

4-1

ユーザインタフェース

Webサイト制作において重視される要素の1つにユーザインタフェースがある。ここでは、ユーザインタフェースの定義や、すぐれたユーザインタフェースが備えているべき要件について解説する。

4-1-1 Webサイトとユーザインタフェース

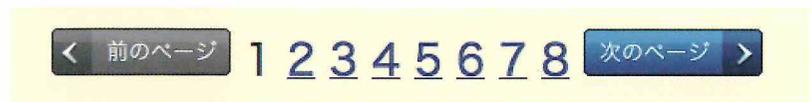
書籍や新聞などの紙メディアは、文字や図版といった、情報のみをもったメディアである。テレビもまた、個々のコンテンツは映像と音声のみから成り立っており、やはり情報のみをもったメディアと考えることができる。それに対してWebサイトは、① というもう1つの要素をもっているところが異なる。そのため、すぐれたコンテンツを提供するだけでなく、すぐれたインタフェースを提供することもWebサイトには求められる。

Webサイトにおけるユーザインタフェースとは、利用者とWebサイトとの接触面である。利用者はユーザインタフェースという接触面を介してWebサイトを操作し、コンテンツに到達したり、望みの作業を完了させる。そのため、ユーザインタフェースには、つぎの3つの役割を果たすことが求められる。

[1]現在の状態をユーザに知らせる

ユーザインタフェースには、Webページを見たときに、どの部分が操作可能であるのか、操作を行うことで何が起きるのか、またそもそも現在、操作可能な状態であるのか、といった情報をユーザに知らせる役割がある。

操作可能な要素、たとえばリンクやボタン、メニューなどは、それ以外の要素とデザイン的に差別化することで、その部分が操作可能であることを示すことができる。また、メッセージを送信するためのボタンには「送信」という文字を配することで、クリック後にメッセージが送信されることが事前に理解可能になる。ボタンなどは図4.1のように、操作

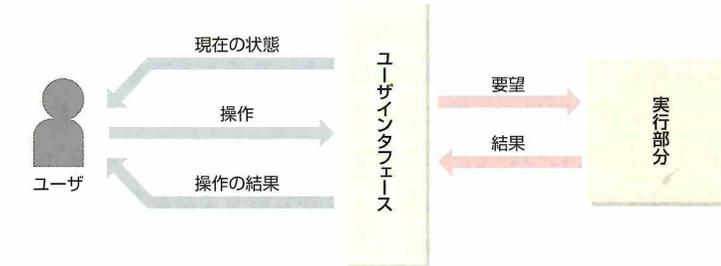


■図4.1——操作可能なボタン（次のページ）は青色で文字もはっきりしているのに対し、操作不可能なボタン（前のページ）は灰色で文字が目立たなくなっている。

可能な状態のときは濃い色で、操作不可能な状態のときは薄い色で表すことで、現在、操作可能であるかどうかを示すことができる。

[2]ユーザの操作を実行部分に渡す

システム的には、ユーザインタフェースは図4.2のように、あくまでもユーザとの接触面であり、ユーザから受け取った操作を、実際に処理を行う実行部分に渡すという、仲介役としての役割を果たしている。ここでいう実行部分とは、たとえばWebサーバやそのなかで稼動しているプログラムのことである。



■図4.2——ユーザとユーザインタフェース、実行部分の関係

[3]操作の結果をユーザに知らせる

ユーザインタフェースには、ユーザが行った操作の結果を知らせる役割もある。これには、ハイパーリンクのクリックのように、リンク先のページを表示するという結果そのものを目的とする場合と、メッセージが無事送信されたこと、決済が完全に終了したことなど、操作の結果報告を目的とする場合とがある。結果報告においては、詳しい説明が必要となることも多い。たとえば、決済が失敗したという結果を報告する場合、その原因が入力した情報のミスなのか、決済サーバの不具合なのか、クレジットカードの失効なのか、といった原因と、可能であれば対処法も提示されるべきである。

4-1-2 使いやすいユーザインタフェースの要素

使いやすいユーザインタフェースを実現するためには、さまざまな要素を考慮する必要がある。ユーザが自然に使いやすいと感じるためには、最低限、つぎの2つの要素が求められる。

[1]メンタルモデル

ユーザは、ユーザインタフェースに接するとき、過去の経験に基づく認識から「このように動くだろう」という予測をもって操作を始める。こうした認識や予測のプロセスをユーザの②という。実際に

操作を始めたとき、そうしたメンタルモデルとの差異が生じると、ユーザは戸惑いや使いにくさを感じることもある。そのため、一般的な動きと大きく異なるユーザインタフェースを構築する際は注意が必要である。

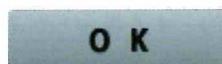
[2] ③

③とは認知科学で用いられる用語で、「与える」という意味のaffordから生み出された造語である。これは、モノそのものが自分自身の扱い方や操作方法に関する手がかりを与えてくれる、という考え方である。たとえば、部屋のなかに椅子が置かれているのを見た人間は、どのように扱うべきか疑問に思うことなく座るはずである。これは、椅子が人間に対して「座る」もしくは「座れる」という手がかりを与えているからである。同様に、壁に設置されたスイッチの押し方、蛇口から水を出す方法などについて、よほど変わったデザインでない限り迷うことはないはずである。逆にいえば、どう押したらよいかわからないスイッチ、どうすれば水が出るかわからない蛇口は、ユーザに対してアフォードをしていないデザインになっているわけである。

Webサイトのユーザインタフェースについても同様のことがいえる。図4.3のように、ボタンやリンクが「クリック可能である」ことをアフォードしていれば、ユーザは自然とそれらの要素に合った操作をしてくれる。逆に、図4.4のように、クリックできるかどうか見ただけではわからないデザインのボタンは、先程のスイッチや蛇口の例と同様に、ユーザを迷わせることになる。



■図4.3——クリック可能であることをアフォードしているボタン



■図4.4——クリック可能であることをアフォードしていないボタン

4-1-3 ユーザインタフェースの種類と役割

Webサイトにおけるユーザインタフェースは、ユーザが目的とする操作を達成するためのものである。こうした操作には、たとえばECサイトにおける商品の選択や決済、会員サイトへのログイン、あるいはゲームなど、さまざまなものがある。なかでも、ユーザが目的のコンテンツに到達するという操作のためのユーザインタフェースを④とよんで区別している。ナビゲーションについては、4-2でさらに詳しく説明する。ここではユーザインタフェースを根本的な目的で分類し、解説する。

[1] 指示選択

指示選択とは、文書内から特定のリンクをクリックしたり、ECサイトにおいて商品を選択するなど、コンテンツ内の要素のなかからユーザが何かを指し示して選択することである。ナビゲーションメニューなどから項目を選択することもこれに相当する。Webサイトにおける、最も基本的なユーザインタフェース要素といえる。

指示選択のされやすさは、デザインによって大きく影響を受ける。そのため、通常のリンクや一般的なナビゲーションメニューなど、各要素の間に優先順位がない場合は、デザインを均一化しておくべきである。それに対して、たとえば特売商品や新着情報など、とくに指示選択を受けたい要素が存在する場合は、優先順位に合わせてデザインの差別化を行うことで、ユーザからの指示選択を誘導しなくてはならない。

[2] 意思決定

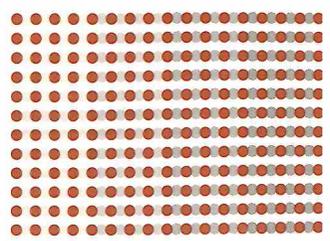
意思決定とは、決済を実行したり、入力した文字を送信したり、選択を確定させたりといったように、ユーザが何らかの意思決定をWebサイトに知らせることである。ECサイトにおいて、商品を希望数だけショッピングカートに入れる、といったものも意思決定に該当する。意思決定のインタフェースは、さらに2つに分けて考えることができる。

1つは、何らかの入力や選択をした後、ボタンなどをクリックすることで行う。意思決定のインタフェースはこの方式で実装されることが多い。この方式では、選択や入力と意思決定が別のアクションになっているため、改めて確認したり、修正したりといったことが可能である。そのため、ユーザに安心感を与える必要があったり、確実な選択が求められるときに適している。

2つ目は、セレクトメニューをマウスプッシュで開き、項目を選択した状態でマウスボタンを離すと意思を決定したことになるなど、ワンアクションで意思決定を行う方式である。この方式は、ボタンをクリックするアクションがない分だけ手間がかからず、スピーディに意思決定を行うことができる。一方で、選択がそのまま意思決定になってしまうため、慎重さが求められる場合には向いておらず、ECサイトなどにおける商品ジャンルを選択など、後からやり直しがきくような操作に向けた方式といえる。

[3] 選択肢

ユーザに何らかの選択肢を提示し、そのなかから選択を求めるインタフェースである。[1]の指示選択と似ているが、指示選択がコンテンツ全体のなかからユーザが任意の要素を指し示すものであるのに対して、選択肢では一定のルールに従ってカテゴライズされた選択肢のなかから、明確に選択を求める点が異なる。実装は一般的に、セレクトメニュー、ラジオボタン、チェックボックスなどが用いられる。



4-2 ナビゲーション

WebサイトもしくはWebコンテンツを制作する際に重要となるのがWebサイト構造を考慮したナビゲーション機能である。ここでは具体的事例を交えて、Webサイト制作におけるナビゲーション機能の実装について解説する。

