
第1章 ユーザーインターフェース設計 の概要

- 1.1 ユーザーインターフェースとは
- 1.2 ユーザビリティとアクセシビリティ
- 1.3 人間中心設計とは

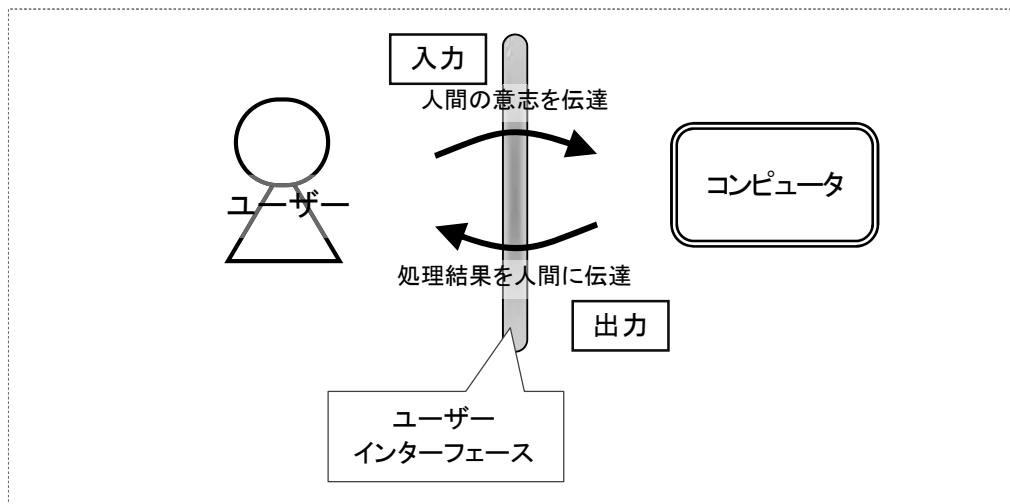
この章のねらい

この章では、ユーザーインターフェースの概要と人間中心設計の考え方を紹介します。

1.1 ユーザーインターフェースとは

1.1.1 ユーザーインターフェースの定義

ユーザーインターフェースとは、人間とコンピュータの間¹で情報をやりとりするためのしくみのことです。ユーザーがコンピュータを操作するために必要な情報を入力し、そのユーザーアクションに対してコンピュータが情報を出力します。



主なユーザーインターフェースに、コマンドを入出力する CUI (Character User Interface) やグラフィック表示の画面を操作する GUI (Graphical User Interface) などがあります。

現在広く利用されているユーザーインターフェースが GUI であるため、ユーザーインターフェースは一般的に GUI のことを指しています。このテキストは GUI を中心に扱います。

ユーザーインターフェースの歴史は長く、1940 年代までさかのぼります。1940 年代～60 年代の初期のコンピュータでは、ユーザーはパンチカードや紙テープを用いてコンピュータを操作していました。その後、1960 年代後半に UNIX が登場し、文字のみを入出力する CUI が主流となります。さらに 1980 年代に Macintosh、1990 年代に Windows 3.1 や Windows 95 が登場したのを機に GUI が普及し、現在に至ります。

ユーザーインターフェースの概念は不変ですが、ユーザーとコンピュータが情報をやり取りのしくみは歴史とともに変化しています。次世代のユーザーインターフェースとして、NUI (Natural User Interface) などが注目されています。NUI とは、人間の自然な動作 (触る、動く、話す、考える) によってコンピュータと会話するしくみのことです。

