = désignation (*pointing*) ~1,1s (devrait être loi de fits->simplification)
- D = dessin (*drawing*) $0,9n+0,16l$ (n segments de longueur l)
- M = activité mentale (*mental activity*) 1,35s
- R = temps de réponse du système (*response time*) $\max(0, r-u)$ r:réponse u:action utilisateur

Exemple: sélectionner avec un rectangle élastique avec main sur le clavier au début: HBDB

valeur des temps trouvée de façon expérimentale
 on considère que l'utilisateur sait ce qu'il a à faire, sans erreur
 prédictions: elles sont réalistes (ça marche assez bien)
permet de comparer des techniques d'interaction

50

Keystroke: règles pour l'opérateur M

Règles pour le placement de l'opérateur M :

R1 insérer M devant tous les K.

R2 insérer M devant un P sur un objet qui correspond à une commande et pas à un argument.

(ex de commande: outil "colorer" ; ex d'argument: forme graphique)

R3 supprimer M si l'opérateur qui suit peut être anticipé.

(ex: un pointage-clic PMK devient le plus souvent PK)

R4 si MKMK...MK constitue un mot connu (ex: "dir", "1978"), le transformer en MKK...K.

R5 si MKK...K MK constitue un mot connu suivi d'un terminateur comme « enter », supprimer le dernier M si le mot est une commande, et pas un argument.

R6 supprimer les M après un R

Le plus important : être cohérent !

51

convertisseur de francs-euros: solution 1

Choisir le type de conversion, taper le montant, puis sur Enter.

Convertir € vers F
 Convertir F vers €

→

Choisir le type: HPK

Taper le montant: HPK H KKKK K (montant 4 chiffres + enter)

Ajouter les M: HMPMK HMKMKMKMKMK

Enlever les M: HMPK HMKKKKMK = 7,15s

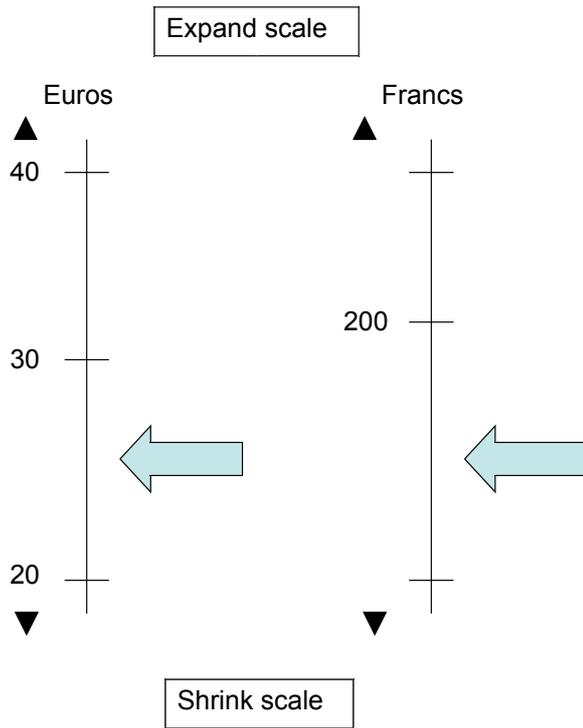
Sans le type: 3,7s



convertisseur de francs-euros: exemple 2

Cas optimum:
 HPK(p+c) PK(d+r)
 HMPMKMK
 HMPKK = 3,25s

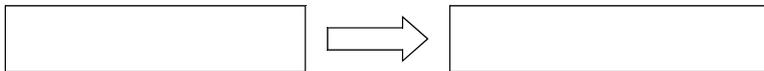
Cas le pire:
 16,8s



53

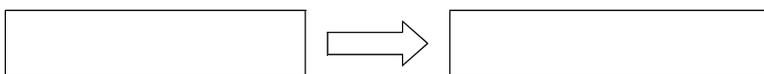
convertisseur de francs-euros: solution 3 et 4

Taper F ou €, puis le montant, puis sur Enter.