4-2-1 ナビゲーションの目的と構成要素

Webサイトに掲載されている情報にユーザが効率的かつ柔軟にアクセスするためには、Webサイト独自のナビゲーションを用意するのが一般的である。

ここでは、ナビゲーションの目的と代表的な手法について解説する。

[1] Webサイト内を案内するためのナビゲーション

Webサイトは複数のコンテンツとWebページで構成される。そのためWebサイトを訪れたユーザが迷うことなく目的のコンテンツへとたどり着けるようにするためのナビゲーション機能が必要である。

この目的には、一般的につぎのような機能を実装することで対応する。

① ナビゲーションバー(メニューバー)

Webページ上部に配置された横位置のメニューを、ナビゲーションバー(メニューバー)とよぶ。Webサイト内のコンテンツを分類・整理

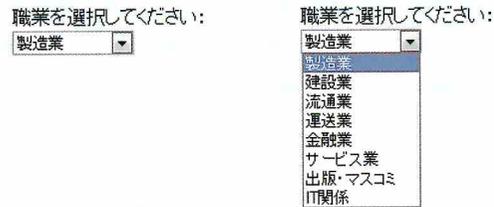
して階層的に管理する場合、分類名をメニュー化し、Webページ上部に配置するのが一般的である。このデザインは、多くのWebサイトで行われている。階層的な分類ではメインメニューからサブメニューといったメニュー自体の階層的表現も必要であり、階層が視覚的に判別できることが望ましい。これを実現させるために、ナビゲーションバーをメインメニューとし、Webページ左側にサブメニュー^{*1}を配置するデザインもある。

図4.8は、Webサイト内コンテンツを分類・整理したジャンル名などをメニューとして、Webページ上部に配置した例である。

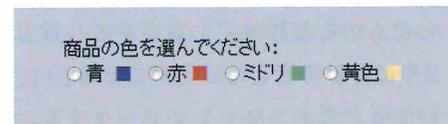


■ 図4.8——ナビゲーションバーとテキストによるリンクの例

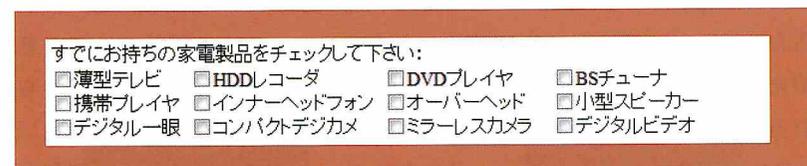
セレクトメニューは、職業や10代、20代といった年齢の範囲、都道府県といったように、数個から数十個ある選択肢のなかから択一的に選択することを求める場合に用いられることが多いが(図4.5)、複数選択も可能である。それに対してラジオボタンは、性別や商品の色など、数個程度の少数の選択肢のなかから択一を求める場合に用いられる(図4.6)。また、好きな音楽のジャンルや所有している家電の種類など、明確に複数の選択を求める場合にはチェックボックスが用いられる(図4.7)。



■ 図4.5——セレクトメニューの例



■ 図4.6——ラジオボタンの例



■ 図4.7——チェックボックスの例

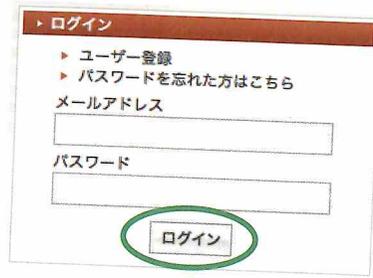
[4] 文字入力

掲示板やブログなどでの文章や、ログイン時のIDとパスワード、会員登録時の各種情報、ECサイトにおける購入数量など、文字入力を行うインタフェースである。

文字入力インタフェースを作成する場合、入力できる文字数の上限を設定したり、数値のみ入力可能とするなど、目的に応じて何らかの制限を加えることが多い。

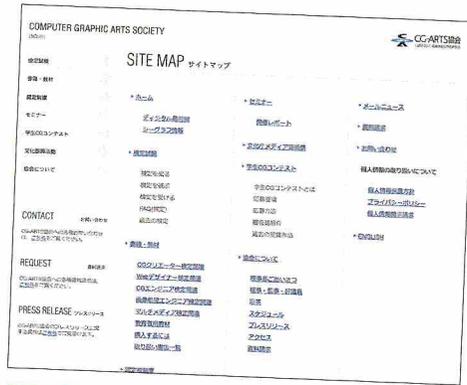
また、6-4-6[1]SQLインジェクションや[2]クロスサイトスクリプティングを防止するために、入力された文字はバックエンド側のプログラムでエスケープ処理を施すのが一般的である。

*1 メインメニューはグローバルナビゲーション、サブメニューはローカルナビゲーションとよばれる場合が多い。4-2-5[3]も参照のこと。



■図 4.9——ボタンによるリンクの例

②リンクにおけるナビゲーション
 リンクはコンテンツ内の各所に組み込まれるもので、その部分をクリックすると別のコンテンツへとジャンプするというナビゲーション機能を備える。そのため、そのことが明示的に理解できるようなデザインが必要である。
 リンクはテキストによるものと、ボタンのような画像もしくはサブミット(登録)ボタンで配置されるものがある(図4.9)。図4.8のようなテキストによるリンクは、コンテンツ本文内に組み込まれる。視覚的效果だけでリンクを行う場合もあるが、ユーザにクリックを強く促す場合は、テキストや画像により補完する場合もある。図4.9のようなボタンによるリンクは、ユーザにページ遷移を強く促す効果がある。フォーム画面ではユーザが入力した情報を送信するためのサブミットボタンを配置するが、これもデフォルト以外に画像ボタンを利用することもできる。



■図 4.10——一般的なサイトマップの例

③サイトマップ

サイトマップは、Webサイト内の膨大なコンテンツを一覧表示することで、全体構造を把握できるだけでなく、各コンテンツ名などをクリックすることで、各Webページへ直接アクセスすることができる(図4.10)。



■図 4.11——Webサイト内検索機能の例

④Webサイト内検索機能

Webサイト内検索機能は、インデックスによるコンテンツの検索ではなく、ユーザが指定したキーワードなどによってWebサイト内を検索できるものである。データベースなどと連携して独自の検索機能を装備するほか、Namazu^{*2}などの全文検索エンジンを利用することもできる(図4.11)。

*2 Namazuとは、日本語を含む任意の文字列で全文検索を行うことのできるソフトウェアもしくはプログラムコードのこと。

[2]Webサイト内での位置情報を示すナビゲーション

膨大なコンテンツで構成されるWebサイトの場合、ユーザは迷子になってしまう可能性がある。ユーザを混乱させないためにも、ユーザが現在閲覧しているコンテンツがWebサイト内のどの場所にあるのかが簡単に把握できるような、位置情報を示すナビゲーション機能を用意する。一般的にはつぎのような手法を用いる。

①パンくずリスト

階層構造で管理されたコンテンツの場合、現在位置とそのコンテンツが属する上位階層構造が把握できれば、ユーザが迷子になるのを防ぐことができる。一般的にはパンくずリストとよばれる手法が用いられる(図4.12)。



■図 4.12——パンくずリストの例

*3 パンくずリストとは、breadcrumb mbs listの和訳。「ヘンゼルとグレーテル」の童話のなかで、森のなかで迷わないためにパンくずを落として歩いたという話が基になってきた言葉。[topic path]ともよばれる。パンくずリストについては、4-2-5 [3]も参照のこと。

②メニューフォーカス

Webサイト内をナビゲートするメニューを装備している場合、メニュー項目のうち現在表示されているWebページに該当する項目の視覚的表示をほかの項目と変えることで、ユーザにそのWebページが表示されていることや、メニュー内での現在位置を把握させることができる(図4.13)。この手法をメニューフォーカスとよぶ。ただし、あまりにも深い階層にあるコンテンツの位置情報を示すことは困難なので、その場合はパンくずリストなどを併用する。



■図 4.13——メニューフォーカスの例

[3]コンテンツ内容・利用法のガイド

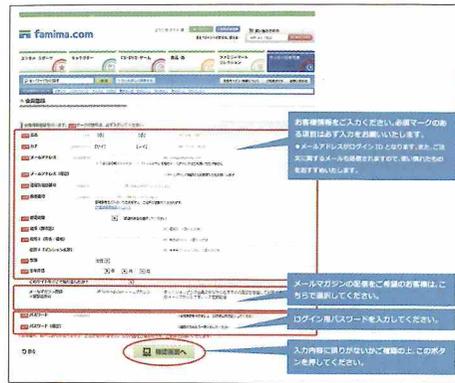
広い意味でのナビゲーションとしては、コンテンツ内容や利用法を解説するためのガイダンスの機能も必要である。一般的にはつぎのようなガイド機能、またはコンテンツを用意する。

①ガイド機能

ユーザの入力を促すフォームでは、入力する文字種や文字数などに制限をかけている場合が多い。基本的には入力チェックはシステムとして組み込み、必要以上の注意書きは表示しないほうが望ましい。しかし、文字数制限などがある場合はユーザが混乱をきたすこともあるので、「〇〇文字以内で入力してください」といった注意書きをガイド機能として設ける(図4.14)。



■図 4.14——ガイド機能の例 (©株式会社ファミマ・ドット・コム)



■図4.15——ヘルプコンテンツの例 (©株式会社ファミマ・ドット・コム)

