ユーザインタフェース

（第１回）

五十嵐 健夫

主な内容

Human Computer Interaction (HCI)
User Interface (UI)

使いやすいソフトウェアをデザインするための方法論

最新のインタフェース研究の紹介

課題を通じたデザインと評価の実践

Course Credit

Attendance

Assignment
Programming and User Testing
(option: non-programming)

Schedule

- 6/2 User Interface Design, Evaluation
- 6/9 Sketching Interfaces (課題出題)
- 6/16 Information Visualization
- 6/23 Programming by Example
- 6/28 課題〆切 24:00
- 6/30 Human Robot Interaction
- 7/7 Real world Computing （課題講評）
- 7/14 No class (休講)
- 7/21 No class (休講)

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

HCI (Human Computer Interaction) とは何か

人間と計算機のかかわりに関する学問。
コンピュータ科学の一分野だが、学際的である。
　計算機科学・認知心理学・デザイン/アート

人間にとって使いやすいインタフェースを開発する。
計算機を使っている人間の行動について研究する。

Why is HCI Important?

- Life
  - 人間の命に関わる
  - 人間生活の質の向上に関わる
- Difficult
  - ソフトウェアの大部分を占める
  - よいものをデザインするのは簡単ではない
- Business
  - 生産性の向上・売り上げの増加に直結する
  - ブランドイメージに結びつく

