

5V-4

異機種携帯電話間で連携可能な写真共有アプリケーションの構築

竹淵 瑛一[†] 平木 浩二朗[†] 三角 甫[†] 濱口 篤士[†] 庄司 諒平[†] 成田 真孝[†]
 加藤 雄輝^{††} 井瀨 裕太^{††} 原澤 有加^{††} 小川 輝樹^{††} 織田 喜雄^{††} 松林 静輝^{††}
 白石 陽^{††} 新美 礼彦^{††} 高橋 修^{††} 渥美 幸雄^{†††} 速水 治夫[†]

神奈川工科大学情報学部情報メディア学科[†]

公立はこだて未来大学システム情報科学部^{††} 専修大学経営学部^{†††}

1 はじめに

神奈川工科大学（以下、神奈工）、公立はこだて未来大学（未来大）、専修大学（専大）、函館工業高等専門学校（高専）の学生の共同プロジェクトで携帯アプリケーションを開発した。4校で基本コンセプト、サービス仕様を決めた後、専大はビジネスモデルを検討し、他の理系3校がプログラムを開発する。今年度は本稿で報告する「フォトリング」と別途報告する「Quecity」を開発した。

フォトリングは未来大、神奈工の2校で開発を行った。開発を行った携帯電話は au, docomo, iPhone, Android, Windows Mobile の5機種である。5機種のアプリケーションは共通のサーバと通信を行い、写真を中心とした情報共有を行う。開発は各機種およびサーバ毎の班に分かれて行った。本稿では各機種のアプリケーションとサーバとの通信に主眼を置いて報告を行う。

2 プログラム開発

2.1 設計

本システムでは、各機種とサーバは以下のような方式で通信を行う。

- 通信はブラウザ上でデータの確認を行うため、直接のパケット通信ではなく、PHPの標準文字出力（print文）で行う。
- 5機種共通の通信方式を取るため、文字のエンコードをUTF-8、画像の変換をBase64で行う。
- 通信するデータの内容・順番を厳守している。

通信において、PHPの標準文字出力を利用した理由は、まず一つに一部の機種が（サーバ班の）手元にならないため、正常に表示されたかどうかの確認が取りづらいため、と言う問題点を回避するためである。また、パケット通信よりもブラウザ上に表示された文字を取得するほうがはるかに容易で、互いに表示結果の確認が取りやすい。

通信方式については当初、docomoではShift-JISの通信にしか対応していないと報告があり、Shift-JISによ

て通信を行っていた。しかし、AndroidのJavaではデフォルトの文字コードがUTF-8であり、これをShift-JISに変換すると一部文字化けするという問題が発生した。docomoについて詳しく調べると受信のみはUTF-8での通信が可能であると分かったため、全体の通信方式をUTF-8に変更した。これにより、Androidで起きた一部の文字化けが解消され、正常に通信できるようになった。iPhoneもShift-JISで通信を行っていたが、一部のコードをコメントアウトするだけでUTF-8に対応できた。

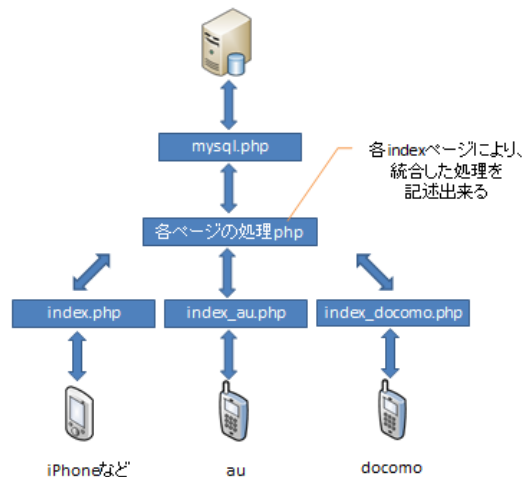


図1 各機種とプログラムの関係

また、auとdocomoではBase64の実装が間に合わず、画像データではなく画像のURLを送信することで対応を図った。当初、端末との窓口となるサーバ側のプログラムは機種共通のindex.phpであったが、共通仕様に合わせられない機種もあるため、index_au.phpやindex_docomo.phpなど、各機種対応の窓口となるプログラムを設けた（図1）。これにより各機種の事情をサーバ側で吸収することで、それぞれの仕様の対応・非対応の差を無くし、あたかも同じ情報を扱っていると思わせることができるようになった。