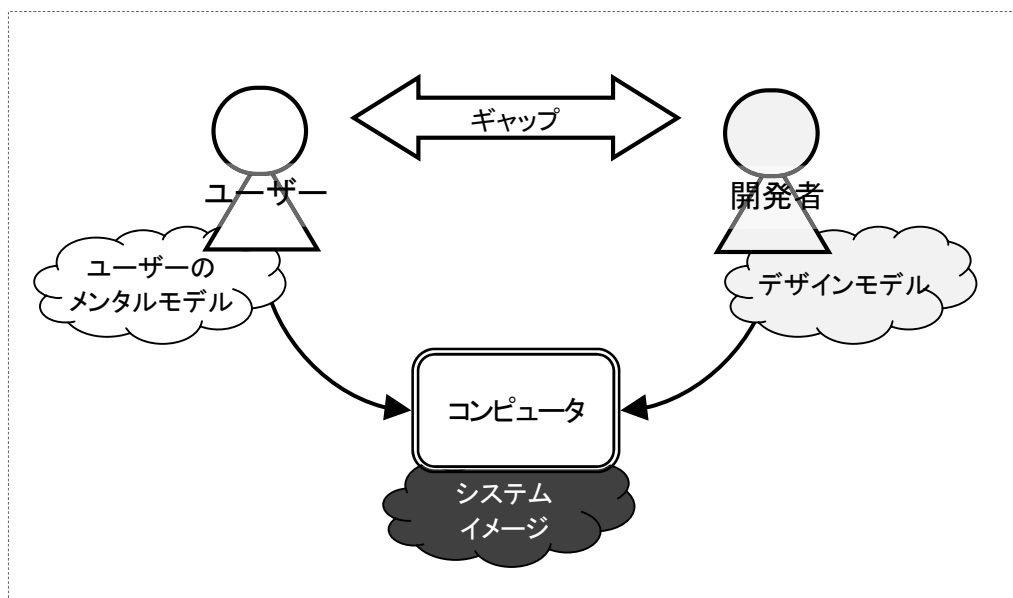
¹ 機械や自動車などのユーザーインターフェースは、HMI (Human Machine Interface) または MMI (Man Machine Interface) といいます。

1.1.2 ユーザーのメンタルモデル

ユーザーはシステムを操作するとき、目で見たと情報と、経験や知識を照らし合わせて、操作方法を考えようとします。この操作を繰り返すことで、ユーザーは経験と知識を新たに記憶します。メンタルモデルとは、人が心の中に持つ、システムの利用方法・動作に関するイメージのことです。

メンタルモデルには、次の特徴があります。

- 人はメンタルモデルに従った動作を期待する
- メンタルモデルと異なる応答があると、迷ったり、ミスをしたりしやすくなる
- ユーザーが持つメンタルモデルとデザインモデル²の差(ギャップ)が大きいと、ユーザーはシステムを使いにくいと感じる



ユーザーが持つメンタルモデルに対して、システム開発を行う担当者は、「このように設計すれば良いシステムになるはずだ」という、デザインモデルを持っています。

² 「デザインモデル」とは、システム開発者が考えた機能仕様や操作仕様のことです。

1.1.3 ユーザーインターフェースの重要性

前述した開発者のデザインモデルとユーザーのメンタルモデルのギャップが広がると、たとえレスポンスが速く処理性能の高いシステムであっても、入出力の画面の操作がわかりにくく、システムを利用するユーザーの作業効率は落ちてしまいます。また、操作のわかりにくさは、ユーザーのオペレーションミスにつながり、ときには大規模なシステム障害を引き起こす可能性もあります。

以下は、オペレーションミスに関連するシステム障害の事例です。

証券システムの誤発注問題	: 61 万円／1 株 を 1 円／61 万株 と誤入力
電話回線の不通	: 回線接続のサーバーHDD 交換時のデータ設定ミス
電話回線のサービス利用者への特典適用漏れ	: システムへの必要情報入力漏れ
金融システムの取引停止	: ATM などの接続台数に関わるデータ設定ミス
入札システムの金額表示の誤り	: ソフトウェア更新時の設定ミス
ミサイル発射の誤警報	: テスト用に残っていた「ミサイル発射」の警報データを動作確認の際に間違って選択
非接触 IC カードによる電子マネーのバス運賃を二重課金	: バス運転手による IC カード読み取り装置の誤操作
航空会社の搭乗手続き・荷物登録が停止	: チェックイン端末を管理するサーバー内の暗号化機能の有効期限設定ミス
医療保険料を誤徴収	: 市町村の担当者がデータ変更を誤るといった「うっかり」ミスが原因で 1 万 8223 人分の保険料を誤って徴収
航空機の尻もち事故	: 機体重量の入力ミス