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

D.Norman “design of everyday things” 「誰のためのデザイン？」

インタフェースデザインの重要性を訴えた本

「失敗するのは、ユーザの責任でなくデザイナの責任」

「デザインの工夫で、効率があがり失敗がへる」

よいデザインをするためのいくつかの知識

よいデザイン、悪いデザインの例とその分析

「アフォーダンス」

使い方を示唆する特徴

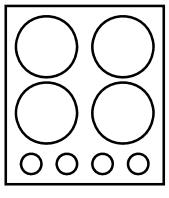
スロット＝差し込む
ノブ＝回す
ひも＝引く
ボタン＝押す

アフォーダンスをうまく使えば
説明が不要になり誤りが減る。

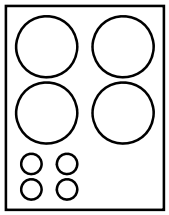
「アフォーダンス」

「自然な対応づけ」

ガスのレンジの例



分かりにくい



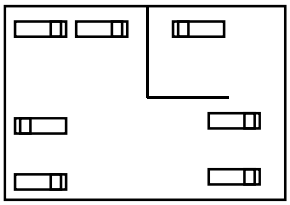
分かりやすい

「アフォーダンス」

「自然な対応づけ」

照明のスイッチの例

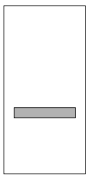
分かりにくい



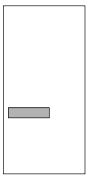
分かりやすい

「アフォーダンス」

「自然な対応づけ」　　ドアの例



開くのはどっち？



開くのは左♪

「可視性とフィードバック」

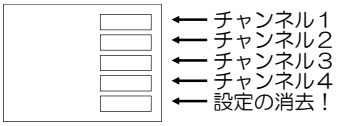
例）ボタンを恣意的な順に押す

フィードバックなし　要マニュアル
　　エラー多

ディスプレイあり　　マニュアル・記憶不要

何が起きているのか見える
入力に対して適切なフィードバックを返す

「エラーの防止」



無線装置の例

人は必ずエラーをする。
エラーを起こりにくくする＆被害を小さくする工夫が必要。

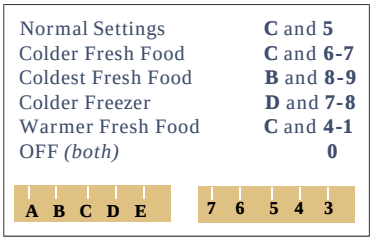
「概念モデル」

「物事がどう動作するのか、その原理に関する心の中のモデル」

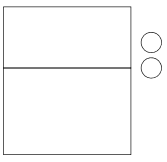
機械の動作の概念モデルを
うまく構築できると操作が楽になる。

ただしい概念モデルを提供し
それにあった動作をするように設計するべき。

「概念モデル」

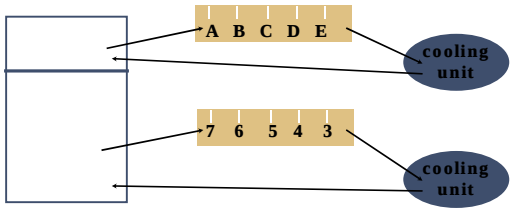


冷蔵庫のスイッチ



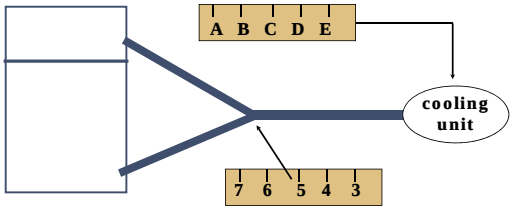
分かりにくいものの例。メンタルモデルを構築できない！

「普通に思い浮かべる概念モデル」



独立した制御

「この冷蔵庫の実際の動作モデル」



解決法

この様子がわかるように表示を工夫する　or
直感的にわかるようにシステムを作り直する

D.Norman 「誰のためのデザイン？」

「使いにくいデザインができる理由」

美的基準によって評価される（デザイン賞など）

デザインする人はエキスパートになってしまう

機能の豊富さが賞賛される。

購入するときにあまり考慮されない。

悪いのはユーザと思い込む。

。。。

D.Norman 「誰のためのデザイン？」

「使いやすいデザインのための原則」

外界にある知識を利用する。（ものの置き場所）

作業の構造を単純化する。（人間の短期記憶）

対象を目に見えるようにする。（冷蔵庫）

自然な対応付けを行う。（コンロ）

自然の制約や人工的な制約を活用する。（レゴ）

エラーに備えたデザインをする。（undo）

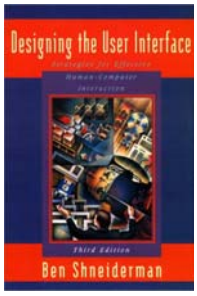
標準化する。（keyboard, 信号, カレンダー）

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

Design Rules デザインにおいて考慮すべき要素

B. Shneiderman
“Designing the User Interface”



B. Shneiderman "Designing the User Interface"

Goals: 満たすべきゴール（評価の基準）

- Learning time 学習時間
- Performance 操作の速さ
- Error rate エラーの発生率
- Retention 時間がたっても覚えているか
- Satisfaction 主観的な満足度

B.Shneiderman "Designing the User Interface"

Priorities

●Critical 生命に関わるもの（原子力・航空・医療）

●Business ビジネス・商業用システム（銀行）

●Office オフィス・家庭・エンターテイメント

●Creative 創造的な活動・デザイン

B. Shneiderman "Designing the User Interface"

デザインにおいて留意すべき点

●Physical 人間の物理的特性・場所の特性

●Cognitive 認知的・知覚的特性（e.g. 輝度と周波数）

●Individual 個人差

●Cultural 文化的・国際的な多様性　OX？

●Impaired 障害者・高齢者・子供 ユニバーサルデザイン

B. Shneiderman "Designing the User Interface"

Eight golden rules

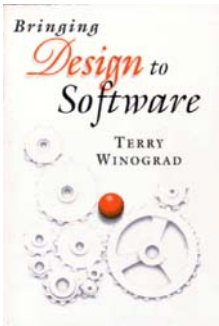
1. Consistency 一貫性を保つように （操作、色、配置、用語 ...）
2. Short-cut 頻繁な操作にはショートカットを
3. Feedback 分かりやすいフィードバックを
4. Group 操作をかたまり毎に処理できるように
5. Error エラーを防ぎ、また簡単に復帰できるように
6. Undo やり直しが簡単にできるように
7. Locus of Control 「自分で制御できている」という感覚をもてるように
8. Memory 短期記憶の負荷を減らすように　（～7±2）

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

Design Methods デザインの方法

ラピッド
プロトタイピング



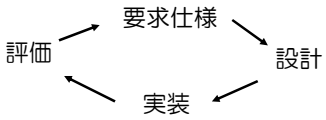
参考文献

Bringing Design to Software
Edited by Terry Winograd

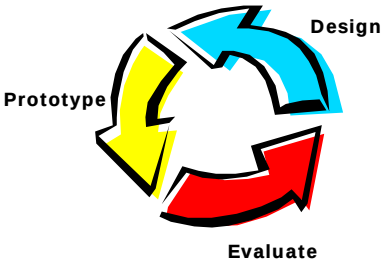
通常のソフトウェア開発

要求仕様 → 設計 → 実装 → 検証

インタフェースデザイン



Designing User Interfaces



手早く様々な動作を試せる環境が重要

プロトタイピングの重要性

- いろいろなデザインのバリエーションを試す。
  - 大きな探索空間の中からより良いものを選択できる。
- それぞれのデザインについて簡単にテストできる。
  - 完全な実装するより安くて早い
- ユーザにフォーカスを当てたデザインができる。

