

Web アプリケーション基盤と開発環境

— そろそろWeb 2.0 化しませんか? —

アブストラクト

1. 既存 Web アプリケーションを Web2.0 化したいけどできないを解決

Google Map に代表される操作性の良い Web アプリケーション、SaaS と呼ばれるサービス形態、また mixi に代表される SNS など、Web2.0 と呼ばれる技術が近年著しく発展している。そのような中で、企業向け Web アプリケーション開発においても「もっと使い勝手の良いものにしたい!」「もっとかっこよくしたい!」という顧客からの要望が増加している。しかしながら、実際に Web2.0 アプリケーションの開発手順、開発環境を考えると以下のような課題に直面し、開発コストが増大したり開発を断念したりする場合もあるだろう。

- (1) 業務アプリケーションに Web2.0 技術を適用するにあたり、機能、採用基準、メリット・デメリットなどが不明確である。
- (2) 既存の設計手法、開発手法をそのまま Web2.0 開発においても適用しており、Web2.0 特有の要素を反映しないまま進めている。

そこで当分科会では、これらの課題を解決するために、既存 Web アプリケーションを Web2.0 化する際の指針となる「Web2.0 適用/方式設計ガイドライン」を策定することを研究の目的とした。

2. 使える「Web2.0 適用/方式設計ガイドライン」の策定手順

実業務で使える「Web2.0 適用/方式設計ガイドライン」を策定するために、アプリケーション構築に関わる技術要素面に絞った観点から Web2.0 を掘り下げることにした。また、広義な解釈ができる Web2.0 の中でも「リッチユーザーインターフェイス」「非同期通信」「マッシュアップ」の3点に重点を置き、ガイドラインの適用範囲を次の通り定めた。

【対象者】

- ・顧客との商談において、Web2.0 を利用した Web アプリケーションシステムの提案を行う人
- ・Web2.0 を利用した Web アプリケーションシステム構築における要件定義を行う人
- ・Web2.0 アプリケーション方式設計者

【利用シーン】

- ・企画工程/要件定義工程/方式設計工程

図表 1 に示す研究手順について以下に説明する。

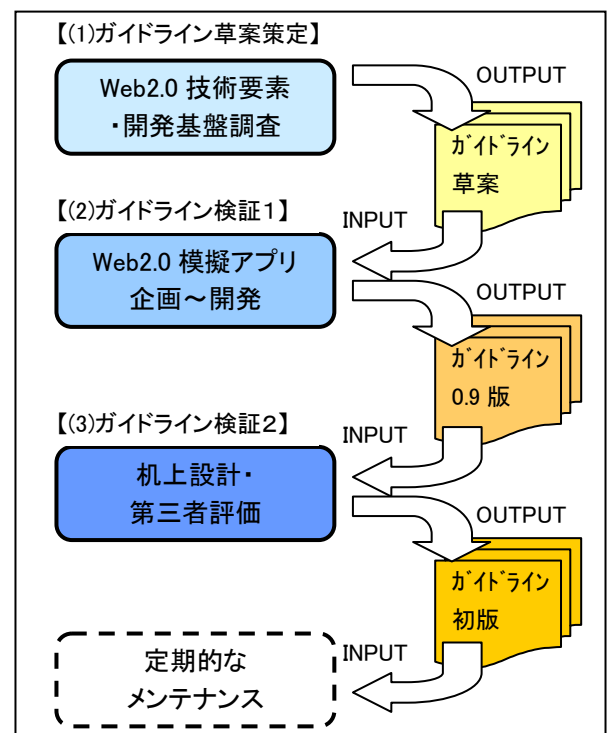
(1) ガイドライン草案策定

顧客要望と技術の関連付け、開発環境の調査、方式設計の考慮点の洗い出しを行い、内容の過不足等検討の上でまとめ、ガイドライン草案とした。

(2) 検証1・・・本当に現場で使えるの？

ガイドライン検証の第1段階として、Web2.0 要素を取り入れた仮想的システムを開発し、そこで得られた課題を反映することとした。ここでは、「ショッピングサイト等に比べ、Web2.0 化が進んでない」などの理由から図書館システムを選定し、Java と .NET の2種類の言語でアプリケーションを開発した。

図表 1 研究手順



結果として、「参照できるポイントが少ない」「考慮点が伝わりづらい」等多くの問題点があることが判明したが、大小さまざまな改善を施した 0.9 版で次の検証を行うこととした。