ユーザーインターフェースの設計において、ユーザーにとって「わかりやすい」インターフェースにすることは必要条件です。そのためには、ユーザーのメンタルモデルとデザインモデルを一致させたシステムイメージを実現しなければなりません。さらに、ユーザーインターフェースには、「人はミスをする」ことを前提にして、ユーザーのミスをフォローする仕組みが必要です。

ユーザーインターフェースに問題があると、上記のようなトラブルにつながる可能性があります。だからこそ、システム開発におけるユーザーインターフェース設計は重要です。

REFERENCE



ヒューマンエラーの対策

ヒューマンエラーとは、人間が本来持っている特性が、人間を取り巻く環境とうまく融合しないために、結果として誘発されるエラーのことです。人間の認知特性はヒューマンエラーと非常に深い関係があるため、エラーを引き起こす人間の認知特性を把握しておくことで、ヒューマンエラー対策をユーザーインターフェースに適用できます。

以下は、ヒューマンエラー対策の4つのステップです。

1 エラーが発生しそうな作業の回数を減らす

- わかりにくい情報や機能は、はじめから配置しない

2 エラーの発生確率を減らす

- ミスをさせないよう操作に制限を設ける
- ミスをしても、簡単に復旧できるようにする
- わかりやすいレイアウトで、間違いや迷いを起こさないようにする
- 操作(作業)手順を工夫し、失敗をさせない
- エラーが発生しない操作をユーザーに学習させる

3 エラーを発見する方策を多重に設ける

- 自分で気付かせる
- 間違いを起しても、すぐにわかるようにする

4 エラー発生時の影響を軽減させるために予め備えておく

※機能設計、データベース設計などと合わせて対処

第4章 デザインパターン

- 4.1 画面レイアウト
- 4.2 ナビゲーション
- 4.3 コントロール部品
- 4.4 基本入力操作
- 4.5 フィードバック
- 4.6 ダイアログボックス
- 4.7 メッセージ

この章のねらい

この章では、ユーザーインターフェースの基本的なデザインパターンを紹介します。Web システムやモバイル端末などの UI に共通する基本的な UI デザインパターンを修得します。また、3章の「人間の特性」の内容と関連づけて、「なぜそのユーザーインターフェースにする必要があるのか」という根拠を説明できるよう修得します。

