

# 携帯電話向け組み込みブラウザの開発

山中 弘 泊 陽一郎 吉田 玲子

三菱電機株式会社 情報技術総合研究所

## 1. はじめに

携帯電話の普及と共に、様々な情報サービスが提供され始めている。その一つとして、移動機からのインターネットアクセスサービスが挙げられる。

携帯電話向けブラウザでは、小さな画面で表示を行うこと、限られたキーによって操作を行うこと、小さなメモリ容量で処理を実行すること、などの制約事項があり、ウィンドウインタフェースとキーボード/マウスを用いて操作を行うパソコン上の一般的なブラウザと比較し、実装を工夫する必要がある。

この発表では、我々が開発を行った携帯電話向け組み込みブラウザについて、上記の制約事項の観点からみた特長を述べる。

## 2. 表示方法

我々がブラウザを実装した携帯電話の表示画面スペースは、98ドット×96ドット(全角文字ベースで8文字×8行)と非常に小さい(図1)。

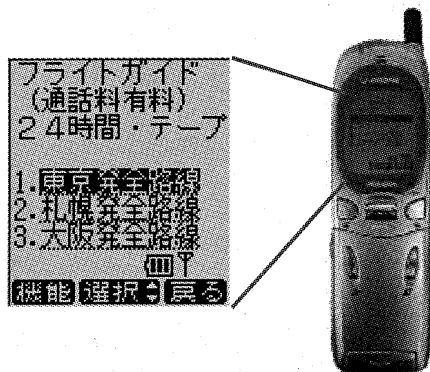


図1. 携帯電話ブラウザ画面表示例

このため、パソコン上の標準的なブラウザと同じ表示を行っていたのでは、ユーザがコンテンツ全体を表示するために縦/横スクロールを併用しなければならなくなることが予想される。

横スクロールが極力発生しないよう、我々は次のように表示方法を工夫した。

Embedded browser for mobile phones

Hiroshi Yamanaka, Yoichiro Tomari, Reiko Yoshida  
Information Technology R&D Center, Mitsubishi Electric Corporation

- ・簡条書きアイテムのインデント  
簡条書き入れ子の階層数について上限値を決め、それを超えたときは上限値の入れ子階層のインデントのまま表示を行う。
- ・表の自動折り返し  
横スクロールの発生頻度を抑えるため、表の中の1つのセルの最大幅は全角8文字とし、セル内で自動的に折り返す。
- ・PRE タグの自動折り返し  
PRE タグで挟まれた部分のテキスト表示についても全角8文字で折り返し、画面内で収まるように表示する。
- ・プッシュボタンやテキストボックスの最大幅  
プッシュボタンやテキストボックスの最大幅を全角8文字分とし、指定された表示幅で表示域に収まらない場合は改行して表示する。複数行テキストボックスでは、指定された桁数が表示域に収まらない場合は、行数×桁数が等しくなるよう行数を増加させる。

## 3. 操作方法

### 3.1. 操作の統一/容易化

操作の一貫性を保つため、ほとんどの操作キーの動作を他の機能と合わせた。

また、コンテンツの操作処理を容易に行えるようにするため、ソフトキーと操作との対応をガイダンスエリアに表示した(図2)。

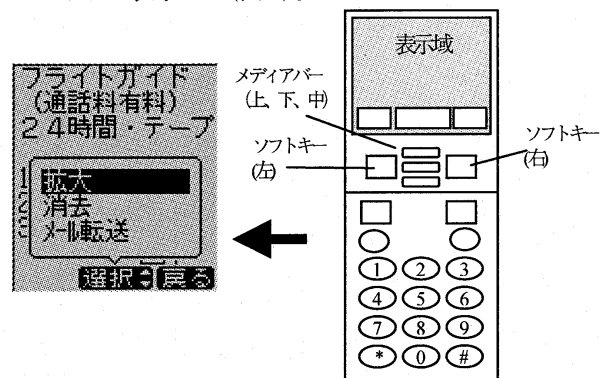


図2. ソフトキーと操作ガイダンス表示例

### 3.2. スクロールとフォーカス移動

アンカー、テキストフィールド、プッシュボタンなど、選択することにより特別な動作を行う表示要素をフォーカス候補と呼ぶ。パソコン上のブラウザのようなマウスによるスクロールや対象選択が可能な環境と異なり、多くの携帯電話ではキー操作のみでフォーカス候補間を移動/選択する操作を提供する。