②ヘルプコンテンツ

オンラインショッピングサイトでの会員登録や商品購入などのコンテンツでは、ユーザは通常のコンテンツ以上に緊張感を強いられる。また、複数のページ遷移で構成されるような場合は、処理完了までの手順が見えないまま入力作業を行うことに不安を感じる場合がある。そこで利用方法や処理完了までの概略を解説するようなヘルプコンテンツを用意することもある(図4.15)。

4-2-2 PCにおけるページレイアウト

PCで表示するWebサイトのページレイアウトを考える場合、まず考慮すべきことがナビゲーションとコンテンツの位置関係である。この位置関係の良し悪しによって、Webサイトの使いやすさが大きく変わってしまう。そのため、Webサイト制作者は、どのような位置関係で配置を行うか、慎重に考える必要がある。

その際に考慮すべきポイントは、人の視線の動きと、ナビゲーションやコンテンツといった各要素の重要度の優先順位である。Webサイトは通常、横組でつくられているが、その場合、人間の視線は上から下に、左から右に動くのが一般的である。したがって、要素の配置もより重要なものは上あるいは左側に置き、優先順位が下がるに従って、下あるいは右に配置していくことになる。

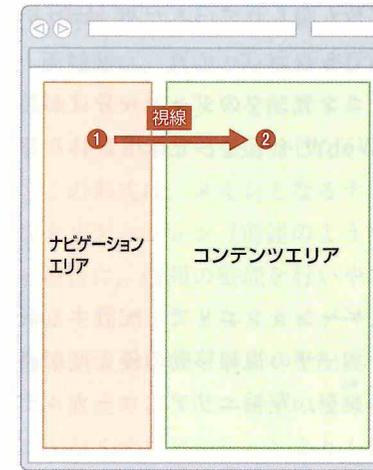
つぎに、そうした視線の動きと要素の優先順位を考慮した配置の典型例を5つ示す。

[1]左袖メニュー型

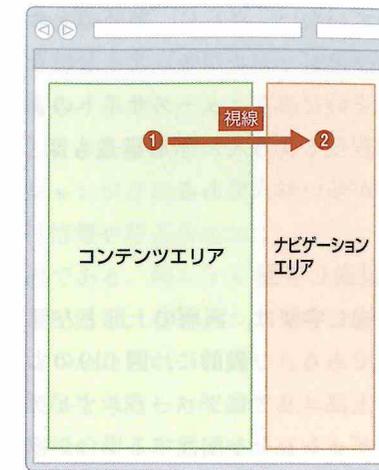
左袖メニュー型は、画面の左側にナビゲーションエリアを、右側にコンテンツエリアを配置する形式である。ほとんどの場合、Webページは縦スクロールを前提に作成されているため、ナビゲーションエリアに配置すべき要素が多い場合に適した形式である。また、ナビゲーションとコンテンツの両方で、縦方向にすべてのエリアを使えることから、ユーザの視線移動を自由に設計しやすい。

左袖メニュー型は、図4.16のようにユーザの視線がまずナビゲーションエリアに向かうため、ナビゲーション要素が重要な場合に採用される。たとえば、ECサイトのように、まずナビゲーションで商品ジャンルなどを選択していき、ある程度情報を絞り込んだところでコンテンツを閲覧し始める場合に適している。

逆に、コンテンツのほうが重視される場合は、図4.17のようにナビゲ



■図4.16——左袖メニュー型の配置図



■図4.17——右袖メニュー型の配置図

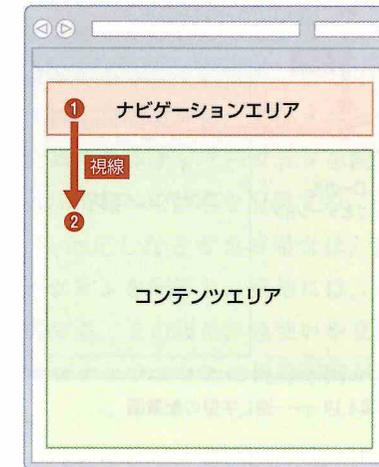
ーションエリアを右側に配置する、**右袖メニュー型**が採用される。たとえば、ブログのように、最新の記事から順番に読んでいくことが基本であり、ナビゲーション要素の操作が重視されず補助的に用いられるページの場合は、右袖メニュー型が適しているといえる。

左袖、右袖のどちらの場合も、メニューを何段階も展開できないため、階層の深いWebサイトには向いていない。階層が2、3段階程度のWebサイト向けの形式である。

[2]上部メニュー型

上部メニュー型は、画面の上側にナビゲーションエリアを、それより下すべてにコンテンツエリアを配置する形式である。図4.18のように、Webページ内を上から下へと移動するユーザの視線を捕まえやすいため、左袖メニュー型同様、ナビゲーションエリアが重視される場合に適している。また、画面の横幅をすべてコンテンツの表示に使えるため、コンテンツ内の配置の自由度が高く、さまざまな要素を配置しやすい。また、広告など、コンテンツと直接関係のない情報を配置するスペースも確保しやすい。

ただし、Webページでは横スクロールはほとんど使われないため、ナビゲーションエリアに配置できる要素数が画面の横幅以内に限定されてしまう。また、ドロップダウンメニュー^{*4}を用いたとしても、2、3段階しか階層をつくれなため、階層構造の深いWebサイトには向



■図4.18——上部メニュー型の配置図

*4 ドロップダウンメニューについては、4-3-3[1]を参照のこと。

いていない。さらに、前述のとおり要素数も限られているため、コンテンツのジャンルが多いサイトにも不向きである。

そのため、ニュースサイトのような、コンテンツのジャンル分けが数個程度と少なく、階層構造も深くないWebサイトなどで採用されることが多い形式である。

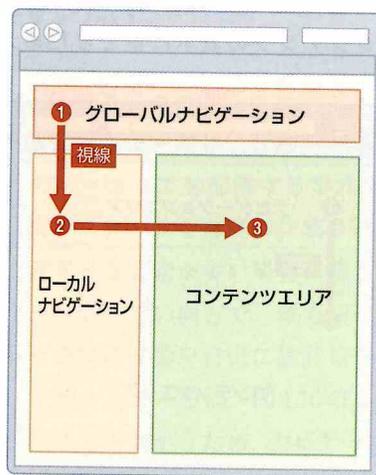
[3] 逆L字型

逆L字型は、画面の上部と左側にナビゲーションエリアを配置する形式である。一般的に、図4.19のように、ユーザの視線移動の優先度が高い上部エリアにグローバルナビゲーション^{*5}を、左袖エリアにローカルナビゲーション^{*5}を配置することが多い。

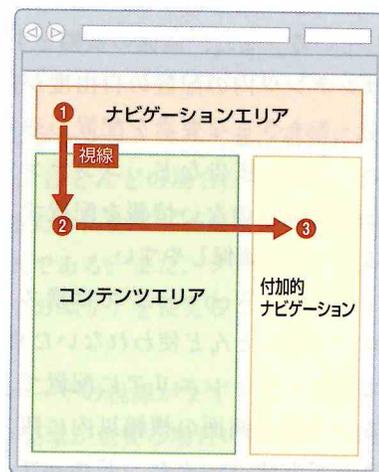
2つのナビゲーションを組み合わせることで、深い階層構造にも対応することが可能であり、また左袖のナビゲーションは項目数も多く配置できるので、コンテンツ量が膨大なうえに分類が複雑なWebサイトにおいて、情報を整理しやすい。そのため、規模の大きなECサイトやコーポレートサイトに適した形式である。

また、前述のような上部エリアと左袖エリアの連動を行わない形式もよく使われる。たとえば、上部エリアにコンテンツを分類した通常のナビゲーションを配置し、左袖エリアに注目情報や特売商品など、とくにユーザを誘導したいコンテンツへと導く付加的ナビゲーションを配置するといった形式である。その際、袖に配置するナビゲーションがコンテンツよりも優先順位が低い場合は、図4.20のように右袖に配置することも多い。

ただし、画面スペースの多くをナビゲーションエリアに割いてしまうため、コンテンツエリアが狭くなってしまうのが、この形式の欠点である。



■図4.19——逆L字型の配置図



■図4.20——逆L字型で袖メニューを右に配置したかたち

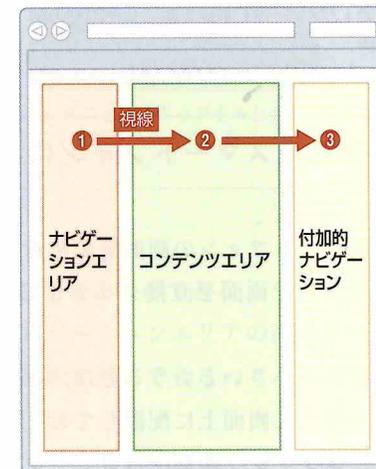
*5 グローバルナビゲーションとローカルナビゲーションについては、4-2-5 [3]を参照のこと。

[4] 両袖型

両袖型は、画面の左袖および右袖にナビゲーションエリアを、その間にコンテンツエリアを配置する形式である。基本的には、左袖メニュー型に右袖メニューを追加したかたちとして考えることができる。

この形式は、メインとなるナビゲーションが存在し、なおかつ付加的なナビゲーション（前述のような、注目情報や特売商品など）が存在する場合に、情報の整理を行いやすい配置である。図4.21のように左袖メニューにメインとなるナビゲーションを配置し、右袖メニューに付加的情報へのナビゲーションを配置するというかたちが一般的である。これは、前述のようにユーザの視線の移動順序を考慮した配置であり、メインとなるナビゲーションよりも付加的情報のほうが重要度が高いようならば、当然ながら逆の配置になる。