4.1 画面レイアウト

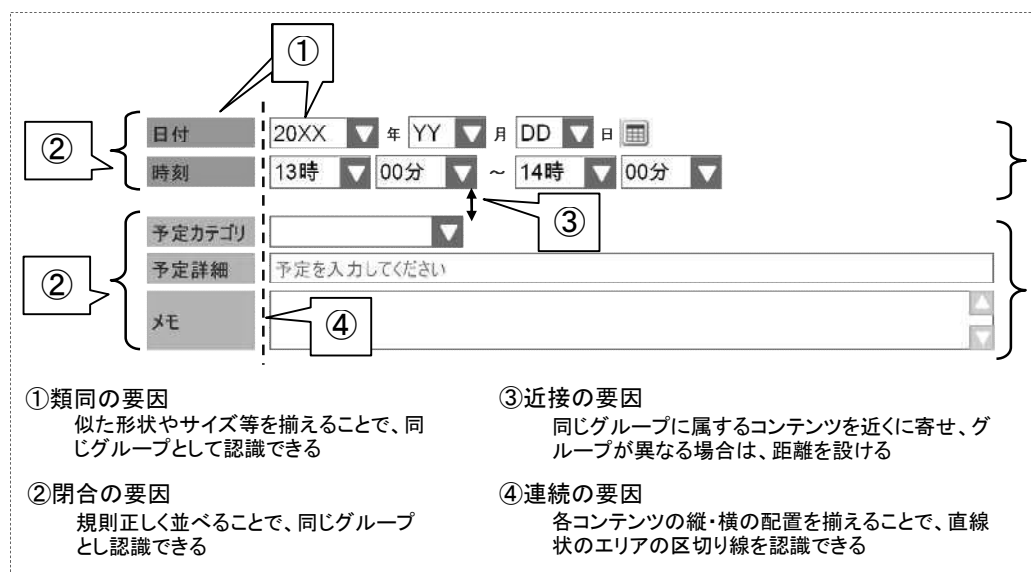
ユーザビリティの高いユーザーインターフェースを実現するための基本原則として、画面レイアウトの設計があります。画面のレイアウトは、単に規則正しく並べるだけではわかりやすいユーザーインターフェースにはなりません。前章で説明した人の特性を活かして、画面レイアウトを設計する必要があります。レイアウト設計の主なポイントは以下のとおりです。