今回開発した機種では、ハードウェアの制約からメディアバー上下キーのみで上下左右スクロールを実現しなければならないため、次のようなキーバインドとした。

- ・上キー短押し：左スクロール
- ・下キー短押し：右スクロール
- ・上キー長押し：上スクロール
- ・下キー長押し：下スクロール

また、スクロールとフォーカス移動を違和感なくスムーズに実行できるように、下記の動作仕様とした。

- ・表示されている次のフォーカス候補が存在する場合はスクロールを行わずにフォーカスの移動のみを行う。
- ・表示されている次のフォーカス候補が存在しない場合はスクロールを実行する。
- ・表示位置がコンテンツの右端/左端のときに右/左キーが押された場合は、1行下/上の左端/右端にページスクロールする。
- ・フォーカス移動が可能な場合は、押下されたキー方向に存在するフォーカス候補にフォーカス移動する。

#### 4. メモリ利用効率化

携帯電話市場は、低コスト/大量販売を前提としており、少しでも安価に機器を製造するためメモリサイズが抑えられる。このため、ブラウザ自体も小メモリの上に実装を行う必要がある。以下、我々が工夫を行った省メモリ化の内容を紹介する。

##### 4.1. メモリアロケーション

コンテンツを表示するためのデータを大きく分けると、「テキストデータ」：表示テキストやリンク先情報を記憶するためのデータと、「構造化データ」：データ間の関連付けを記憶するためのデータの二つに分類される。前者はタグ間に記述されるテキスト情報であり、後者は<DIVISION>、<TABLE>、<OL>などの構造化タグによって生成されるデータである。

この二種類のデータ間には次の性質がある。

##### 1. メモリ領域を相殺

コンテンツ記述において、テキストデータが多いほど構造化データは少なくなり、その逆も成立する。メモリ管理についても同じことが成り立つ。

##### 2. メモリバウンダリの違い

テキストデータは1バイトバウンダリで保存可能であるが、構造化データは4バイトバウンダリ(構造体でデータ管理)となる。

これらのことを踏まえ、次のようなメモリ割り当てを実装し領域を無駄なく利用できるようにした(図3)。

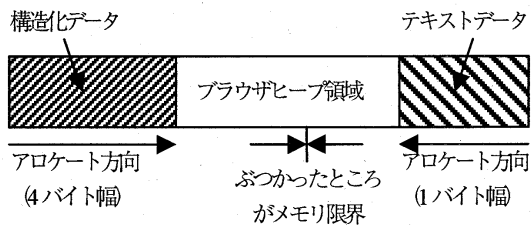


図3. ブラウザメモリアロケーション方式

##### 4.2. 再帰関数呼び出しを用いない実装

ブラウザでは与えられたコンテンツを字句解析し、それによって得られた情報を表示/操作する。これらの処理においては、解析木のような再帰的なデータ構造が生成される。

再帰的なデータ構造を処理する上では、関数の再帰呼び出しアルゴリズムを用いるが、スタックの必要サイズを見積もれなくなるおそれがある。

我々はアルゴリズムを工夫し、ブラウザの全処理において再帰関数呼び出しを用いずに実装を行った。

#### 5. まとめ

表示方法/操作方法/メモリ利用効率化の観点から、我々の組み込みブラウザの特長について説明した。

携帯電話のブラウザでは、表示領域が全角ベースで8文字×8行と非常に小さい。このためパソコン上の標準的なブラウザのような表示を行ったのでは、閲覧操作が大変になることから、極力横スクロールが発生しないよう表示方法を工夫した。

また、限られたキーによってコンテンツの操作を実行しなければならないことから、ソフトキーによる操作項目の選択や、短押し/長押しによるキーの区別を行った。更にキー操作のみでスクロールとフォーカス移動をスムーズに実行できるように、スクロール動作の実装を工夫した。

また、小メモリによって処理を実現する必要がある。そこでメモリアロケーションロジックを工夫し、メモリの無駄を極力抑える配慮を行った。また、再帰関数呼び出しを使わないことで、どのようなサイズ/タグ記述構造を持つコンテンツが与えられても決まったスタック容量でオーバーフローをなくすることが可能となった。