ラピッドプロトタイピングツール

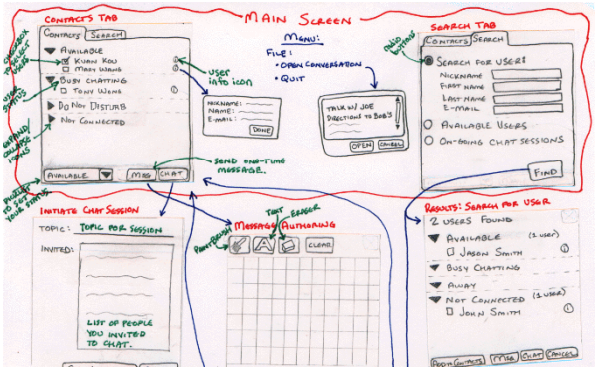
Low Fidelity

紙とペン、ホワイトボード
Wizard of Oz
HyperCard
MacroMedia Director/Flash
Visual Basic
Java

High Fidelity

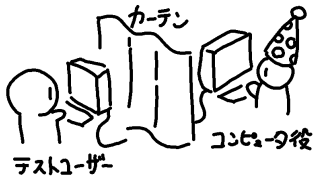
紙とペン、ホワイトボード

紙とペン、ホワイトボード



Wizard of Oz

人間が裏に隠れてコンピュータの振りをする。
（複雑なコードを書かずにテストを行う）



音声認識や文字認識、ＡＩを利用した
インタフェースデザインの検討に用いる

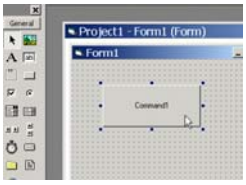
プロトタイピングツール

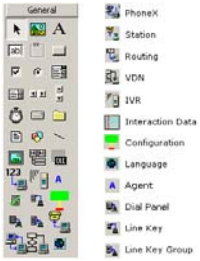
MacroMedia Director / Flash

時間軸に従った制御。デザイナによく使われる。NonGUI
簡単なスクリプトが書ける。

インタフェースビルダー

Visual Basic





GUI部品を並べていく。
イベントに対する動作を記述する。

Outline

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users

Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective（インタビュー、アンケート、フォーカスグループ）
  - Log Analysis（プロトコル解析、ログ解析、時間計測）
  - Observation（対話、ハーフミラー、ビデオ）

インタフェースの評価法

参考文献

Usability Engineering
Jakob Nielsen

Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective（インタビュー、アンケート、フォーカスグループ）
  - Log Analysis（プロトコル解析、ログ解析、時間計測）
  - Observation（対話、ハーフミラー、ビデオ）

ガイドラインに沿ったチェック

Heuristic Evaluation by Experts

1) Pre-evaluation training
   - give evaluators needed domain knowledge and information on the scenario
2) Evaluation
   - individuals evaluate and then aggregate results
3) Severity rating
   - determine how severe each problem is (priority)
     - can do this first individually and then as a group
4) Debriefing
   - discuss the outcome with design team

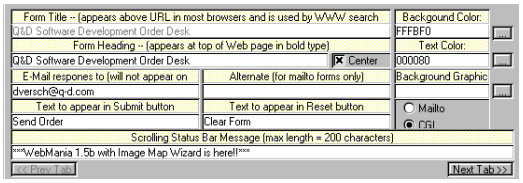
ガイドラインに沿ったチェック

Check List

1. シンプルで自然な対話
2. ユーザの言葉で話す
3. 記憶負荷を最小限にする
4. 一貫性
5. フィードバック
6. 出口を明らかにする
7. ショートカット
8. 適切なエラーメッセージ
9. エラーを防ぐ
10. ヘルプとドキュメンテーション

ガイドラインに沿ったチェック

1）シンプルで自然な対話

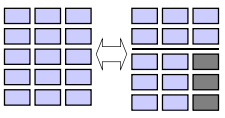


?