MK KKKK MK=3.9s

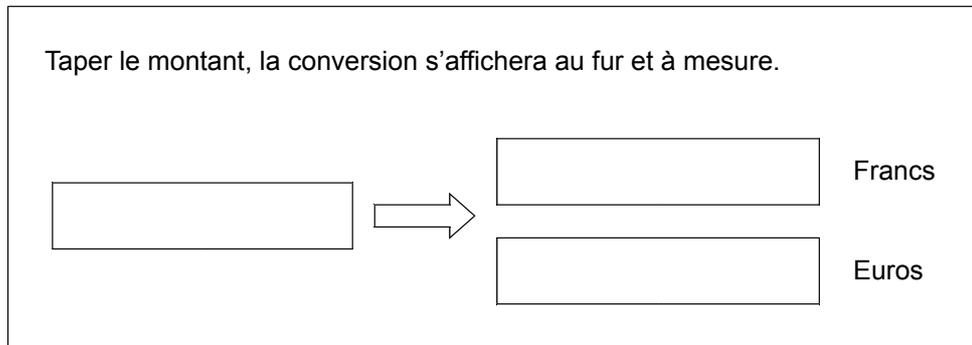
Taper le montant, puis sur F ou €.



MKMKMK MK=3.7s

54

convertisseur de francs-euros: exemple 5



MK KKK = 2,15s

100% efficacité en termes de caractères entrés

KLM: prédiction, et surtout **comparaison** pour faire des **choix**

1: 3,7s 2: 3,25s à 17s 3: 3,9s 4: 3,7s 5: 2,15s

55

méthodes prédictives: synthèse

- permet de prédire les performances (en temps, en nombre d'actions)
- les propriétés des instruments
- KLM permet de prédire et comparer des temps entre interactions différentes, et donc de faire des **choix argumentés**
 - et de comparer par rapport à l'existant ! et donc justifier que votre système est meilleur que celui d'avant...

56

1. intro
2. méthodes descriptives
 1. quelques méthodes possibles
 2. cognitive dimensions of notation
3. méthodes prédictives
 1. propriétés des instruments
 2. keystroke-level model
- 4. méthodes génératives**
 1. réification, polymorphisme, réutilisation
 2. modèle conceptuel

57

interaction instrumentale: principes de conception

- Réification: transforme un concept en objet directement manipulable
 - Étend la notion de ce qui constitue un objet d'intérêt
 - Ex: groupe, style
 - Meta instruments: tool palette
- Polymorphisme: étend la portée des commandes sur ces objets
 - Copy, paste, cut, undo/redo (sur tout, dont les groupes par ex)
 - Groupes homogènes ou hétérogènes
- Réutilisation: moyen de capturer et de réutiliser des motifs d'utilisations
 - Output / input
 - Redo, coller multiple

58

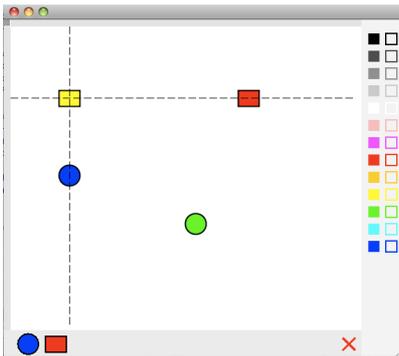
58

exemple: guides magnétiques et instrument de recherche

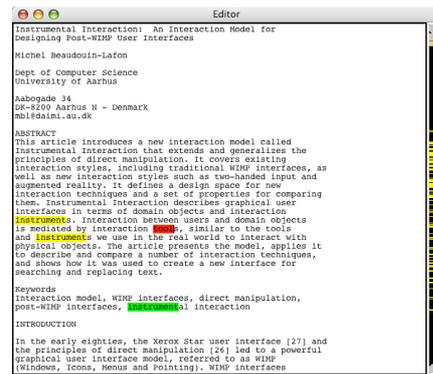
beaudouin-lafon, CHI 1999



rechercher: interaction traditionnelle



implémentation: sébastien maubert, M2IHM03



rechercher: interaction instrumentale

59

59

application des principes: guide magnétique

- guide magnétique:
 - réification de l'alignement
 - réutilisation en tant que groupe
 - déplacement commun
 - application de styles
 - polymorphisme
 - manip à la souris
 - application de styles à des objets individuels et à des groupes

60

60

application des principes: chercher et remplacer

- Pas de boutons *ok*, ou *chercher*: le texte est trouvé tout de suite
 - faible indirection temporelle
 - on voit immédiatement si on a fait une erreur de frappe
- Toutes les occurrences sont visibles:
 - pas de bouton *next*, *previous*: meilleure indirection spatiale pour la recherche
 - pour remplacer/ne pas remplacer on clique sur l'occurrence: faible indirection spatiale
 - scrollbar modifiée pour voir les occurrences
 - vitesse variable pour réduire la division de l'attention
- On peut continuer à taper du texte
 - pas de mode
-
- Réification ? Polymorphisme ? Reuse ??