<設計ポイント>

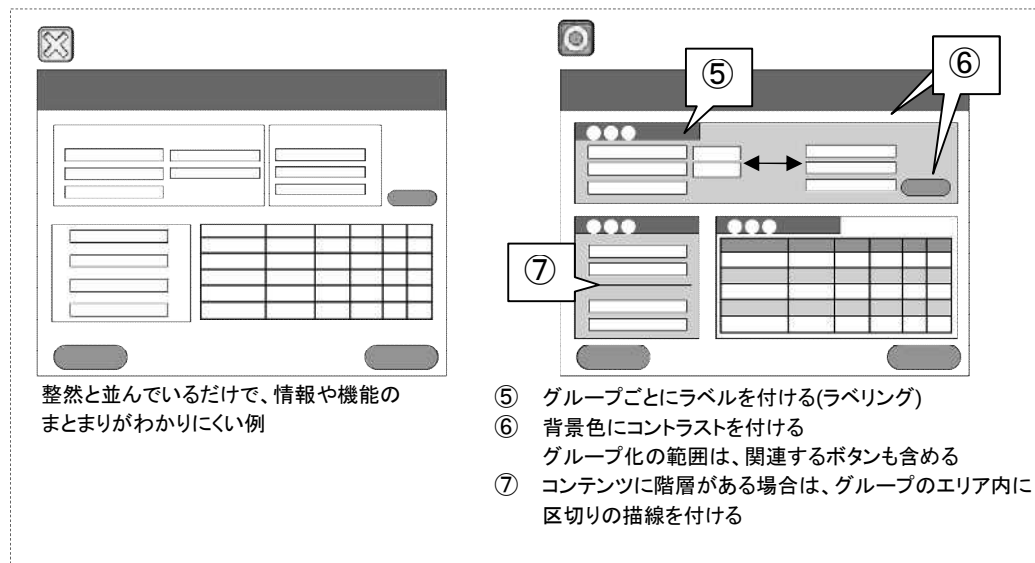
- 情報や機能(コンテンツ)をグループ化する

次のように同じ情報のグループ、機能グループごとの集まりを意識して、レイアウトを設計します。

以下は、プレグナツの法則にもとづいたグルーピングの例です。



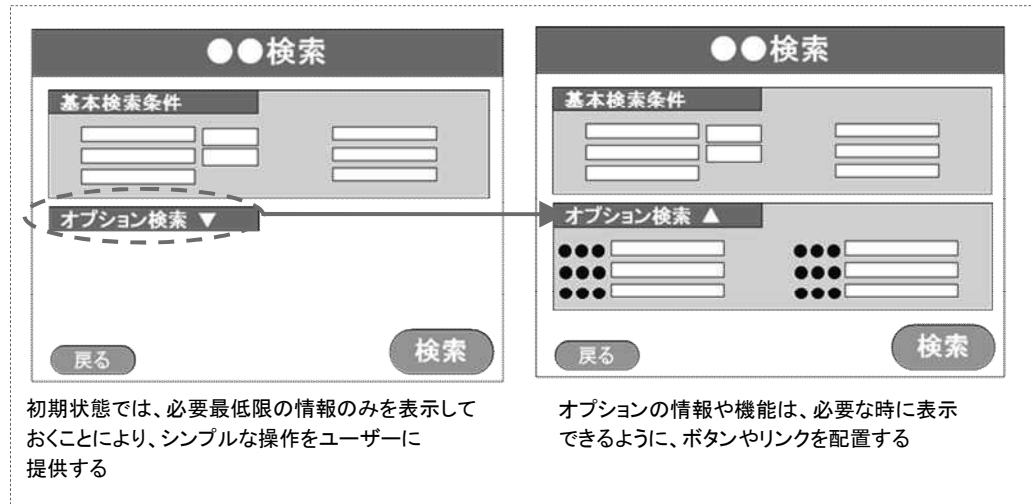
情報や機能としてのまとまりを適切にグルーピングする方法は他にもあります。



- 必要最小限の情報のみ表示する

多くの情報を画面上に表示すると、必要な情報を見つけにくくなります。そこで、ユーザーが必要な情報だけを表示するレイアウトを検討します。また、必要に応じてユーザーが情報の表示／非表示を選択できる仕組みを用意することが大切です。

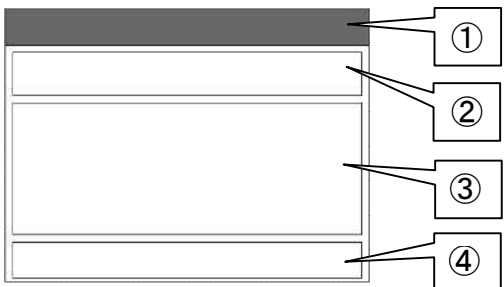
「この情報や機能は、ユーザーがこの場面で必要としているものかどうか？」という観点で、画面レイアウトを設計しましょう。



● 領域の明確化

情報や機能のグループの配置場所を決定するために、情報の表示領域と場所を明らかにします。

●一般的なエリアの配置例



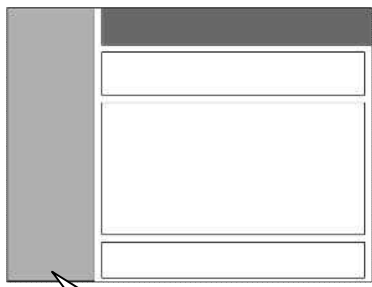
①ヘッダエリア
ロゴ・トップメニューへ戻る・ヘルプ・ログアウト・
ログインユーザ名・パンくずリストなどを配置するエリア

②メッセージエリア
ガイダンスメッセージ(操作方法、ユーザーへの指示など)を
表示するエリア

③メインエリア
検索画面、編集画面、参照画面ごとに作成するエリア

④ボタンエリア
画面遷移に関連する意思決定ボタン
(決定、キャンセルなど)を配置する

●左ペインにメニューを配置した例



メニュー一覧や検索機能など、いつでも使用できる項目を配置する

REFERENCE



画面レイアウトの検証方法

作成した画面レイアウトを領域ごとにあててぼかして、文字を判読不能な状態にし、純粋にユーザーインターフェースの視覚的な効果を検証する方法があります。これは、文字情報がなくても視覚的にある程度使い方が示唆されるユーザーインターフェースになっているかどうかを確認するために用いる方法です。そのほか、文字をぼかさずに、ランダムな文字列に置換して検証する「グリーキング」という方法もあります。