

# 組み込みソフトウェア UI カスタマイズ機構の試作

島田将行 種村嘉高 広沢昌司 音川英之

シャープ株式会社 研究開発本部 ソフトウェア開発センター

## 1. はじめに

近年、家電製品や携帯機器の開発コストにおいて、ソフトウェア開発コストの占める割合が増加している。特に、ユーザー・インターフェイス (以下 UI) に関わる部分は、その多くが類似する要素で構成されているにもかかわらず、微調整や再開発に要するコストが少なくない。

筆者らの所属する研究グループでは、これまで家電向けネットワークサービスのプラットフォームである Ex システムの開発を行ってきた。

部品化された機能を結合させてアプリケーションを実現することができる Ex システムの特徴を、組み込みソフトウェアの UI 部分に適用することで、UI の部品化、再利用性の向上を実現することができると考え、近年モバイル向けプラットフォームとして注目されている Android 上に試作した。

本稿では、この試作を用いた UI の部品化の仕組みと、当該システムに適合させるためのコストについて述べ、その有効性について説明する。

## 2. 試作の概要

### 2.1 Ex システムの概要

Ex システムは、ネットワーク上のサービスと端末独自の機能とを動的に結合させる点に特徴があり、これまでにネットワークサービスのプラットフォームとして用いられてきた。

Ex システムは OMA (Open Mobile Alliance) の MMS Message Template Specification で規定される XML 記述 (MTD 記述という) を独自拡張した XML 記述を解釈、実行し、端末機能とネットワーク上のリソースとを緩やかに結合するためのエンジンや、UI フレームワークの貧弱な組み込み機器に対して、端末のハードウェア (以下 HW) 機能を生かした描画機能を提供するレンダリング・エンジンなどを持っている [1]。

端末固有の HW 機能とネットワークサービスとは疎結合であり、実行時にコンテンツによって結合される。したがって、画像展開機能や、赤外線通信機能、動画再生機能などといった新しい HW 機能を生かしたサービスを、HW 機能の製品搭載後、短期間で提供することができる。

### 2.2 再利用の重要性

近年では、ソフトウェア開発コストの削減などのため、ソフトウェア開発のアウトソース、特に、オフショア開発の採用が進んでいる。

しかしながら、個別の機能や製品を実現することを目的に開発されたソフトウェアは結合度や凝集度などの考慮がなされておらず、再利用することが難しい。

このようなソフトウェアを再利用するときには、それをブラックボックスとして扱い、ほとんど手を加えずに再利用できなければ、返って再利用コストが高くなってしまう可能性がある。しかし、個別の開発案件に抽象化や汎用化を求めすぎると、目指すパフォーマンスが実現できなかったり、開発コストがかさんでしまったりする。

### 2.3 Android 環境への適用

今回我々は、近年急速に普及が進んでいる Android の環境に Ex システムを移植し、改良を加えた。Android は PC 上のエミュレータや GUI エディタなどといった開発ツールが充実しており、アプリケーションや UI の開発に対する敷居が低く、従来の組み込みソフトウェアのプラットフォームに比べて、委託開発しやすい。

モジュールを緩やかに結合させられる Ex システムの特徴を生かすことで、再利用性や保守性の作り込みが困難な外製ソフトウェアの再利用を促進することができると思われる。

[2]に述べられているように、組み込みソフトウェアの UI では、「一つの画面で行える操作を限定し、複数の画面を切り替えることで多くの捜査を可能とするのが一般的」であった。従来の Ex システムもこのアプローチ度同様に、細切れにした UI の状態遷移モデルによってアプリケーションを構成していた。

しかし、Android 環境のように UI 部品や画面を構築するためのライブラリが充実している環境においては、すべての構成要素を統一された方法で設計、実装するのではなく、自立的に動作する個々の部品を緩やかに連携させることが重要であると考えた。また、汎用化を求めない点で、外製ソフトウェアの再利用にも適している。

### 2.3 UI の部品化

今回の試作では、UI を構成する部品を以下の 2 つのレベルでモデル化し、結合させる試作を行った。

- A) アプリケーション間の結合
- B) 画面を構成する部品の結合

Android の UI は Activity と呼ばれる画面を単位として構成されており、アプリケーションや UI を持つ機能モジュールは Activity の集合で実装される。これら Activity 間の遷移はソースコードにハードコーディングされる。

Ex システムの仕組みを使うと、このような Activity 集合間の遷移をソースコードから独立して記述し、動的に適用することができる。そのため、これらを互いに疎結合にし、再利用性を高めることができる。

Prototyping of UI customization module for embedded software  
Masayuki Shimada, Yoshitaka Tanemura, Masashi Hirose, Hideyuki Otokawa  
Software Development Center, Corporate Research and Development Group, Sharp Corporation