61

61

modèle conceptuel



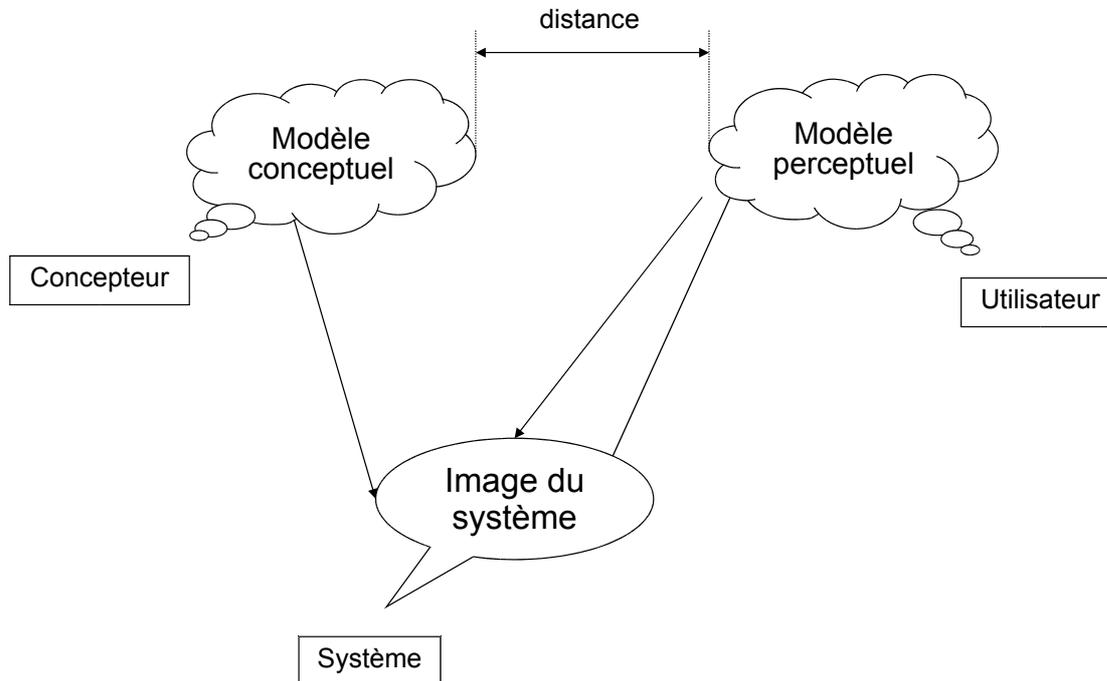
D. Norman

- Concept issu du « modèle de l'action »
- Le modèle de l'action modélise les processus psychologiques qui conduisent à un comportement
- Donne un guide de lecture permettant d'évaluer a priori les interfaces du point de vue perception/action/cognition

62

62

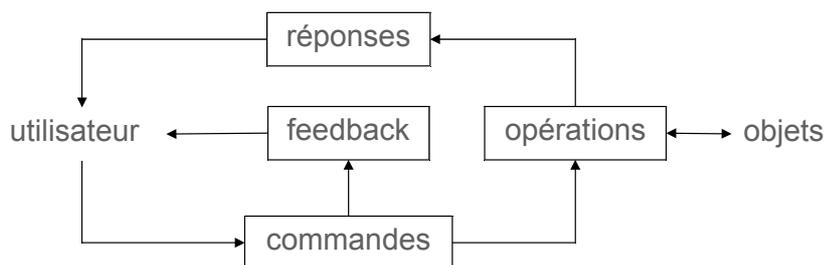
Modèles perceptuel et conceptuel



63

63

Structuration du modèle conceptuel



- Identifier les objets
Ce que l'utilisateur manipule
ex desktop: fichier, dossier
- Identifier les opérations sur les objets
Ce que l'utilisateur peut faire avec ces objets
ex desktop: mettre dans un dossier, détruire
- Identifier les commandes
Comment exécuter effectivement les opérations avec l'interface
ex desktop: glisser-déposer (drag'n'drop) dans un dossier, dans la corbeille

64

64

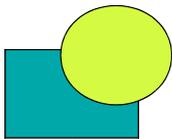
Objets	Représentation	Attributs	Opérations
Fichier	Icône (type) + nom	Chemin d'accès + type + nom + taille etc.	Créer, renommer, déplacer, détruire...

