

Desktop RIA フレームワークの開発

モバイル・クラウド時代の新しいフレームワーク

1. 背景

近年便利なインターネットアプリケーションが多数出ているが、ネットワーク遅延や切断による不安感がある。またモバイル環境においても、インターネットを介したアプリケーションは”サクサク動く”体験を与えられるものは少ない。このような問題に対処するため、インターネットアプリケーションにおいてローカルリソースを適切に扱えるように変更した Desktop Rich Internet Application (Desktop RIA) 用の基盤技術が発達してきた。

しかし、既存のインターネットアプリケーションフレームワークはこれらの基盤技術のメリットを生かしていないため、サービスにもメリットが反映されないことが多い。そのため Desktop RIA を開発するには大きなコストがかかる。

また、Desktop RIA はサーバサイドにデータベースのみを必要とする場合が多いため、近年流行しているクラウドデータベースを用いることで安価にスケーラビリティを持つサービスを開発することができる可能性がある。ただし、クラウドデータベースの利用には高度な知識が必要であるため、開発に大きなコストがかかる。

このようにローカルストレージとクラウドデータベースを利用した Desktop RIA の開発には現状、多大なコストがかかる問題がある。

2. 目的

本プロジェクトでは、開発者が容易にローカルストレージやクラウドデータベースを生かした Desktop RIA を構築できるような新しいフレームワークを開発することを目的とする。

3. 開発の内容

Desktop RIA 用フレームワークとして、主に二つの機能を実装した。それぞれ説明する。最後にデモアプリケーションについて説明する。これらのフレームワークは、Desktop RIA 実行基盤である Adobe Flex/Air を用いて作成した。

Active Sync モジュール

Active Sync モジュールは、既存の Active Record パターンを拡張したモジュールであり、ローカルストレージとクラウドデータベースの暗黙的な同期を行う O/R マッパーである。

従来のインターネットアプリケーションやデスクトップアプリケーションでは、データの永続化先となるデータベースは一つだけだったが、Desktop RIA ではローカルストレージとインターネットコネクションの背後にあるサーバデータベースの二つになる。そのため、既存のフレームワークが持つ O/R マッピングのパターンはそのまま適用することができない（図 1）。

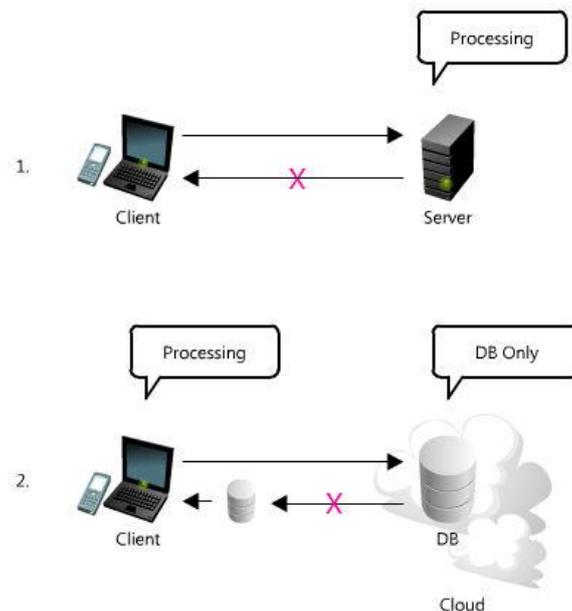


図 1 永続化方法の違い

そこで、Active Record パターンを拡張した新しいパターンである Active Sync パターンを設計し、提案フレームワーク上に実装した。サーバサイドには Google App Engine と Amazon SimpleDB に接続するモジュールを用意した。

このモジュールによって、開発者からは複雑になりやすいオフライン処理への対応と、クラウドデータベースへの接続を隠ぺいすることができた。

Async DI モジュール

Async DI モジュールは、大規模化したアプリケーションにおいて複雑化しやすいイベントによる依存関係を XML 形式で外部に記述することができるモジュールである。Flex 標準や既存のフレームワークでは、Model-View-Controller (MVC) モデルを実装する場合に、イベントによるオブジェクト間の依存関係が肥大化し、複雑になる傾向がある。

そこで Flex のもつ MXML 形式 (XML ライクな書式) でのプログラミングを生かして、イ

ベントによる依存関係を外部に記述できるようにし、可読性の向上とモジュール化の促進を進めることで、アプリケーションの全体構造を把握しやすくした。
このようにすることで、大規模な Desktop RIA 開発を素早く円滑に進めることができるようにした。

デモアプリケーション

提案フレームワークの機能を概観することができるように、デモアプリケーションを二つ作成した。一つが Active Sync 機能をデモする単純なメモアプリケーション（図 2 右）であり、もう一つは、Async DI 機能をデモできるようにある程度、規模が大きく複雑な音楽検索・共有アプリケーションである（図 2 左）。

