

# Mobicom: GUI モバイルアプリケーションの部分的なマイグレーションを支援するフレームワーク

氏名<sup>†</sup> 福田 浩章 氏名<sup>†</sup> 山本 喜一  
 所属<sup>†</sup> 慶應義塾大学 所属<sup>†</sup> 慶應義塾大学

## 1. はじめに

近年コンピュータの小型化，低価格化に伴い，複数のコンピュータ(以下端末と呼ぶ)を使用して作業を継続する機会も増え，従来のノート型端末だけではなく，携帯電話や PDA といった携帯端末でも多数のアプリケーションを利用できるようになった．また，無線 LAN などのネットワークインフラも整備され，至る所でネットワーク接続が可能になっている．しかし，我々が利用するアプリケーションの多くは単一の端末で処理が完結することを前提にしており，仮にネットワークにつながった複数の端末で継続して作業を行う場合でも，一度データを保存して転送し，別の端末でアプリケーションを起動してデータを読み込む必要がある．そのため，データの種類に適したアプリケーションの有無によって，使用できる端末は制限される．また，近年のアプリケーションは多機能化し，ユーザが実際に使用するのはその中の一部であることも少なくない．

そこで本論文では，データと状態を保持したままネットワークを介してユーザが必要な機能だけを移動するアプリケーションを提案し，そのために必要となる部分的なマイグレーションを実現するフレームワーク，Mobicom の提案および実装を行う．本論文では導入例として GUI アプリケーションを対象とし，Mobicom を用いてアプリケーションを構築し，動作を確認する．

## 2. GUI アプリケーションの特徴と問題点

図 1 に GUI アプリケーションの外観(a)とそれを構成するオブジェクトの関係(b)を示す．GUI アプリケーションの実行形式はイベント駆動型であり，一般的に MVC モデルで構築される．図 1 では，ユーザが(V1)を操作して発生したイベン

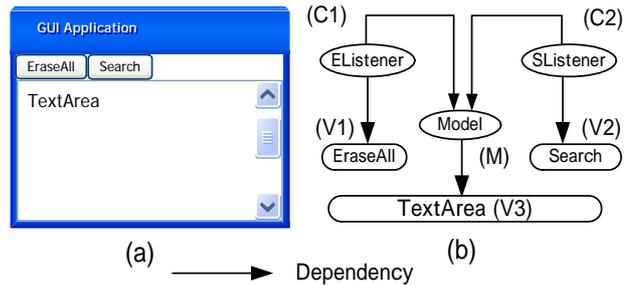


図 1 GUI アプリケーションの外観と構造

トを(C1)が処理し，(M)を変更することで(V3)が変化する．アプリケーションが図 1 の(b)で示す構造の場合，個々の機能は(C1)と(C2)で実現され，それらは(M)への参照があればよくお互いの参照は不要である．このように，各オブジェクトにはプログラムには現れない依存の方向(以下オブジェクトの依存関係と呼ぶ)がある．しかし，既存のモバイルエージェントシステム[1][2]などを利用して図 1 のアプリケーションを移動させる場合，(C1)と(C2)には(M)を介した参照関係があるためすべてのオブジェクト，つまりアプリケーション全体を移動せざるを得ない．

## 3. アプローチ

本研究では，2 節で述べたオブジェクトの依存関係を利用し，Java 言語を用いて部分的なマイグレーションを実現する．そのためにアプリケーションの実行環境(以下コアシステムと呼ぶ)と，依存関係を記述する API(以下 MobicomAPI と呼ぶ)を提供し，次のようにして部分的なマイグレーションを実現する．

### 3.1 部分的なマイグレーションの実現法

図 1 で示した構造のアプリケーションを移動する方法を図 2 に示す．図 2 では，アプリケーションの移動元を Home，移動先を Remote とし，ユーザはリモートで Home から(C2)を移動する．また，Home と Remote で動作するコアシステムを Home コア，Remote コアとする．

Mobicom: A framework for supporting partial migration of GUI applications  
<sup>†</sup> Keio University