ただし、左袖と右袖の間にコンテンツエリアが存在し、ユーザの視線移動が妨げられるため、両袖の間の関連性が低いということに注意しなくてはならない。そのため、左袖メニューにグローバルナビゲーションを、右袖メニューにローカルナビゲーションを配置するといったように、階層構造をつくるには適していない。このような場合は、逆L字型を用いるべきである。



■図4.21——両袖型の配置図

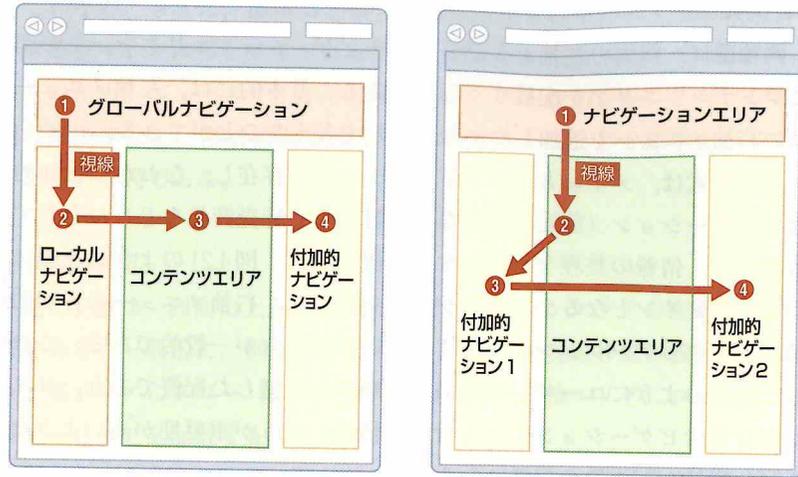
[5] 門型

門型は、上部メニュー型および両袖型を合わせた形である。情報量が多く、その種類も多い場合に、情報の整理を行いやすい配置であり、大規模ECサイトなどにおいて採用されることが多い。

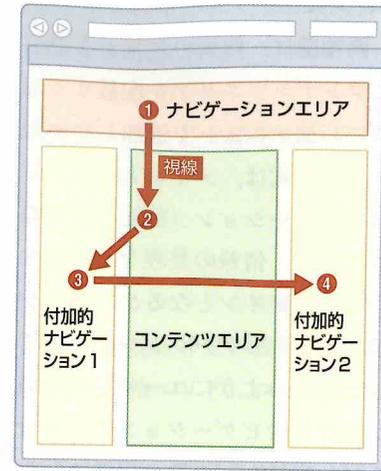
門型の配置は、逆L字型に袖をもう1つ足したと考える場合と、両袖型に上部メニューを足したと考える場合とがある。

逆L字型に袖を足したと考える場合、図4.22のように、上部メニューにグローバルナビゲーションを、左袖にローカルナビゲーションを配置することで階層構造をつくり、右袖に付加情報への誘導を配置する。

それに対して、両袖型に上部メニューを足したと考える場合は、図4.23のように、上部メニューにナビゲーションを配置し、両袖には、それぞれ異なった付加情報への誘導を配置する。この場合、色使いや文字サイズなどによって、両袖よりも先にコンテンツエリアに視線が向かうようにするというも行われる。



■図4.22——逆L字型を基にした門型の配置図



■図4.23——両袖型を基にした門型の配置図

4-2-3 スマートフォンにおけるページレイアウト

スマートフォンの画面構成を考える場合、画面が小さいということと、操作は指で画面を直接タッチして行うという操作性の2点を考慮する必要がある。

画面が小さいということは、コンテンツエリアとナビゲーションエリアを同時に画面上に配置しておくことが難しいということである。そこで、コンテンツ閲覧中はコンテンツエリアを主体に、ナビゲーション操作中はナビゲーションエリアを主体に表示するようにし、それぞれの場合に合わせて両者を切り替えながら利用してもらう、といった工夫が必要となる。

また、指で画面を直接タッチして行うため、PCのように小さなパーツを正確にタッチするのは難しく、個々のナビゲーションパーツは大きめに作成する必要がある。

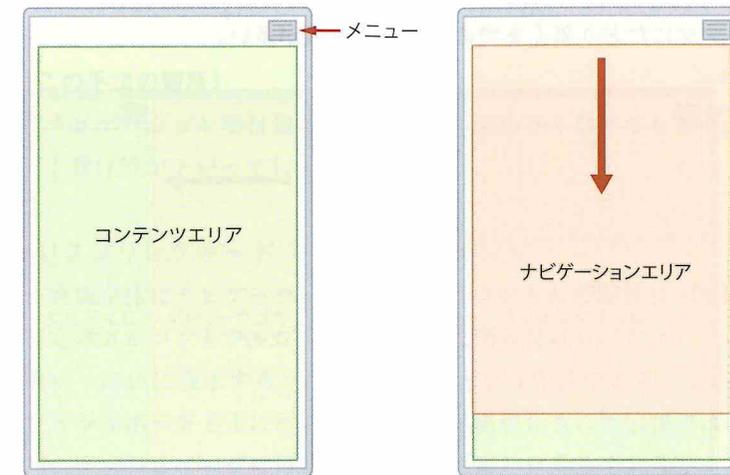
こうしたことを考慮して、スマートフォン向けの画面構成の手法がいくつも考案されている。ここではそのなかでも代表的な4つの手法について紹介する。

[1] ①

画面上部に1つだけメニューボタンを設けておき、それ以外の画面すべてをコンテンツエリアとして使用する配置である(図4.24 [a])。メニューボタンをタップすると、そこからナビゲーションエリアが下方に向かって滑り込むように入ってきて表示される(図4.24 [b])。

ナビゲーション項目をタップするとナビゲーションエリアが下位のナビゲーション要素によって書き換わる、というかたちで階層構造をもったナビゲーションを実現することも可能である。これはドリルダウンな

どとよばれている。ナビゲーション要素が多く、階層が深い場合など規模の大きなサイトに適した手法である。



[a] 画面上部にメニューを配置し、残りをコンテンツエリアとして利用

[b] メニューをタップするとナビゲーションエリアが滑り込んでくる

■図4.24——ドロップダウンの配置図

[この手法の長所]

- ・画面のほとんどをコンテンツエリアにすることができる。
- ・ナビゲーション項目の数に応じてナビゲーションエリアの高さを変更するなど、Webサイトの情報構造に柔軟に対応できる。

[この手法の短所]

- ・後述のタブやスプリングボードに比べてコンテンツにたどり着くまでのアクション数が多い。

[2] スライド

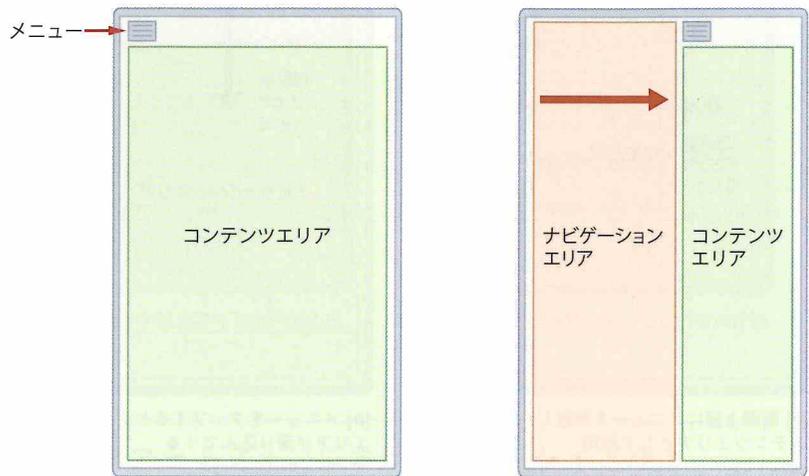
画面上部にひとつだけメニューボタンを設けておき、それ以外の画面すべてをコンテンツエリアとして使用する配置である(図4.25 [a])。メニューボタンをタップするとたとえば画面の左側からナビゲーションエリアが滑り込むように入ってきて表示される(図4.25 [b])。この手法は、ドローとよばれることもある。ドロップダウンと似た手法であるが、階層化された情報ではなく、同格の選択肢をユーザに提示したい場合に適した手法といえる。

[この手法の長所]

