

スマートデバイス向けアプリケーション開発への Functional Reactive Programmingの適用評価

中村 秀樹

開発における問題点

スマートデバイス向けアプリケーションは、様々なセンサーデバイスを活用したり、サーバとのやり取りが増え、今後複雑さが増大していくと予想される。一方で、このようなアプリケーション開発には、開発費・開発期間はあまりかけられず、開発容易性、保守性、拡張性の確保が重要になりつつある。

手法・ツールの適用による解決

スマートデバイス向けアプリケーション開発に、Functional Reactive Programming(FRP)を適用したことを想定し、開発容易性、保守性、拡張性について検証する。スマートデバイス向けアプリケーション開発に適用できるかという観点で、FRPの中でもHTML/JavaScript/CSSで出力するElmを用いて検証した。

スマートデバイス向けアプリの複雑性

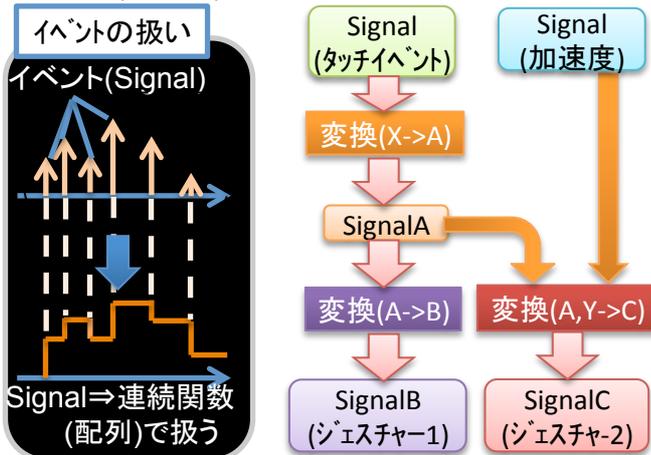


スマートデバイス向けアプリは、GUI処理、様々なセンサー処理、サーバ通信処理が複雑に絡み合い、また、非同期型APIの提供により、複雑な非同期プログラムになる。非同期プログラムは、処理が分断化され、見通しが悪く、保守性が低いという問題がある。

呼出しとハンドラ処理で分断化
⇒見通し悪く、保守性低下

FRP

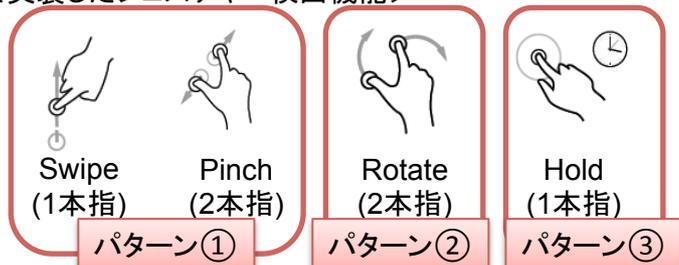
入力(イベント)から出力への変換ロジックを記述するプログラミングパラダイム。イベントをSignalとして扱い、変換(合成等)が容易。



検証

タッチイベントからジェスチャー検出を行うプログラムに対して、実際にFRP(実装言語は「Elm」)を適用した開発を行い、効果を検証した。検証を通して実装パターンが①単純変換、②状態参照、③タイミング相違の3つあることが判明した。

<実装したジェスチャー検出機能>



評価・まとめ

実装したジェスチャーのLOCを比較し、実装容易性を確認した。また、機能追加を通して、処理の分断化、既存コードへの影響がなく、保守性・拡張性が高いコードであることが確認できた。今後は、LOC以外の観点でのより詳細な実装容易性の評価、複雑な処理での検証を進めていく。

ジェスチャー	パターン	Elm	(参考)JavaScript※
Swipe	①	18行	60行(+233%)
Pinch	②	18行	49行(+172%)
Rotate	②	37行	58行(+57%)
Hold	③	31行	42行(+35%)

※JavaScriptライブラリ「Quo.js」から算出