ガイドラインに沿ったチェック

1）シンプルで自然な対話

グラフィックデザインの原則
（ゲシュタルト理論）



残高　1,000円
2,000円

少ないほど良い　オブジェクト数、色数

ガイドラインに沿ったチェック

3）記憶負荷を最小限にする

例や単位を画面に表示する

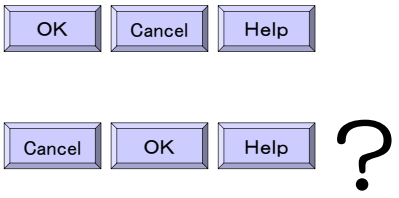
e.g. 日付を入力して下さい (DD-MM-YY 例 2-AUG-93)

少数のルールで多くの操作ができるように。

汎用コマンド　（コピー、UNDO、etc）

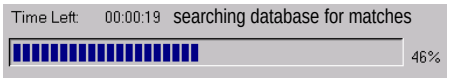
ガイドラインに沿ったチェック

4）一貫性



ガイドラインに沿ったチェック

4）フィードバック



瞬時と感じる応答	～0.1秒
問題ない応答	～１秒
対話に集中できる	～１０秒
要待ち時間表示	１０秒～

ガイドラインに沿ったチェック

8）適切なエラーメッセージ

?

```
Computer:    Type user name
Bissert      Bisseret
Computer:    Error, type user name
```

ユーザの理解できる言葉で理由を説明する。
解決法を提示すること。

ガイドラインに沿ったチェック

9）エラーを防ぐ

モードの使用を避ける。

どのモードなのかを明示する。

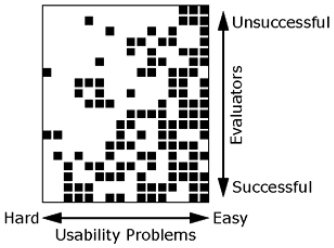
Heuristic Evaluation by Experts

実際にユーザを使うテストよりも安くて早い。
複数人で独立にチェックすること。（3～5人）

人によって違う問題を発見する。

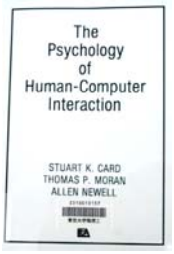
複数人を揃えることで問題を多く発見できる。

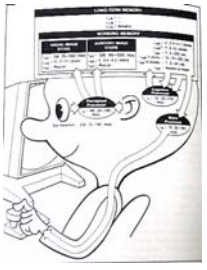
ポアソン分布になる



形式的タスク分析

Stuart K. Card　“The Psychology of Human-Computer Interaction”





認知心理学の知見・手法をHCIに応用した

Stuart K. Card

"The Psychology of Human-Computer Interaction"

KLM model (Key-stroke Level Model)

K: キー打鍵の時間(平均0.2秒。0.08～1.2秒)

P: マウスのポインティングの時間(平均1.1秒。0.8～1.5秒)

H: 手の移動時間(平均0.4秒)

D: 長さlの線分をn本描画する時間(0.9 n + 0.16 l秒)

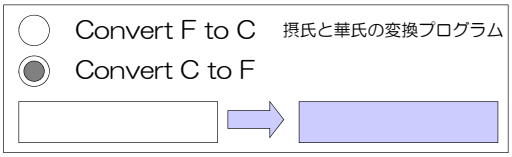
M: 精神的準備時間(平均1.35秒)

R: システムの応答時間(t)