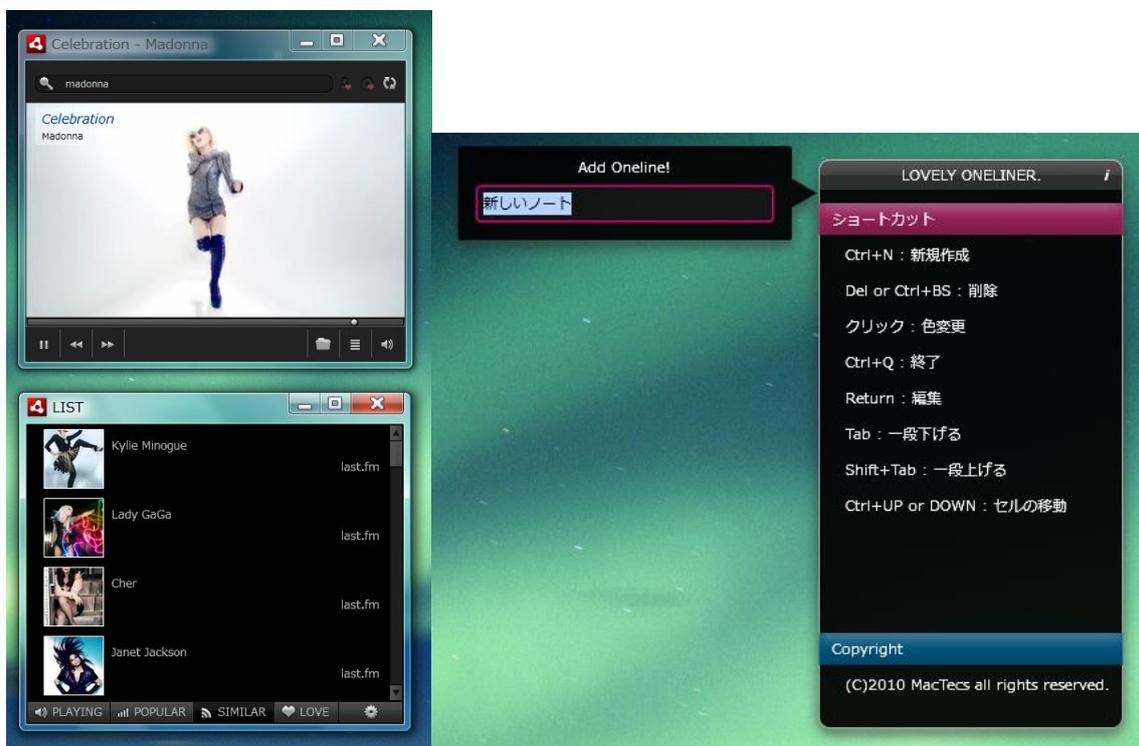


図 2：(左) 音楽検索・共有アプリケーション (右) メモアプリケーション

4. 従来の技術との相違

既存の Flex フレームワーク [PureMVC, Swiz, Mate など] は、インターネットアプリケーションの焼増しであり、Desktop RIA に適したローカルデータベースとサーバデータベースの同期を含んだプログラミングパターンを実装しているものはない。また Active Sync パターン自体も、本プロジェクトで提案した新しい概念である。その点において開発したフレームワークは既存のフレームワークに比べて優位性があると言える。

5. 期待される効果

本プロジェクトで開発したフレームワークの利用ターゲットは、Desktop RIA 開発者である。本フレームワークを用いれば、Desktop RIA を通常の RIA と同じように開発できるのに加え、既存の RIA を即座に Desktop RIA に変更することも可能となる。さらにサーバサイドのデータベースにスケーラビリティの高いクラウドデータベースを用いることも可能である。

このような特徴から、二つの効果が期待できる。一つ目は Desktop RIA の開発サイクルのスピードが向上し、多くの便利なサービスが社会に出ることが期待される。二つ目は、クラウドデータベースの利用が簡便になることで、電子機器の効率的な利用が促進され、グリーン IT に貢献できることが期待される。

6. 普及の見通し

今回の開発プロジェクトには普及に関する開発内容を含んでいなかったため、フレームワークの普及は今後の課題として残されている。そのため、今後フレームワークを普及するためにドキュメントの整備やデモやチュートリアルを整備などを行う予定である。特にフレームワークというものは、キラーアプリケーションと共に認知度が高まるものであるため、まずデモアプリケーションとして作成した音楽検索・共有アプリケーションやメモアプリケーションを数万～10 万人程度にインストールしてもらう必要がある（Adobe Air のアプリケーション配布環境は優れているため難しい目標設定ではないと考えられる）。そのうえで、フレームワークの普及を図ることとする。

7. クリエータ名

柴山直樹（東京大学大学院情報理工学系研究科）