- ・画面のほとんどをコンテンツエリアにすることができる。
- ・画面の縦いっぱいにナビゲーションを表示できるため、多くの選択肢を表示できる。

【この手法の短所】

- ・画面の縦いっぱいを利用するため、ナビゲーション項目の数が少ないと、画面が間延びしてしまう。
- ・コンテンツにたどり着くまでのアクション数が多い。



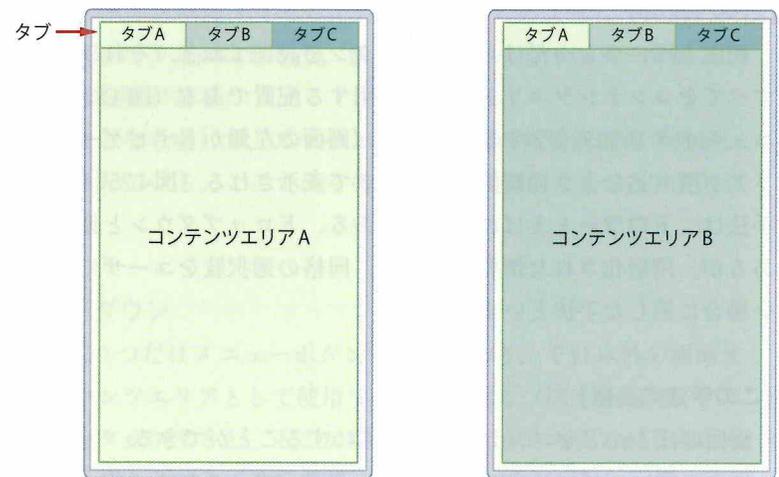
[a] 画面上部にメニューを配置し、残りをコンテンツエリアとして利用

[b] メニューをタップするとナビゲーションエリアが滑り込んでくる

■図4.25——スライドの配置図

【3】タブ

画面にラベルのつけられた「タブ」とよばれるナビゲーション要素を表示しておき（図4.26 [a]）、特定のタブをタップすると、そこに分類付けられたコンテンツやナビゲーション項目を表示する（図4.26 [b]）、という手法である。画面上部をタブ領域、それ以外をコンテンツ領域とする場合が多い。階層化された情報ではなく、同格の選択肢をユーザに提示したい場合に適した手法である。



[a] 画面上部にタブを配置する

[b] タブに関連付けられたコンテンツエリアが表示される

■図4.26——タブの配置図

【この手法の長所】

- ・どのような選択肢があるのか一目で把握しやすい。
- ・1タップで目的の画面が見られるという移動のしやすさ。

【この手法の短所】

- ・ナビゲーション項目数が多いと各タブが小さくなってしまい、視認性と操作性が下がってしまう。

【4】②

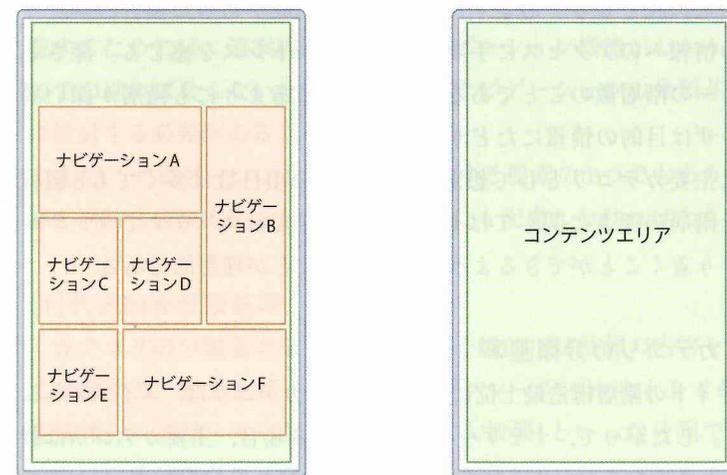
画面全体にナビゲーション項目をアイコンとして配置し（図4.27[a]）、特定のアイコンをタップすると、そこに関連付けられたコンテンツを画面いっぱいに表示する（図4.27 [b]）、という手法である。この手法は、ダッシュボードとよばれることもある。階層化された情報ではなく、同格の選択肢をユーザに提示したい場合に適した手法である。また、アイコンのデザインや配置の自由度が高くデザイン性の高いナビゲーションをつくりやすい。

【この手法の長所】

- ・どのような選択肢があるのか一目で把握しやすい。
- ・1タップで目的のコンテンツへ移動できる。
- ・タブよりも多くのナビゲーション項目を表示できる。

【この手法の短所】

- ・個々の選択肢の必要度に差がありすぎると、ユーザにとって不要な情報の多いナビゲーションになってしまう。
- ・ナビゲーションしかない画面となるため、ほかの手法に比べて情報量が少ない。



[a] 画面全体にナビゲーション項目を配置

[b] 各ナビゲーション項目に関連付けられたコンテンツエリアを表示

■図4.27——スプリングボードの配置図

4-2-4 ナビゲーションとWebサイト構造

Webサイトの構造とナビゲーションは密接な関係にある。とくにナビゲーションとしてのメニューの設計に際しては、コンテンツが階層構造として整然と配置されている必要がある。

[1] ツリー構造によるサイトの構造化

一般的にWebサイトは複数のコンテンツから構成され、さらにコンテンツは単一または複数のWebページで構成される。したがって、ナビゲーションとしてメニューを設計する場合は、まず第一にWebサイト内のコンテンツの整理と構造化が必須である。

コンテンツは基本的にツリー構造(階層構造)として整理する。たとえば、「企業情報」、「商品情報」、「サポート情報」など主要なカテゴリを想定し、さらに必要に応じてサブカテゴリを第二階層として想定する。ただし、階層化を行う際は、階層の幅や深さに注意しなければならない(図4.28)。

■図4.28——ツリー構造

幅とはメニューとして用意される**主要カテゴリ**や**サブカテゴリ**の個数であり、これは同時にユーザーにとっての**選択肢**となる。あまりにも多くの選択肢があった場合、ユーザーは目的の情報へのアクセスに手間取ることでストレスを感じる。深さとはメニューの階層数のことである。幅と同様にあまりにも階層が深い場合、ユーザーは目的の情報にたどり着きにくい。

とくに主要カテゴリとして想定するメニュー項目数は多くても8個程度、また階層の深さに関してはせいぜい4回程度のクリックでコンテンツにたどり着くことができるように想定することが理想的である。

[2] 主要カテゴリの分類基準

Webサイトの階層構造最上位に位置する主要カテゴリは、メインメニューとなる。したがって、ナビゲーションを考える場合、主要カテゴリは個数があまり多くならないように注意すると同時に、ユーザーにもわかりやすい基準で分類しなければならない。基本的に、Webサイト開設者独自の基準を、ユーザーは理解することはできないと考える必要がある。

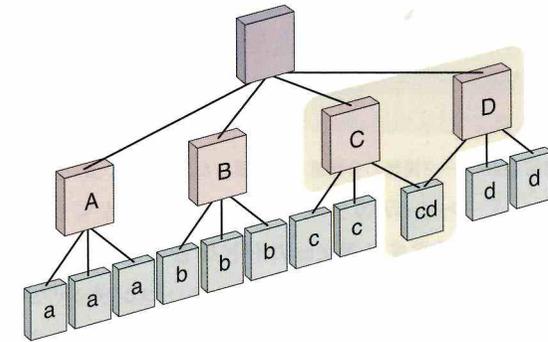
主要カテゴリの分類に一般的な基準を採用し、わかりやすいラベリング^{*7}を施すことで、ユーザーは主要カテゴリを概観しただけでWebサイト全体の構造を理解し、目的の情報をスムーズに探すことができるようになる。つまり主要カテゴリのわかりやすさが、Webサイト全体の使いやすさを決定するのである。したがって、主要カテゴリの分類基準は、検討を重ねて慎重に決定する必要がある。

*7 ラベリングについては、3-2-2を参照のこと。

[3] 相互排他的な分類

基本的に、カテゴリによる分類は相互排他的なものとしなければならない。一般的に階層構造における**相互排他的な分類**は、日常にある多くの検索対象が有している性質であり、ユーザーはそのことを前提にWebサイト内で情報を検索する(図4.29)。

したがって、主要カテゴリをはじめとして、各階層のメニューはユーザーが情報を検索するときの基準となるものである。このとき複数のカテゴリに同じコンテンツが存在していた場合、ユーザーは混乱をきたし、サイト全体の構造を把握できなくなる可能性が高い。



■図4.29——相互排他的ではないツリー構造

4-2-5 ナビゲーションの設計

複数のコンテンツで構成されるWebサイトをユーザーがスムーズに閲覧するためには、道案内としてのナビゲーション機能がわかりやすいものでなければならない。そのためにはナビゲーションの分類基準を慎重に検討する必要がある。

また、ナビゲーションは画面を構成する要素の1つでもあり、分類基準と同時に表現方法としてのデザインにも配慮しなければならない。

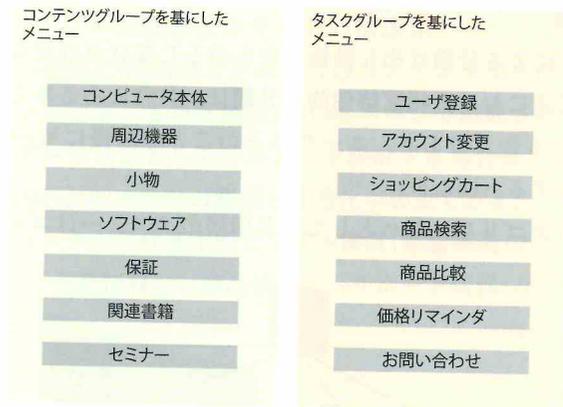
[1] 代表的な分類基準

カテゴリの分類基準を考える場合、一般的には情報の内容を基準として分類する。たとえば一般的な企業の商業サイトであれば「企業情報」、「商品・サービス情報」、「求人情報」といった分類である。これを**コンテンツグループ**とよぶ。

しかし、イントラネットなど業務システムとして活用するようなWebサイトの場合、ユーザーの目的はWebサイト上で提供される機能

(タスク)にあり、タスクを基にして分類するほうがわかりやすい場合がある。このようにWebサイト上で提供されるタスクを基準として分類することを③とよぶ。

厳密にはコンテンツグループとタスクグループは分類基準が異なるため、両者を混在させることは混乱を招く可能性があるが、メニュー項目数がそれほど多くない場合などには両者を混在させることもある(図4.30)。