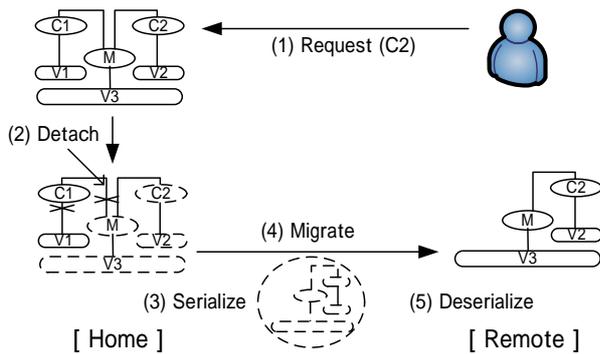


図2 マイグレーションの実現法

(1)ユーザは Remote で Home に(C2)の取得要求を出す。(2)Home コアはオブジェクトの依存関係を基に、(C2)が直接依存しない(V1)と(C1)を分離する。(3)Home コアは Java の直列化機構を用いてアプリケーションをデータ化し、(4)Remote コアに転送する。(5)Remote コアはデータ化したアプリケーションを復元する。

以上の処理によって、アプリケーションを部分的に移動することができる。また、ユーザが Remote でアプリケーションを使用中に(C1)が必要な場合、同様の方法で Home から Remote に移動した後、実行中のアプリケーションに再結合する。

### 3.2 MobicomAPI

GUI アプリケーション開発では、パネルやボタン、イベントリスナーなどを適切なメソッドを実行して結合する部分(以下結合部分と呼ぶ)がある。結合とはオブジェクトの参照を保持するということであり、オブジェクトの依存関係に深く関係する。そこで Mobicom では、結合部分のオブジェクト生成やメソッド実行を利用し、開発者がそれらの処理に MobicomAPI を使用することで、依存関係を定義する。そして、コアシステムが結合部分を実行するとき、開発者が定義した依存関係を認識する。コアシステムは、オブジェクトを移動するときそれが依存するオブジェクトが移動していなければ移動させる。依存関係によって、移動するオブジェクトと、それが機能するために必要なオブジェクトを最小単位で移動することができる。これを実現するために、開発者はアプリケーション本来の機能を実装し、結合部分の実装に MobicomAPI を使用して依存関係を定義するだけでよく、それ以外の処理はコアシステムが行う。ただし、3.1 で述べたように、移動できるオブジェクトは直列

化可能なオブジェクトに限られる。

### 4. Mobicom を使用したアプリケーション

Mobicom を使用したアプリケーションとして、文字の色、フォント、サイズの変更、位置や太さを変更できるエディタを作成した。

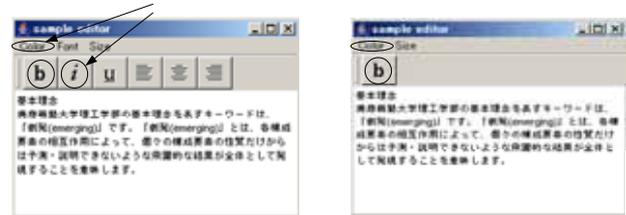


図3 Mobicom を使用したエディタ

図3の(a)は Home ですべての機能が利用できるエディタで、Remote で文字の色と太さを変える機能(オブジェクト)を移動したのが図3の(b)である。このように、Mobicom によってアプリケーションの状態とデータを保持したまま、必要な機能だけを移動して利用することができる。

### 5. まとめと今後の課題

本論文では、データと状態を保持したままネットワークを介してユーザが必要な機能だけを移動するアプリケーションを提案し、そのために必要となる部分的なマイグレーションを実現するフレームワーク、Mobicom の提案および実装を行った。Mobicom によってデータやアプリケーションに制約されずに、ネットワークにつながった端末で継続した作業を行うことが可能であり、部分的にマイグレートすることで、ユーザが実際に使う機能だけを効率よく移動することができる。

今後の課題として、本論文では述べていないが、ユーザが必要なオブジェクトを指定して移動するための分かりやすいインタフェースを提供する必要がある。

### 参考文献

- [1] Lange D. and Oshima M. "Programming and Deploying Java Mobile Agents with Aglets", Addison Wesley, 1998.
- [2] I. Satoh, "A Mobile Agent-Based Framework for Active Networks", In Proceedings of IEEE Systems, Man, and Cybernetics Conference (SMC 99), October, 1999.