

3X-1

携帯電話向けユーザインタフェース構築システム

朝日啓太

電気通信大学 大学院電気通信学研究科

1 はじめに

現在の携帯電話は、メール、Web ブラウザ、カメラ、ゲーム等通話以外の機能を多く持っており、頻繁に使用する機能はユーザによって異なっている。従って、ユーザの多様な使用方法に適合するためには、メニューを初めとするアプリケーション UI のカスタマイズが必須になると考えられる。

しかし、現在の UI はメーカーまたはアプリケーションの開発者が用意した限られた選択肢の中でカスタマイズを行うため自由度が低い場合が多く、自らカスタマイズしたものと同等の UI を他機種で使うことができる保証が無い。

加えて、機種毎に UI が統一されておらず、機種変更や複数機種の同時使用時に混乱が生じるという問題もある。

本研究では、これらの問題を解決するため携帯電話のアプリケーション UI に関して、自由度の高いカスタマイズ及び機種間の互換性を実現する手法を考案し、独自にカスタマイズした UI を様々な機種で使用することを可能にするシステムを開発した。

2 設計方針

本システムで UI を表現するデータはクロスプラットフォーム性及びカスタマイズ可能な柔軟性が求められるので、XML で UI を記述するマークアップ言語 XML-based User-interface Language (XUL) を採用した。これにより専用の XUL エンジンを使用して XML で記述した UI を構築・実行することができるので、エンジンが動作するプラットフォームなら同じ UI が動作するという利点がある。代表的な XUL の例として、Mozilla XUL[1] や Java アプレットやモバイル機器での使用を想定した Thinlet[2] がある。

本システムではこの手法を利用し、携帯電話で動作する専用のエンジンが XUL データを解析し、UI を構築・アプリケーションを実行する。また、携帯電話上で XUL データを編集することで、自由度の高い UI のカスタマイズを可能にした。

システムの実装に当たっては、NTTDoCoMo の i アプリ (DoJa3.5 プロファイル [3]) を使用した。しかし、

携帯電話の内部データへのアクセスが許可されていないため携帯電話で使用できるメニューを実際に作成することは不可能である。従って、メニュー等の内部データにアクセスするアプリケーションに関しては仮想的に実現した。

3 システム構成

本システムは XUL データ、携帯電話のアプリケーションとして動作する XUL エンジン及び XUL エディタから構成される。

XUL エンジンは、ネットワーク上や携帯電話に保存されている XUL データを解析し、UI を構築してアプリケーションを実行する機能を持つ。

XUL エディタは、実際の UI を見ながらコンポーネントの位置やサイズを変更することや、XUL データを直に編集することで更に複雑な UI のカスタマイズをする機能を持つ。

以下に本システムの使用概念図を示す。

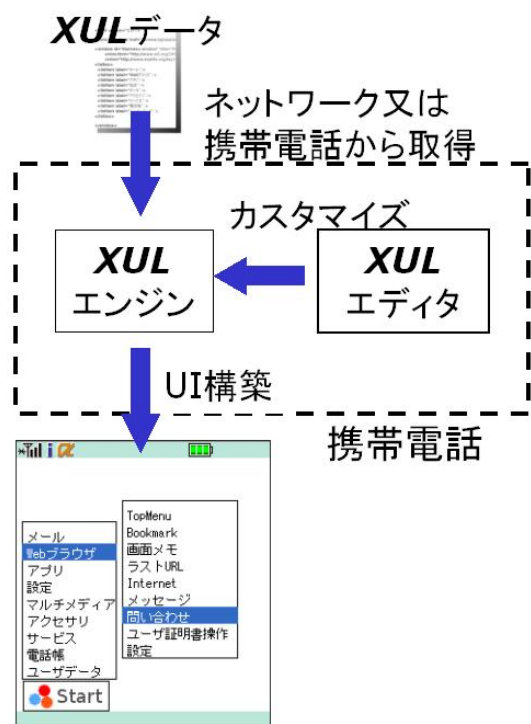


図 1: 使用概念図

3.1 XUL データ

本システムの XUL の仕様は MozillaXUL をベースにし、容易に自由度の高いカスタマイズが行えるよう変更を加えた独自形式である。後述の「外観」に関しては図形描画のために Scalable Vector Graphics(SVG)[4] のサブセットを使用した。

XUL で記述されたアプリケーションの UI は、XUL エンジンにより構築されるため、エンジンが動くプラットフォームであればほぼ同一の UI を提供することが可能である。使用可能な UI コンポーネントは携帯電話でよく使われているものを選択した。

本システムで使用している形式の特徴は、UI を構造および振る舞いを記述する「モデル」、コンポーネントの配置方法を記述する「レイアウト」、UI コンポーネントの見た目を記述する「外観」に分離し記述する点である。

例えば、携帯電話のメニューを本方式で記述する場合、それぞれ以下のような記述になる。

「モデル」メニューの構造と項目を選択したときの動作

例：ルートメニューには、電話帳、メール、設定... のメニュー項目があり、メールを選択するとメールのメニューが表示される... 等。

「レイアウト」各メニュー項目の配置方法

例：ルートメニューはボタンを格子状に並べ、メールのメニューはリストにする... 等。

「外観」各メニュー項目やボタンの見た目、背景の画像等

3.2 カスタマイズ

カスタマイズは PC 等で直接 XUL データを編集するか、携帯電話上の XUL エディタを使って行う。

カスタマイズの際には、配置を変える場合は「レイアウト」部分を変更、UI コンポーネントの見た目を変える場合には「外観」部分を変更することでカスタマイズを行う。「モデル」はカスタマイズの対象とならない。

レイアウトのカスタマイズは、個々の UI コンポーネントの位置・サイズを自由に変更することや、リスト形式のメニューをボタンを格子状に並べたメニューにしたり、Windows のスタートメニューのように複数階層を同時表示する方式に変更するといったコンポーネントの配置方法の変更も可能である。

外観のカスタマイズは、個々のコンポーネントの外観を任意の図形（線・矩形・円・楕円・多角形）や画像・テキストで自由に描画することができる。

図 2 に携帯電話のルートメニューを格子状に並べたボタンにし、メールのメニューをリストにして複数階層のメニューを同時に表示しているカスタマイズ例を示す。



図 2: カスタマイズ例

3.3 アプリケーション

現在の実装では、仮想メニュー、メモ帳といった単純なアプリケーションは全て XUL データのみで作成することができる。しかし、複雑な処理が行われるアプリケーションについては、本システムでは処理しきれないためロジックを Java で記述し、配布ファイルに追加しなければならない。これは、XUL 中に JavaScript による処理を埋め込むことで解決可能であると考えられる。

4 まとめ

本研究では携帯電話のアプリケーション UI に関して自由度の高いカスタマイズ・機種間の互換性を実現する手法を考案し、独自にカスタマイズした UI を様々な機種でを使用することを可能にするシステムを開発した。

今後の課題としては、本システムの有用性の評価や他キャリアの携帯電話での実装、本システムで使用する XUL の仕様の改良などがあげられる。

参考文献

- [1] Mozilla XUL
<http://www.mozilla.org/>
- [2] Thinlet
<http://www.thinlet.com/>
- [3] DoJa3.5 プロファイル
http://www.nttdocomo.co.jp/p_s/imode/make/java/
- [4] Scalable Vector Graphics
<http://www.w3.org/Graphics/SVG/>