■図4.30——2種類の分類基準を配置したナビゲーションとしてのメニュー

企業が運営するWebサイトなどでは、情報開示と同時にサービスの提供も行うのが一般的であるため、コンテンツグループとタスクグループの両基準によるメニューは必須ともいえる。それぞれ異なるナビゲーションとして表現することで、ユーザの混乱を招くことを少なくできると考えられる。

[2] ユーザインタフェースとナビゲーションの手法

ここまでで説明したナビゲーションを具体化する要素は別々に存在するのではなく、一般的には、Webページ上に複数同時に表示される。

ただし、あまりにも多くの要素を配置すると、物理的に多くの領域を使用してしまうことになり、コンテンツ自体を表示する領域が狭くなってしまいう問題点があげられる。また、要素ごとの役割をユーザが把握しきれず混乱してしまう可能性もある。

そのため、ユーザインタフェースの設計時には、どの要素を採用するか慎重に検討しなければならない。

[3] Webページ上に配置されるナビゲーションの種類

Webページ上に配置されるナビゲーションには、図4.31に示すような種類がある。

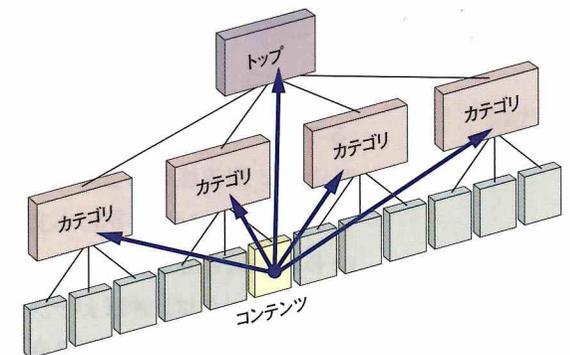


■図4.31——Webページ上のナビゲーションの種類

① ④ ④とは、基本的にすべてのWebページ上でつねに同じ位置にレイアウトされるナビゲーションのことである。したがって、グローバルナビゲーションにはWebサイト全体を自在に移動するためのメニューが並んでいることが基本であり、一般的にはWebサイトの入り口となるトップへのリンクが含まれている(図4.32)。

グローバルナビゲーションには、Webサイト内のコンテンツを階層構造としてとらえた場合の最上位階層(情報の大分類)がメニューとして配列されており、ユーザは大項目間の横移動を自由に行うことができる。また各メニューをクリックした場合には、その項目のトップページが表示されるべきであり、そのページにはサブカテゴリとしてメニューが配置されたインデックスページが採用されるケースが多い。またローカルナビゲーションが表示されるケースもある。

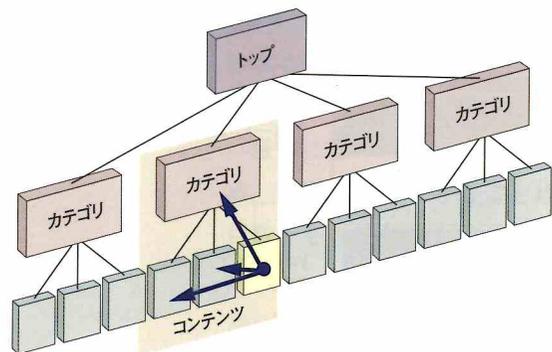
メニュー項目数が少ない場合、グローバルナビゲーションにはコンテンツグループとタスクグループが混在することもあるが、ユーザの混乱を引き起こさないためにも、その並び順などには注意を払わなければならない。



■図4.32——グローバルナビゲーション

*8 「トップ」とは、一般的な意味でのトップページまたはエントランスページのことを指す。ユーザ導線のためのページ名称については3-3-1を参照のこと。

② ⑤ ⑤とは、Webサイト内の特定のセクションに

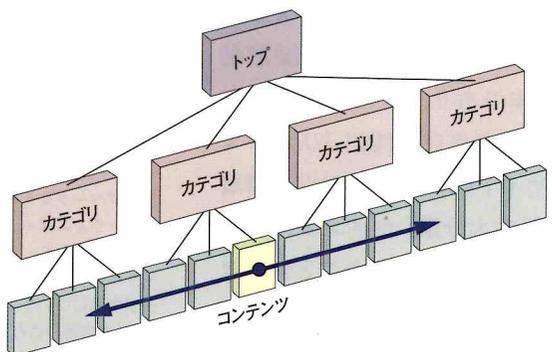


■図4.33——ローカルナビゲーション

配置されることでのみ利用されるメニューである。ローカルナビゲーションは、そのセクションに分類された複数のコンテンツ間をユーザが自在に移動するためのものである。グローバルナビゲーションと同様、そのセクションの閲覧中は、つねに同じ位置にレイアウトされていることが重要である(図4.33)。

たとえばグローバルナビゲーションの「商品情報」というメニュー配下に「パソコン」、「モニター」、「プリンタ」といったサブカテゴリが存在するような場合、これら3つのサブカテゴリがローカルナビゲーションとして採用される。さらに「パソコン」というメニュー配下に「デスクトップ」、「ノート」などのサブカテゴリが存在する場合は、「パソコン」配下のローカルナビゲーションとなる。

ローカルナビゲーションは、特定のカテゴリ内だけに配置され、ユーザはカテゴリ内のコンテンツと上位階層には移動できるが、異なるカテゴリには移動できない。



■図4.34——直接ナビゲーション

③ ⑥ ⑥とは、Webサイト内の階層構造とはまったく関係なく、コンテンツの内容に関連性のある別のコンテンツへと直接アクセスするためのリンクである(図4.34)。

たとえば、企業の商業サイトなどでは「トップ」上に新着ニュースや新製品のプロモーション用のコメント、もしくは画像などを配置しているケースが多い。これらはWebサイト構造を

無視して、「トップ」から直接深い階層にあるコンテンツへとアクセスすることができる。

また、関連性のあるコンテンツが同じセクションにない場合も多く、その場合はコンテンツの本文内にリンクを用意したり、関連リンクをまとめた領域を用意したりといった方法がとられる。

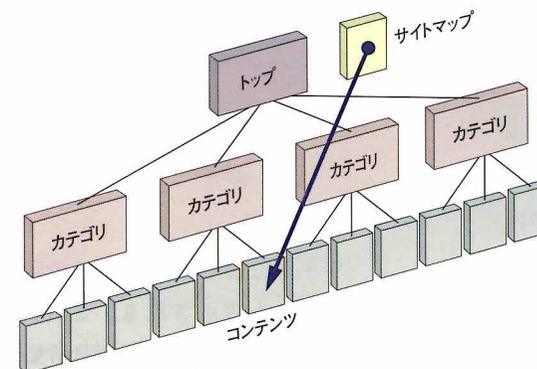
直接リンクは、ハイパーメディアとしてのWebの特性を最も具現化した機能であり、ユーザは直接リンクによって情報の内容に即した関連情報をトピックとして自由に閲覧できる。しかし、直接リンクを必要以上に配置したり、無秩序に配置した場合、ユーザが迷子になる可能性が高くなるので注意しなければならない。

④ ⑦ ⑦

とは、グローバルナビゲーションやローカルナビゲーションを補完する役割を果たすもので、具体的にはサイトマップやサブカテゴリのトップなどに採用するインデックス(リンク一覧)などがこれにあたる(図4.35)。

グローバルナビゲーションのメニュー項目数が多すぎるとユーザに不案内となるため、サブカテゴリであるローカルナビゲーションを用意することになるが、ローカルナビゲーションの項目数が多かたり、さらに下位階層が存在する場合、ユーザはコンテンツの配置構造を把握しにくい。そこでサイトマップでWebサイト全体の構造を、インデックスでセクションごとの構造をそれぞれ把握できるようにする。

また、場合によってはWebサイトの利用方法を紹介するガイダンスページのようなものを用意し、階層構造とは別の視点でコンテンツを案内するナビゲーションなども作成するが、これもリモートナビゲーションと考えることができる。



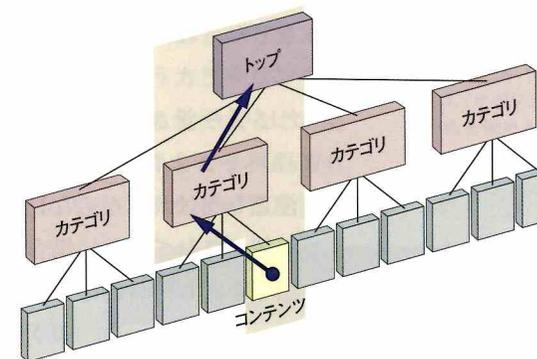
■図4.35——リモートナビゲーション

⑤パンくずリスト(階層ナビゲーション)

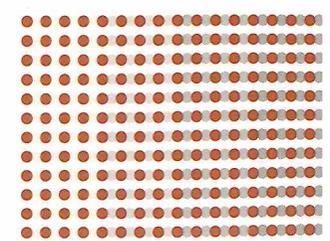
閲覧中のWebページの上位階層を羅列して表示することで、現在の位置情報を示すパンくずリストもナビゲーションの1つと考えられ、これを階層ナビゲーションなどと呼ぶ(図4.36)。