例： プルダウンメニューから候補の一つを選択：

ＭＨＰＫＭＰＫ＝5.7秒

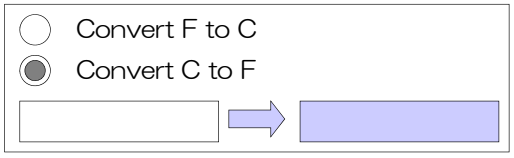
KLM model による分析の例



H　手をマウスへ
P　マウスを移動
K　マウスでクリック
H　手をキーボードへ
KKKK　4桁の数値入力
K　リターン

→ HPKHKKKKK

KLM model による分析の例



HPKHKKKKK

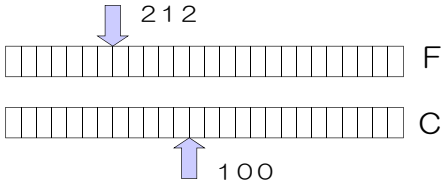
→ HMPKHMKKKKMK　合計7.15秒

摂氏と華氏があらかじめ選択されていた場合、

MKKKKMK　合計3.7秒

平均5.4秒

KLM model による分析の例



HMPKMPK　合計5.7秒

KLM model による分析の例

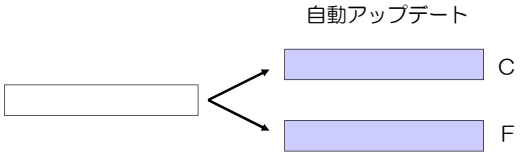
C 0 1 2 3 [Enter]

MKKKKKMK 合計3.9秒

0 1 2 3 C

MKKKKMK　合計3.7秒

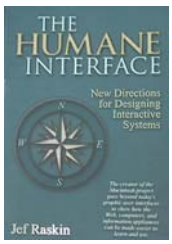
KLM model による分析の例



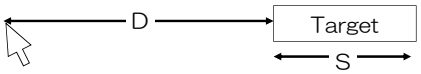
MKKKK　合計2.15秒

KLM model による分析の例

正確な時間の予測はできないが
複数のデザインの間の
検討に役に立つ。



Fit's law



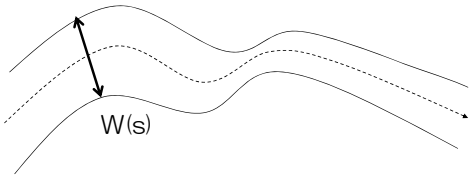
Time = a + b $\log_2$ (D/S+1)

Hick's law

n個の中から１個選ぶ

Time = a + b $\log_2$ (n+1)

Steering law

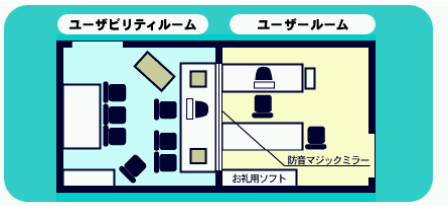


$$T = a + b\int_c \frac{ds}{W(s)}$$

Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective（インタビュー、アンケート、フォーカスグループ）
  - Log Analysis（プロトコル解析、ログ解析、時間計測）
  - Observation（対話、ハーフミラー、ビデオ）

Usability Laboratory



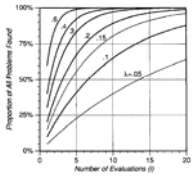
マイクロソフトのページより

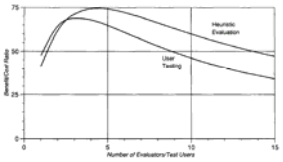
ユーザテストの仕方

1．準備
2．目的の説明
　「評価するのは製品であってユーザではない」
　「失敗するのは製品のせいである」
3．「いつでもやめてよい」と知らせる
4．部屋の装置について説明する
5．「声を出しながら考えること」を教える
6．「操作を助けることはしません」と伝える
7．ソフトウェアと作業内容を説明する
8．質問がないか聞いてから始める
9．まとめ
　何を知りたかったのか説明する
　質問がないか聞く。感想を聞く。説明を求める。

A Mathematical Model of the Finding of Usability Problems
J. Nielsen and T. K. Landauer '93

問題点を見つけるのに必要なユーザテストの数を解析している。





小さいシステムなら３～５人くらいでOK

http://www.useit.com/alertbox/20000319.html

Project Size	Cost	Benefits	Benefit/ Cost Ratio
Small	$10,000	$37,900	3.8
Medium-large	$18,000	$613,000	34
Very large	$46,000	$8,200,000	178

Table 5 Cost–benefit analysis for using the optimal number of test users in user testing.

注意すべき点

被験者の個人差が大きい。個人差で２倍の速さ。
被験者の学習。一度使ったらもう使えない。
要素の混乱。条件を平等に。順番のバランス。
　within group　　一人が両方テスト
　between group　　一人は片方だけ

倫理的問題。被験者はモルモット？
　　ストレス・プライバシー

Summary

- Background
- 「Design of Everyday Things」
- Design Rules
- Design Methods (Prototyping)
- Evaluation Methods
  - Without Test Users
  - With Test Users