2.2 実装

プログラムの実装では、サーバにおいてデータベースを構築し、各機種が必要に応じて情報を問い合わせる形式を採った。従って、機種ごとのアプリケーションの動作はデータベース内で参照した情報を基に行っている。これにより、問い合わせた情報によって機種ごとに独自の動作をさせることができるようになっていく。これはスマートフォンである iPhone や Android など、スマートフォンらしいインターフェースの実装を考慮した結果である。

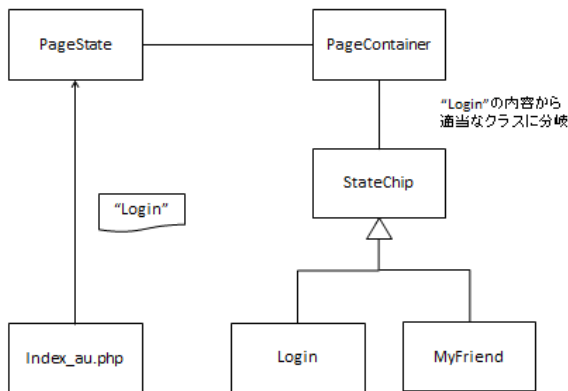


図2 サーバで実装したクラス図

サーバから返す情報については、各班の取り決めによって共通の画面遷移図が作成され、その画面によって表示される結果を基に仕様の策定を行った。さらに、サーバのシステムにおいて Factory Method Pattern を導入することで、早期からモックデータを利用し、仕様変更・追加などに対して柔軟に対応できるようにした(図2)。

2.3 実装における問題点

実装中の仕様変更、テストの不完全さが主な問題点であった。加えて、au, docomo, Windows Mobile は未来大学にあり、iPhone, Android 及びサーバは神奈工にあるため、情報伝達に問題が生じた。

2.1 節で示した通り、実装途中の仕様変更により、それまで正常であった部分に仕様変更を強いることとなった。開発の終盤であったため、各開発班に大きな負担を与えてしまった。また、Base64 での通信は可能であったが、技術的に困難なため、代わりに画像の転送方法を URL の直接指定にて行う機種があった。これにより、サーバ上にそれぞれ Base64 形式、jpeg 形式と重複したデータを保存するなどの問題が発生した。

これらは設計段階でのテスト項目の少なさに問題があると思われる。実際に行ったテストは、実環境を模

したのではなく、画像の Base64 形式の通信が行えるかの確認だけであった。そのため、実環境でのテストを行った際に文字化けの問題や、サーバで利用している Base64 の仕様と iPhone で利用している Base64 の仕様が異なるなどの問題が発生した。この問題により、サーバの通信の仕様では画面遷移図を元に、各開発班と相談しつつ、同時に作成しながら決定した。

また、神奈工に存在するサーバを未来大学より利用したため、上述の通信の仕様を各開発班に伝達する際、wiki のみで行ったため、各開発班が発見した問題に対応するのに時間を要した。

2.4 問題点の解決

開発中の仕様変更では、各開発班のスケジュールの見直しを行い、テストの不完全さはサーバ側による動作テストを多く実施することで問題を解決することができた。

特に情報伝達については、報告、連絡、相談を重ねる毎に、要点を押さえられるようになった。開発の序盤では、通信に関するエラーが発生した場合に、情報交換をせずに各班で問題解決にあたることが多く、端末かサーバのエラーなのか確認に時間を取るが多かった。そこで、端末側での通信エラーが発生した場合、早急にサーバ班と Skype や電話などで相談するよう心がけた。その結果、サーバと端末双方から問題点を絞り出すことで、問題解決にかかる時間が短縮した。

上記の問題解決の過程から、分散環境での開発では、情報交換の重要性を学んだ。今回は、週1回の定例会時に相談する方法を取り、その時間でしか報告を行わなかったために情報伝達に遅れが発生した。この問題点を受け、各開発班は問題が発生した時点でサーバとの連絡を密に取ることによって、開発の遅れを取り戻すことができた。

3 終わりに

本稿は異機種携帯電話間で連携可能な写真共有アプリケーションのシステムの開発・実装について述べた。

今後の課題として、バグの対応や検証は双方の時間を消費してしまうため、相談前のチェック項目などを作成し、相談する必要があるか事前に評価する方法の検討を進める。

参考文献

[1] 公立はこだて未来大学システム情報科学部[†] 専修大学経営学部: ケータイアプリによる体感型 RPG を用いた地域活性化サービスの提案: Quacity