パンくずリストでは、リスト内に表示される上位階層にリンクを指定しておくことで、上位階層のコンテンツページやインデックスへ直接移動できるようになる。

パンくずリストには、ユーザがWebサイト内で各種コンテンツを閲覧している最中に、現在の位置情報を把握し、迷子になることを防ぐというメリットがある。同時にユーザがWebサイト外からWebサイト内の下位階層に直接アクセスしてきた場合(たとえば検索エンジンから直接アクセスするような場合)、パンくずリストが表示されていることで上位階層の構造が把握でき、検索効率が高まるというメリットも大きい。



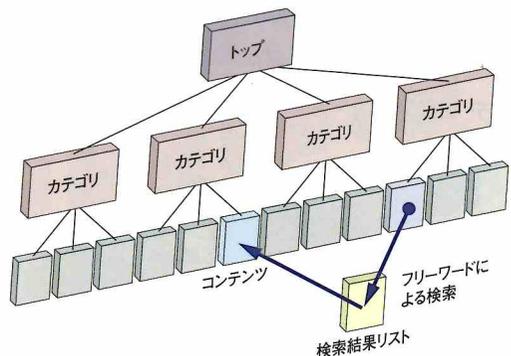
■図4.36——パンくずリスト(階層ナビゲーション)



4-3

ナビゲーションデザインの手法

インターネットにおいては不特定多数のユーザが情報を求めてWebサイトを訪れてくる。したがって、ユーザが目的の情報にスムーズにたどり着けるための手段としてのナビゲーションは、誰にでもわかりやすいことが理想である。ここでは、ナビゲーションデザインの手法について解説する。



■図4.37—Webサイト内検索機能

⑥ Webサイト内検索機能

先に紹介しているハイパーリンクを基本としたナビゲーションとは機能的に異なるが、検索機能もまたナビゲーションの1つととらえることができる。とくにWebサイト内検索機能は、ユーザが任意のキーワードでコンテンツをフリーに検索できることから、インデックスからのリンクの張り方によっては非常に効率性の高いナビゲーションとなる(図4.37)。

一般的にWebサイト内検索機能はフリーワードによるキーワード検索を基本とし、検索結果はインデックスとしてリスト表示される。これはCGIやWebアプリケーションなどで実現されるものであり、検索結果は指定されたキーワードに合わせてカスタマイズして生成される動的コンテンツ^{*9}である。ユーザはリスト上の任意の項目をクリックすることで、目的のコンテンツへと導かれる。

Webサイト内検索機能で重要なことは、コンテンツを特定するためのキーワードの選定にある。Webサイト開設者が想定するキーワードとユーザが想定するキーワードに食い違いがあった場合、ユーザは目的のコンテンツを検索することができず、逆にストレスを感じる結果となることが考えられる。全文検索であっても、キーワードのインデックスからのリンクの張り方によって効率性が左右されるという側面があることに注意しなければならない。

[4]ナビゲーションの表示・操作性の統一

Webページ上にナビゲーションを表現する場合には、統一性と一貫性に注意しなければならない。ナビゲーションのレイアウト配置やデザインが統一されていない場合、ユーザはナビゲーション自体の操作方法に戸惑いを感じ、目的のコンテンツを探る行為をやめてしまう可能性がある。

4-3-1 Webコンテンツにおけるナビゲーションデザイン

Webコンテンツはハイパーメディアとしての特徴をもつために、たとえば、キーワード検索の結果がリンク一覧となって表示されるといった、独自のナビゲーション機能を生み出している。しかし、多くのナビゲーション機能の操作性は日常的な行為に由来しているものが多い。

コンテンツを階層化してインデックスを用意するといったメニュー形式は、すでにほかのメディアなどでも採用されており、ユーザはその機能性のある程度理解していると考えられる。このようにユーザにとって既知の手法を採用することは、操作方法を習得するといったユーザのストレスを軽減することができるため有効である。

たとえば、PC上で動作するWebブラウザの画面に表示されるテキストリンクは、「未訪問：青色」、「訪問済み：紫色」というカラーリング^{*10}が定着している。これは、青と紫がその機能性を表現する色であるということではなく、単にWebブラウザが誕生した時点でたまたま採用されたカラーリングが常識化してしまい、現在もWebブラウザの初期設定値として採用されているにすぎない。したがって、リンク色として、ほかのカラーリングを採用することはもちろん可能である。しかし、多くのユーザが瞬時にリンクと判断するのは青と紫であるとえられる(図4.38)。

たとえば、コーポレートカラーなどに合わせて、リンクカラーに「青」と「紫」以外の色を意図的に採用する場合は、つぎのような基本的ルールを決めておく方法が考えられる。

- ・未訪問、訪問済みで色を変える。
- ・リンクテキストにはスタイルとして下線をつける。
- ・非リンク部分とコントラストのあるカラーリングにする。



■図4.38—一般的なテキストリンクカラーの例

*10 カラーリングとは、オブジェクトの色を決めること。

*9 動的コンテンツについては、6-1-3を参照のこと。

これにより、ユーザーに操作方法をあまり意識させないデザインのユーザーインタフェースへと近づけることができる。

4-3-2 JavaScriptとスタイルシート

Webサイトのナビゲーション機能の実装には、①が用いられることが多い。また、わかりやすいナビゲーションの実現のために、①とHTMLおよびCSSを用いて、いわゆるDHTMLを実装するケースもある。

ただし、すべてのユーザーがJavaScriptを使える環境にあるとは限らないため、実装には注意が必要である。

[1] JavaScriptのメリットとデメリット

JavaScriptによるユーザーインタフェースは、さまざまな用途で活用されている。JavaScriptではスタイルシートの属性をリアルタイムに変化させることができ、これによってWebページにさまざまな視覚的効果を与えることができる。

また、JavaScriptではWebブラウザに搭載されたエンジンによって、スクリプトをWebブラウザ側で処理することで動的なHTMLを表示できる。作成したプログラム(スクリプト)はHTMLとは別のファイルとして用意され、適宜HTMLからよび出すことができるため、たとえばグローバルナビゲーションのようにWebサイト内のコンテンツで共有するようなパーツを一元的に管理することができる。さらにJavaScriptはローカル環境で実行されるスクリプトであるため、サーバへの負荷がかからないというメリットもある。しかし、これらのメリットはあくまでもWebサイト開設者のものである。

ユーザーにとって、JavaScriptを利用することのメリットは、わかりやすいナビゲーションとして活用できることにある。

一方、Webサイト開設者およびユーザーのデメリットは、JavaScriptはすべての環境で実行可能であるとはいえない点である。JavaScriptを実装する場合は、ユーザーが意図的にJavaScriptの動作をOFFにしている場合なども含めて、それが使えない環境での代替手段などを用意しなければならない。

[2] JavaScriptの代替手段

4-3-2[1]で述べたように、JavaScriptはすべての閲覧環境で実行できるとは限らない。そのためJavaScriptによるプログラムを実装する場合は、非実行環境用の代替手段を用意しておかなければならない。しかし、実際にはJavaScriptが動作しない環境に対しては、通常のHTMLによるコンテンツを別途作成するのが一般的な手段である。

* 11 JavaScriptについては、6-3-4を参照のこと。

* 12 HTMLについては、6-3-2を参照のこと。

* 13 CSSについては、6-3-3を参照のこと。

* 14 DHTMLについては、4-3-3、6-3-5を参照のこと。

たとえば、リンク先を別のウィンドウで開くとき、JavaScriptによってウィンドウオープン^{*15}を制御しているケースも多いが、この場合には、JavaScriptが動作しない環境ではリンク先を閲覧することができない。したがって通常のHTMLのみによる同様のコンテンツを代替データとして用意するしかない。しかし、Webサイト開設者の制作予算やメンテナンスといった問題から、代替データが用意されることは少ないというのが現状である。

ただし、アクセシビリティ^{*16}の観点からは、何らかの代替手段を用意するか、もしくは最低限NOSCRIPT要素を指定して、JavaScriptの動作環境が必要であることを促すメッセージを表示するなどの対応が必要と考えられる。

4-3-3 DHTMLによるナビゲーション機能の例

ナビゲーション機能の実装にDHTMLを用いるおもな目的は、より優れた操作性やユーザービリティの実現、わかりやすさの向上、ユーザーへの利便性の提供、限られた画面内でより多くの情報を無理なくユーザーに提示する、といったものが一般的である。つぎに、DHTMLを用いたナビゲーション機能のおもなものを例として紹介する。

[1] ドロップダウンメニュー

メニューバーをクリック、あるいはマウスオーバーするとサブメニューが表示されるしくみをドロップダウンメニューという。図4.39のように、必要なときだけメニューが表示されるため、ナビゲーションエリアに割くスペースを最小限に抑えつつ、階層をもったナビゲーションを実現できる手法である。また、PCの一般的な操作に近いため、ユーザーにとっても理解しやすいという利点がある。図4.40のように、左袖メニューなどから横方向に開くメニューもよく使われている。

メニューが表示されるタイミングは、マウスオーバーを契機とするものと、クリックを契機とするものが考えられる。マウスオーバーを契機とする場合、クリックの手間がないうえに、マウスオーバーだけでメニューが開くため、その項目がメニューであるということに気づいてもらいやすいという利点がある。一方で、マウスオーバーするたびにメニューが開くことに煩わしさを感じるユーザーがいることにも留意すべきである。クリックでメニューを開く場合は逆に、その項目がメニューであることに気づいてもらうため、各項目の末尾に矢印を配置するなどのデザイン的な工夫が必要となる場合がある。