Opérations	Commandes	Feedback	Réponse
Détruire un fichier	Drag and drop dans la poubelle	Fantôme de l'icône	L'icône disparaît et la poubelle grossit
	Sélectionner + backspace	Icone en surbrillance	L'icône se déplace vers la poubelle et disparaît

65

65

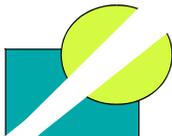
Modèle conceptuel : exemple dessin bitmap vs vectoriel



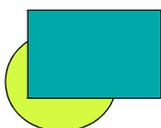
De quoi est fait ce dessin ?

Comment est fait ce dessin ?

Quelles sont les actions possibles ?



Photoshop: C'est un ensemble de points que l'on peut effacer

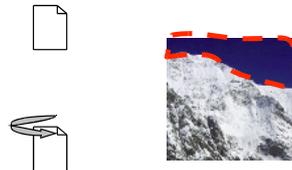


Illustrator: C'est un rectangle et un cercle que l'on peut déplacer

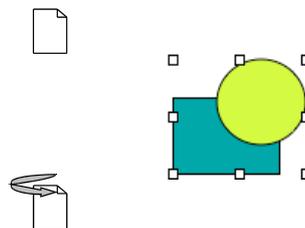
66

66

- Bitmap: images formées de pixels
 - Objets de base: ensemble de pixels (zones)
 - Opérations de base:
 - Définir une zone
 - Appliquer une opération aux pixels d'une zone



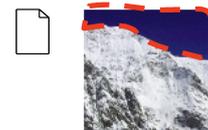
- Édition vectorielle: images formés de contours
 - Objets de base: objets vectoriels en 2D 1/2 (+ couche)
 - Opérations de base:
 - Modifier la géométrie d'un objet
 - Modifier les attributs d'un objet
 - Changer l'ordre de superposition



67

67

Édition Bitmap: objets

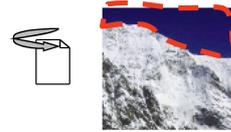


Objets	Représentation	Attributs	Opérations
Zone	Contour clignotant	Ensemble des pixels de la zone	Définir, modifier, remplir
Brosse	Forme du curseur	Forme Couleur Transparence	Peindre
Palette d'outils	Fenêtre flottante	Liste d'outils Outil sélectionné	Choisir un outil
Etc.			

68

68

Édition Bitmap: opérations

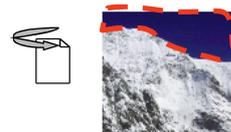


Opérations	Commandes	Feedback	Réponse
Sélectionner une zone	Clic Outil rectangle de sélection + clic & resize	Rectangle élastique	Contour clignotant
	Outil lasso (forme libre)+clic&draw	Tracé libre	
Peindre dans la zone sélectionnée	Clic outil pot de peinture + clic dans la zone sélectionnée	Curseur	Zone remplie par la couleur courante
Peindre avec une brosse	Clic outil brosse + clic&drag	Curseur	Changement de couleur des pixels sous la brosse

69

69

Édition Bitmap: opérations (2)

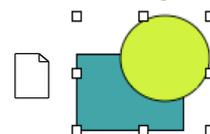


Opérations	Commandes	Feedback	Réponse
Modifier la zone sélectionnée	Commande « inverser » du menu « sélection »	??	Echange la zone sélectionnée avec le reste
	Commande « étendre » du menu « sélection »		Etend d'un pixel autour de la zone
Transformer les pixels de la zone sélectionnée	Sélection d'un filtre du menu « filtres »	Boîte de dialogue des paramètres du filtre	Application du filtre aux pixels de la zone sélectionnée
Etc.			

70

70

Édition vectorielle: objets

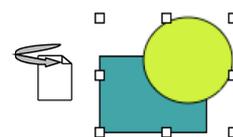


Objets	Représentation	Attributs	Opérations
Contours vectoriels	Forme graphique à l'écran	Géométrie Attributs graphiques	Créer, déplacer, redimensionner, modifier, détruire...
Inspecteur de propriétés	Fenêtre flottante	Liste d'attributs (couleur, transparence)	Changer la valeur d'un attribut
Palette d'outils	Fenêtre flottante	Liste d'outils Outil sélectionné	Choisir un outil
Etc.			