今回の試作では、このような遷移を MTD 記述によって表現し、遷移先を決定するための条件変数の管理や、上記 Activity 集合の入出力パラメータを管理する機構などを Android 上に実装した (図 1)。

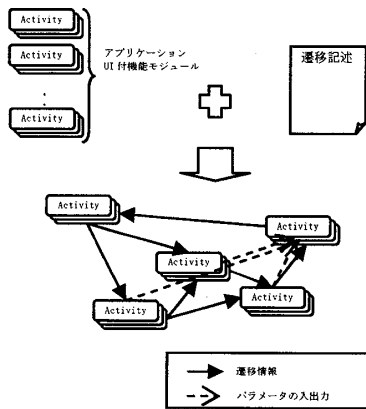


図 1 画面遷移の記述例

また、1つの画面 (Activity) を構成する部品の再利用を考えたとき、部品ごとに外部公開する状態のリストを定義し、2つの部品間で、状態どうしの対応付けを外部リソースに記述するようにした。これをリンク記述という。図 2にリンク記述の例を示す。

リンク記述の要素は、ある部品が状態遷移したとき、別の部品を対応する状態に遷移させることを意味する。

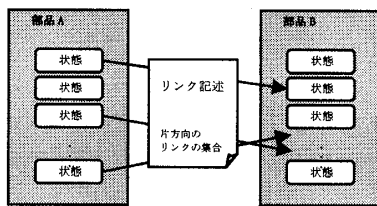


図 2 リンク記述の例

このようなモデルによる部品が連携して遷移処理を行えるように、リンク記述を処理し、遷移を伝播させるモジュールを実装した。このモジュールをリンクマネージャーという (以降、LM)。ある部品が外部公開される状態に遷移したときには、この LM に通知することとし、LM はリンク記述に従って、逐次部品に遷移を指示する (図 3)。

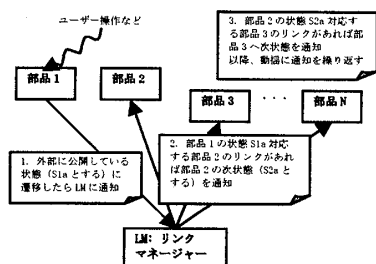


図 3 遷移伝播処理の例

### 3. 評価

本試作は特にプロダクトライン開発を意識したものではないが、[3]に倣い、再利用コストとソフトウェア資産化コストについて、ソースコード行数で評価することを検討した。

Ex システムでは、部品の依存関係をソースコードではなく外部リソースに記述するため、再利用コストはこの外部リソースの作成コストとなる。しかし、現段階ではこのサンプルが少なく、評価が困難であったため、ソフトウェア資産化コストについてのみ評価した。

今回部品化した既存モジュールの多くは、C 言語によって特定製品のプラットフォームに向けて実装されていたため、Android 環境への移植コストが発生している。それに加えて、既存のモジュールを Ex システムに適合させるために行った修正のコストが必要であった。

Ex システムへ適用させるコストを全体のコードサイズ、および、Android 環境への移植コストと比較したところ、次のような結果であった。

	モジュール A	モジュール B
全体コード行数 (C / Java)	420KLOC / 25KLOC	210KLOC / 5.5KLOC
Android への移植に要した修正コード行数 (C / Java)	15KLOC / 25KLOC	8KLOC / 5.5KLOC
本試作適用のための修正コード行数 (Java)	0.8KLOC	0.1KLOC

表 1 Ex システム適合の修正コスト

表 1の結果が示すように、本試作に適合させるための修正コストは、Android への移植コストや全体コード数に比べて十分に小さく、Ex システムを用いることにより、UI 部品の資産化が低コストで行えることが確認できた。

### 4. まとめ

今回の試作により、Ex システムが UI 部品を低コストで資産化するためのフレームワークとして有効であることがわかった。

今後は、外部リソースに記述することとした部分の開発コスト、即ち再利用コストを評価する必要がある。また、部品化の粒度、モデル化が適切であるかを検証するために、再利用率についてもあわせて追跡評価していく必要があると考えている。

### 5. 参考文献

[1] 中西 正洋, 坂倉 健太郎, 財満 博昭, 畑山 尚毅, 小野 修一郎, 尾上 孝雄, 組込み機器上で Web サービスを扱うプラットフォームの構築, 情報処理学会研究報告, DD 34, 83-88, 2007

[2] 豊岡明, 小中裕喜, UI 設計ツール NINA による組み込み機器 UI ソフトウェアの再利用性向上, 情報処理学会第 71 回全国大会 1B-6, 2009

[3] 野中 誠, ソフトウェアプロダクトライン開発のマネジメント: 課題と技法, 情報処理学会 Vol. 50 No. 4, 2009