* 15 ウィンドウオープンとは、リンク先のページを新規のウィンドウに表示する際のウィンドウの表示方法設定のこと。

* 16 アクセシビリティについては、6-1-5を参照のこと。

* 17 NOSCRIPT要素を記述することによって、JavaScriptに対応していないWebブラウザに、その旨のメッセージを表示させることができる。



図 4.39 下方向に表示されるドロップダウンメニューの例



図 4.40 横方向に表示されるドロップダウンメニューの例

[2] ②

項目をクリック、あるいはマウスオーバーするとメニューが浮き上がるように表示されるしきみをポップアップという。技術的なしきみとしては [1] ドロップダウンメニューと同じだが、ドロップダウンメニューがナビゲーションエリアに設置されるのに対して、②はコンテンツエリア内に配置されることが多い。あるいは、図 4.41 のように、画面全体をイメージとし、そのなかにポップアップメニューを配置するといった利用法もよく行われている。

図 4.41 のように、ポップアップしたメニューから、さらにサブメニューを表示させることで、深い階層のナビゲーションを作成することも可能である(ドロップダウンメニューでも、同じことは行える)。



図 4.41 ポップアップメニューの例

[3] 項目のスクロール

いくつかの項目を横、あるいは縦方向に並べ、スクロールすることによって表示項目を切り替えていく手法である。少ない面積でより多くの情報を提供することが可能であり、図 4.42 のように EC サイトにおけるオススメ商品や、ニュースサイトなどにおける関連情報の表示などに利用されている。また Ajax を活用してサーバと通信することで、ユーザーの購入履歴に応じたオススメ商品や、その時々最新のニュースなどを動的に表示することも可能となる。

* 18 Ajax については、6-3-6 を参照のこと。



図 4.42 項目のスクロールの例

[4] 情報のポップアップ

コンテンツエリア内の何らかの要素をクリック、あるいはマウスオーバーすると、その要素に関連する情報などをポップアップした枠内に表示する手法である。通常の HTML のみで実装した場合に比べ、Webサーバとの通信を行わないため、スピーディに情報が表示される。そのため、図 4.43 のように、用語解説や追加情報といった別ページを開くほどではない情報を表示するのに適している。

また、図 4.44 のように、透過した状態で情報を表示することも可能であるため、ユーザーの閲覧行動を大きく妨げることなく情報を提供することができる。そのため、EC サイトなどのような、大量の項目を一覧表示したいが、個々の項目の情報も可能な限り詳しく提供したい、といった場合によく利用される。

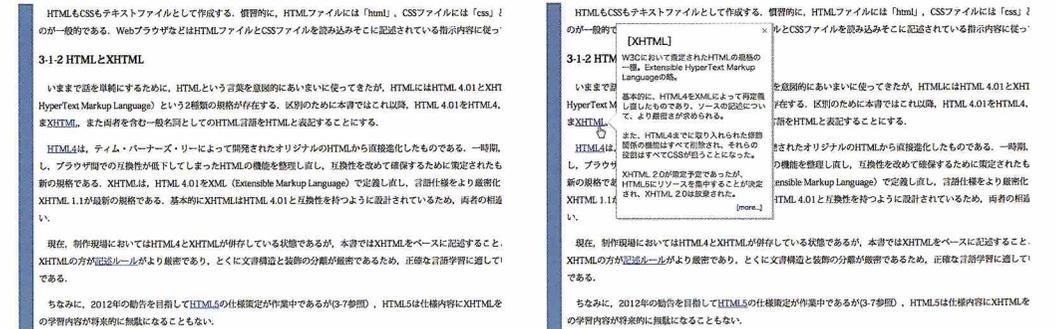


図 4.43 情報のポップアップの例

*19 ただし、Flashはスマートフォンでサポートされておらず、またPCでも急速に利用が減少しており、いままでFlashで実現してきた機能の多くがHTML5やJavaScriptなどで置き換えられつつある。2016年1月現在において、Flashの利用が完全になくなったわけではないが、今後の利用には注意が必要である。



■図4.44——透過した状態での情報のポップアップの例

[5] フォーム入力におけるエラー処理

ユーザインタフェースにおけるDHTMLの活用分野として、フォーム入力におけるエラー処理も重要である。HTMLのみで実装されたフォームでは、ユーザが情報をすべて入力し、Webサーバに情報を送信した後に入力ミスをチェックすることが多いため、ユーザに負担をかけることが多い。それに対してDHTMLを用いた場合、フォームの各項目への入力中に入力ミスのチェックが行えるため、ユーザは逐次修正しながら入力が行える。こうした処理の実装はユーザにとっても、Webサーバにとっても負担軽減となるものである。

4-3-4 Flashによるインタフェース^{*19}

よりダイナミックかつエンタテインメント性に富んだコンテンツとしてFlashムービーがWebサイトで利用されるケースが多い。Flashもまた柔軟性のあるユーザインタフェースを実現するための技術として評価することができる。

ただし、動作環境としてプラグインソフトを必要とし、あらゆる環境で動作するわけではないため、その利用には注意が必要である。

[1] Flashムービーのメリットとデメリット

Flashムービーはプラグインソフトさえインストールされていれば環境に依存することなく、誰でも同じコンテンツを楽しむことができる。また、音声やアニメーションといったリッチメディアとしての要素が使用できることから、とくにエンタテインメント性の高いコンテンツで多用されているのが現状である。

ある時期、サイト全体をFlashムービーだけで制作する、いわゆるフルFlashコンテンツのWebページの流行もあったが、それはマルチメディア素材を活かしてコンテンツ内容に即したさまざまなナビゲーションシステムを構築できるからである。また、Flashはユーザの操作に対する画面内容の動的な変化をページの再読み込みなしで可能にするた

め、商品プロモーションのようなコンテンツだけでなく、複雑なWebアプリケーションでも活用意義が大きいといえる。

ただし、公共性の高いコンテンツには、Flashを利用すべきではない。プラグインソフトがインストールされていない環境で動作しないという問題だけでなく、Flashによる過度の演出がユーザにとってはストレスとなる可能性もあるからである。ユーザの目的が情報検索の場合、一般的にユーザは情報に単純な操作ですばやくたどり着くことを期待している。

[2] インターネット上のリソースとしての問題点

インターネット上に展開されているHTMLデータはいわば全世界的な共有リソースともいえる。HTMLは文書構造化言語としてテキストを規定するものだが、FlashはHTMLデータのような構造とはならず、インターネット上に展開されるリソースとしては特異なデータである。

インターネット上での情報検索対象としての問題もあり、前述したようにとくに公共性の高いコンテンツでは利用すべきではないと考えられる。また、すべてのユーザがPC上のWebブラウザでアクセスしてくるとは限らないため、幅広いユーザに向けて発信される情報もFlashを使うべきではなく、より純粋なHTML(もしくはXHTML)として提供されるべきであるといえる。

[3] Flashの代替手段

FlashコンテンツもJavaScriptと同様に、代替手段としては、同様の情報を掲載したHTMLデータを別途作成すると同時に、プラグインソフトをダウンロードできるWebサイトへのリンクを用意することが考えられる。ただし、すべての環境でプラグインソフトが利用できるわけではないことにも注意しなければならない。

chapter 4

keywords

■ユーザインタフェース

- ・ユーザインタフェースは、「現在の状態をユーザに知らせる」、「ユーザの操作を実行部分に渡す」、「操作の結果をユーザに知らせる」という役割を果たしている。
- ・ユーザインタフェースのあり方を考えるうえで、ユーザが自然に使いやすいと感じる要素について考察することは非常に有用である。
- ・ユーザインタフェースには、「指示選択」、「意思決定」、「選択肢」、「文字入力」などの種類がある。

ユーザインタフェース	メンタルモデル	アフォーダンス	ナビゲーション
指示選択	意思決定	選択肢	セレクトメニュー
ラジオボタン	チェックボックス	文字入力	

■ナビゲーション

- ・Webサイトに掲載されている情報を、ユーザが効率的に取得できるようにナビゲーション機能を用意する。
- ・ナビゲーションを適正に行うため、適切なナビゲーションパーツを使用する。
- ・それぞれのWebサイトの構造に合ったナビゲーションを設計し実装する。

ナビゲーションバー(メニューバー)	リンク	サブミット
フォーム	サイトマップ	Webサイト内検索機能
パンくずリスト	メニューフォーカス	ガイド機能
ヘルプコンテンツ	左袖メニュー型	右袖メニュー型
上部メニュー型	逆L字型	両袖型
門型	ドロップダウン	ドリルダウン
スライド	ドロワー	タブ
スプリングボード	ダッシュボード	ツリー構造(階層構造)
階層化	主要カテゴリ	サブカテゴリ
ラベリング	相互排他的な分類	コンテンツグループ
タスクグループ	グローバルナビゲーション	トップ
ローカルナビゲーション	直接ナビゲーション	リモートナビゲーション
インデックス	階層ナビゲーション	

■ナビゲーションデザインの手法

- ・Webコンテンツに実装するナビゲーションデザインは、誰にでもわかりやすい必要がある。
- ・JavaScriptとスタイルシートを使ってナビゲーション機能を実装する。
- ・ユーザインタフェースにFlashを使う。

カラーリング	リンクカラー	JavaScript	HTML
CSS	DHTML	代替手段	ウィンドウオープン
アクセシビリティ	ドロップダウンメニュー	ポップアップ	ポップアップメニュー
スクロール	エラー処理	Flash ムービー	