71

71

Édition vectorielle: opérations

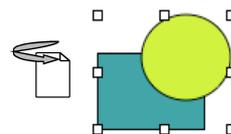


Opérations	Commandes	Feedback	Réponse
Créer un objet	Sélection d'outils dans la palette + clic&drag	Rectangle élastique ou forme fantôme	Nouvel objet au premier plan avec attributs courants + ajout de poignées
	Sélectionner la plume + séquence de clic&drag	Changement de curseur + ligne élastique	Nouveau contour au premier plan avec attributs courants + ajout de poignées

72

72

Édition vectorielle: opérations (2)

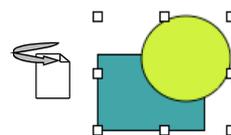


Opérations	Commandes	Feedback	Réponse
Sélectionner un ou des objets	Clic sur fond de l'objet (s'il en a un) ou tracé (s'il en a un)		Apparition de poignées de manipulation Valeurs des attributs s'affichent dans l'inspecteur
	Clic sur fond de l'écran + drag	Rectangle élastique	Apparition de poignées de manipulation Valeurs <i>communes</i> des attributs s'affichent dans l'inspecteur

73

73

Édition vectorielle: opérations (3)

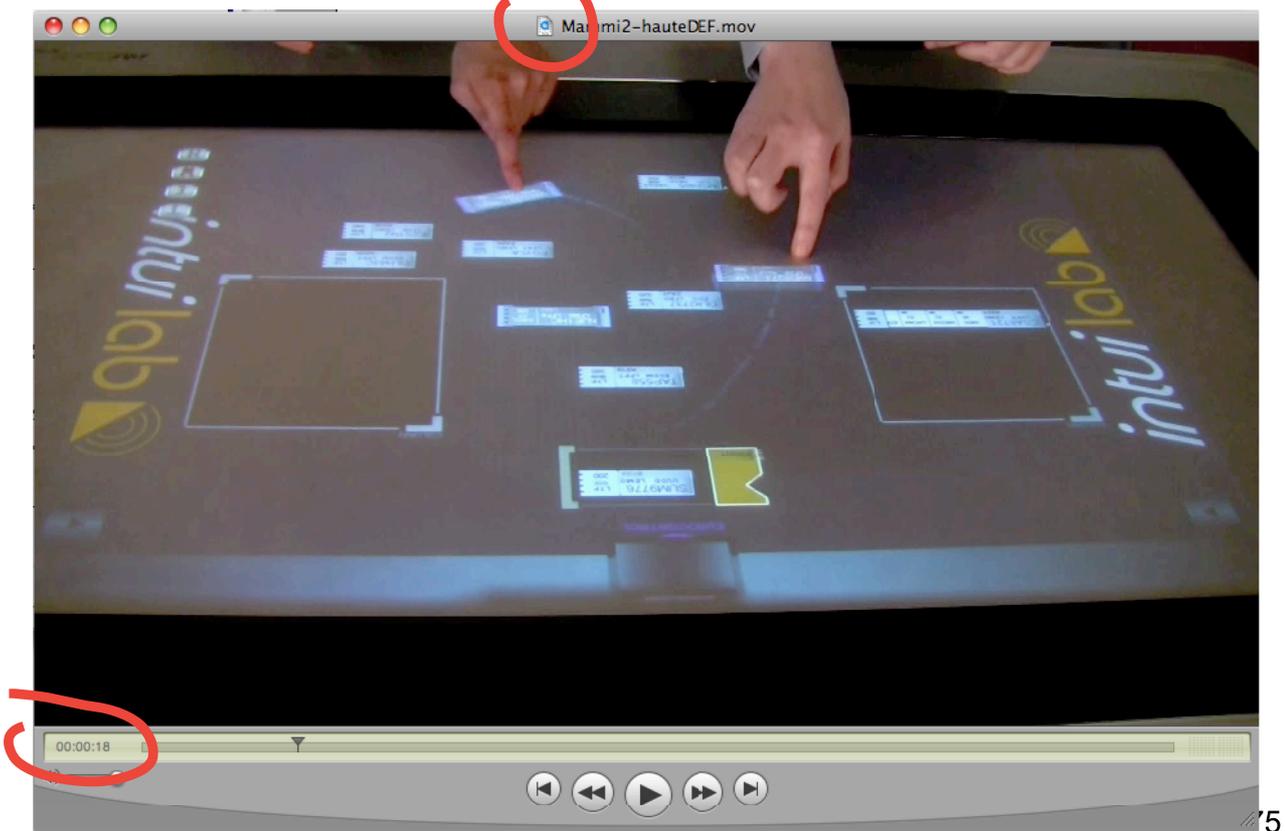


Opérations	Commandes	Feedback	Réponse
Modifier la géométrie de l'objet sélectionné	Clic sur une poignée & drag	La poignée suit le curseur	Géométrie modifiée
Modifier les attributs	Modification d'un attribut dans la liste de l'inspecteur		Application des nouvelles valeurs
Changer l'ordre de superposition	Commande « avancer » menu		Réaffichage en tenant compte de l'ordre

74

74

« tout est éditible/manipulable... »



Quelques règles

- Grouper les commandes par catégorie
 - Gestion de l'espace de travail
 - Édition globale
 - Édition locale
 - Etc.
- Vérifier la complétude
 - Mêmes opérations dans les 2 tableaux
 - Toute propriété doit être visible et éditible
- Vérifier la cohérence
 - Interactions similaires produisent les mêmes effets

Quelques règles

- Appliquer les principes de conception
- Réification
 - Identifier de nouveaux objets
