

携帯端末インターフェイスの可変方法の検討

羽藤 淳平[†] 佐々木 幹郎[‡]

三菱電機株式会社 情報技術総合研究所[‡]

1. 背景

現在の携帯情報端末、特に携帯電話は形状、サイズ等の制約により、実装可能なハードウェアボタン数を飛躍的に増加させる事は困難であり、その点もユーザーインターフェイスの自由度を下げる要因となっている。そのため、カテゴリ別のメニューに振り分ける、コマンド入力による機能呼出などのソフトウェアによって問題を解消している。しかし、年々携帯情報端末に実装される機能は増加しており、個々のユーザーの利用スタイルに合わせる事が難しくなっている。

本稿では上述の様なユーザーインターフェイスの構成変更権を携帯端末ユーザーに与える事で、インターフェイスをユーザーに合わせる方法を提供し、携帯情報端末の操作性を向上させる方法を検討する。

2. 現状の問題点

現在の携帯端末のユーザーインターフェイス構成は一般に従来通り製品開発段階で決定されている。開発段階でユーザーが頻繁に利用すると思われる機能の実行方法を簡易化する、各種機能の実行方法を統一化するなどの方法により、使いやすさ向上を図っているが、現在の携帯端末に実装される機能は多種多様であり、メーカーの想定する平均的なユーザーには適合するが、そこから外れるユーザーには不満の残る結果となる事がある。また、現在の携帯電話の多くは、待受け画面等の画像、時計表示の有無、フォントなどが変更できる様になっている。

しかし、変更できる要素は同様に開発段階で決定しており、より詳細に変更を加える事は出来ない。

3. 設計提案

以上の問題点は、ユーザーインターフェイス構成の殆どが開発段階で固定されてしまう事が一つの原因となっている。

そこで、本稿ではPC上のユーザーインターフェイス実装を参考にし、ユーザーインターフェイス構成において最低限必要となる機能のみを開発段階で決定し、それ以外の要素に関しては、可変とする事を提案する。

4. システム構成例

ここではシステム構成の一例を紹介する。

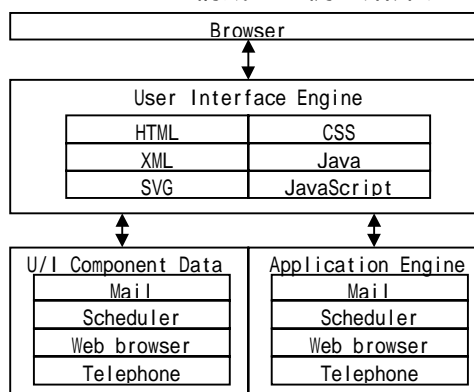


図1 システム構成例

図1はシステム構成例である。本システムでは一般ブラウザが実装しているHTML、CSS、Java等の機能を用いてユーザーインターフェイス構成を一般のWebページと同様の形式で表現する。

各アプリケーションはアプリケーションのエンジン部とユーザーインターフェイスとを独立させユーザーインターフェイス構成は構成情報として保持する。構成情報をユーザーが書き換える事でカスタマイズを可能としている。

The Method of Changeable Mobile Device Interface

[†] Jumpei Hato, Mikio Sasaki

[‡] Mitsubishi Electric Corporation

Information Technology R&D Center

次に、現在の携帯端末は項目選択やボタン押しといったイベントによって機能を出している。そこで、エンジン部にはイベントで呼出す機能の API を登録し、JavaScript 等のスクリプト言語でイベント監視命令を記述し、呼出す API をコールする方式を取る。図 2 はボタンを押すと Mail_Send API がコールされるタグの動作の流れを示した図である。onClick イベントを event listener が検知し、JavaScript が実行開始し、Mail_Send をコールする。その結果に応じて JavaScript で次の状態遷移を決定している。

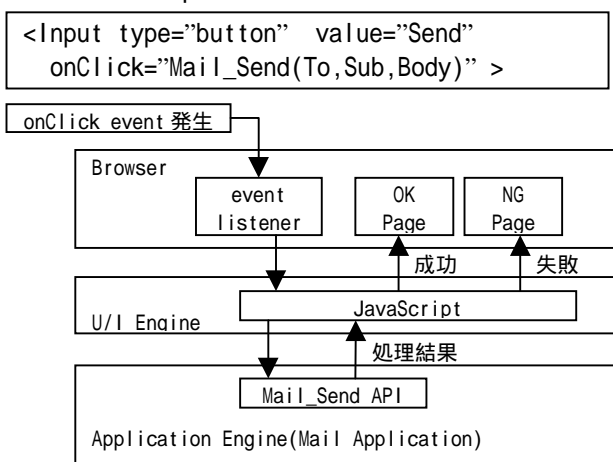


図 2 Application Engine API 呼出方法

また、ファイル削除処理の前に警告を表示等ユーザーに変更されないようにしたい場合もある。その場合は、Application Engine から User Interface Engine に、指定した User Interface Data の表示指示を行う API を用意する事で対処する。図 3 は例となるファイル削除の処理方法を表した図である。Data Folder Application の Delete_File 関数の処理中に User Interface Engine の API をコールする事で警告表示と終了表示を行っている。

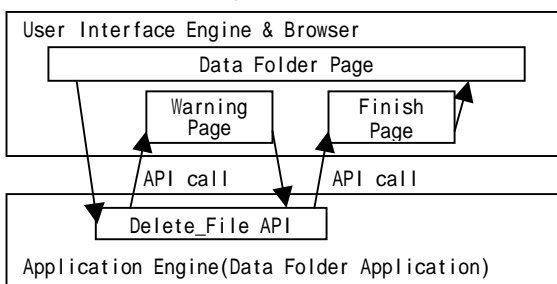


図 3 強制表示方法

5. その他の利点

ユーザーインターフェイスの構成を変更可能とする事その他の利点は以下の項目が考えられる。

5.1. 開発の効率化

アプリケーションの機能とユーザーインターフェイスを独立してモジュール化し、HTML などの既存技術を利用する事で、新たに仕様を理解する必要もなく、市販の効率的な開発環境を利用して開発できる。また、既存の Web ブラウザに実装されている機能を利用するため新規開発項目を最小限に抑える事が出来る。

5.2. ネットワーク通信

データ化する事で、インターネット等を通してユーザーに対して多種多様なユーザーインターフェイスデータを提供する事が可能となり、ユーザーは好みに合ったユーザーインターフェイスを選択する事が可能となる。

6. まとめ・今後の課題

本稿では、ユーザーインターフェイス設計思想の一つとして、ユーザーインターフェイスの構成変更をユーザーに許可する事による可能性を検討し、一つのシステム提案を行った。

今後の課題は、ユーザーインターフェイスを処理するために User Interface Engine 等の中間モジュールを実装しているため、従来方法の実装と比較し処理性能の検討が必要となる。

また、今回提案したシステムでは PC 上で一般的に実装されている機能を実装したブラウザを想定している。しかし、性能上の制約により完全実装は難しい。そして、携帯端末特有のボタン長押しや着信等のイベント対応が必要となる。

以上の問題を解決するため、User Interface Engine の持つ各機能を携帯端末に特化した必要十分な機能で軽量化を実現する仕様の検討が必要である。