 - Ex: Palette d'outils = objet
- Polymorphisme
 - Factoriser les commandes
 - Exemples:
 - quelles commandes existantes peuvent s'appliquer à la palette ?
 - qu'est-ce que ça veut dire de bouger ça, modifier ceci ? (icône barre de fenêtre du mac, temps quicktime)
- Réutilisation
 - ... des sorties: favoriser des commandes qui utilisent les objets existant

77

77

Un bon modèle conceptuel

- David Liddle, concepteur du Star: une bonne interface c'est
 - 1: un bon modèle conceptuel
 - 2: invocation de commandes consistantes entre « applications »
 - 3: l'affichage
- Ex Visicalc:
 - Affichage pauvre
 - Commandes pas terrible
 - Super modèle conceptuel
- Ex Desktop: n'a marché que sous forme de caricature

méthodes génératives: synthèse

- permettent d'appliquer des principes qui guident et génèrent des solutions durant la conception de système
- réification, polymorphisme et réutilisation permette de générer/créer des interactions **cohérentes**
- la définition d'un bon modèle conceptuel (objets, opérations et commandes) permet de réduire les risques d'incompréhension de la part des utilisateurs et de réfléchir à des interactions cohérentes

79

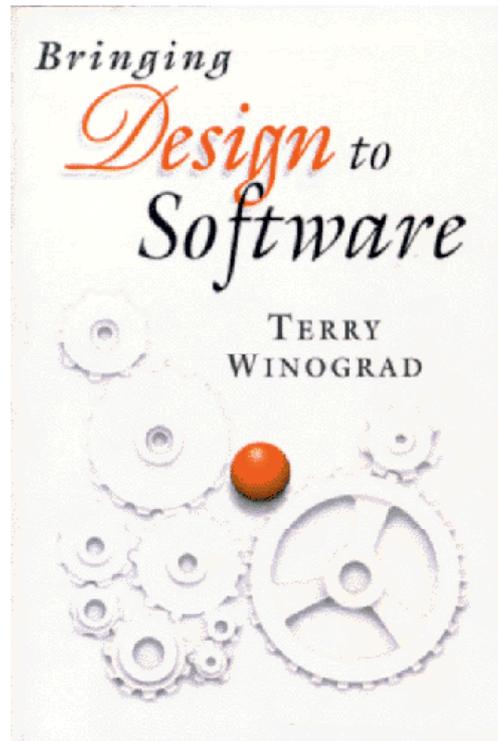
79

phase de conception détaillée: synthèse

- concevoir de façon détaillée tout ou une partie du système
- ce n'est pas que de la créativité, ou du bon sens
- on peut appliquer des méthodes (c'est ce qui fait de vous des spécialistes)
- *interaction designer, user-experience designer (UX designer)*: nouveau métier
- cf CHI et UIST

80

80



<http://hci.stanford.edu/bds/>

81

81

travail personnel+tp

- faire un prototype papier
- faire des scénarios de conception:
 - comme un scénario de travail, mais qui illustre l'utilisation du nouveau système
 - à partir d'un ou de scénarios de travail
- filmer le déroulement du scénario de conception sur le prototype papier

82

82