(3) 検証 2・・・目的、対象者に合っているの？

ガイドライン検証の第 2 段階として、検証 1 で開発した図書館システムへの機能追加を想定した再検証と、第三者によるアンケート調査による評価を行った。

再検証では、「技術調査の為に時間が短縮され開発効率のアップにつながった」「手戻りなく開発を進められた」と問題点が改善されている結果が得られた。また、アンケート調査では「具体性・論理性」「各社で利用可能か」等 6 項目中 5 項目で「そう思う」「どちらかといえばそう思う」という高評価が 90%に上った。ただし、「長く使える内容か」については 70%にとどまった。評価コメントで多数あった要望は可能な限り反映させ、ガイドライン初版として、完成させた。

3. 「Web2.0 適用／方式設計ガイドライン初版」は使える！

図表 2、図表 3 に完成したガイドラインの一部を紹介する。

図表 2 『Web2.0 システム要件の実装方式』より抜粋

システムに対する要求仕様(ユーザー要件・システム要件)				実装方法	
No.	分類項目	ユーザー要求	ユーザー要求の説明・具体例		
28	レスポンスの向上	更新速度向上支援	更新時にページ全体を更新したくない	2. 4. 1 (1)	
29		レスポンスの軽減	動作を軽くしたい		画面遷移、更新中の待ち時間によるストレスを軽減したい。
30			画面上の表示情報を増してもレスポンスは悪化させたくない		情報増加による画面更新中の待ち時間によるストレスを軽減したい。

図表 2 は提案や要件定義時に顧客からの要望が想定される Web2.0 の技術要素や各種制限について、実装可否および実装方法が一目でわかるよう「顧客要望」と「実装方法」をセットにしたマトリクスである。ガイドラインでは 7 分類 22 項目の Web2.0 要求項目について複数の実装方法を記述している。

図表 3 は、Web2.0 の方式設計時に検討が必要な項目について、「特徴」「選択指針」「事例」を記述し、Web2.0 開発未経験者でもアプリケーションの特性に応じて何を採用すればよいかの指標がわかるようになっている。

この通り、「Web2.0 適用／方式設計ガイドライン」はフレームワーク等技術情報部分の定期的なメンテナンスが必要であることが課題ではあるものの、当初の課題を解決し、本来の目的を十分に達成しうるものとする。また更なる改良を要する点としては、開発技術者以外の人への配慮を行うことや、具体的な図やフローの掲載等が挙げられる。

図表 3 『Web2.0 アプリケーション方式設計指針』より抜粋

2. サーバサイド方式
サーバサイドで Web サービスを呼び出した後、得られた XML や JSON の中から欲しい情報を抜き出してマッシュアップする。

* 結果は、Web アプリケーションにて、A, B, C すべての検索が完了後、まとめてブラウザへ返す

(2) 【選択指針】
システムの要求事項により、以下のメリット/デメリットをふまえて方式を選択する。

要求事項	クライアントサイド方式		サーバサイド方式
	プロキシ	JSONP	
データの同期取得	△	×	○
データの非同期取得	○	○	×
データの二次加工	△	△	○
開発効率	△	○	○
セキュリティ	○	×	○
Web サービス利用のログ記録	○	×	○

(3) 事例

- ・ 横断検索で各サービスの検索結果を非同期に取得したい場合はクライアントサイド方式のどちらかを選択する。しかし、ブラウザの同時アクセス制限を考慮する必要がある。
- ・ ユーザーインターフェイスを持つサービスを統合する場合は、クライアントサイド方式のプロキシを利用すると、ユーザーインターフェイスの修正が容易である。
- ・ JavaScript で提供された Web サービスを利用したい場合は、クライアントサイド方式の JSONP を利用すると、開発が容易である。
- ・ 二次加工するデータが多い場合はサーバサイド方式を選択する。

4. Web2.0 化の波をあなたの会社でも ～当分科会からの提言～

Web2.0 といっても、内部で使用されている技術は特別新しいわけではなく、既存技術の組み合わせである。それら既存の技術を組み合わせることにより、サービスの可能性が無限の広がりを見せる。また、既存の便利なサービスを自身のアプリケーションに組み込むことで、サービスの向上及び大幅なコスト削減が図れる、という点も大きな魅力である。技術革新のスピードが加速している現代で「新しい仕組みを取り込む部分」「既存の仕組みを残す部分」の取捨選択を迫られることはどのようなシステムにおいても避けては通れない道となる。まさにその指標を示してくれるのが「Web2.0 適用／方式設計ガイドライン」であるだろう。

まだ Web2.0 化していないアプリケーションには、多くの宝が眠っているはずである。これから Web2.0 アプリケーションを初めて開発する人も、既に Web2.0 アプリケーションを開発している人もガイドラインを活用し、宝を掘り起こす時が来た。

「そろそろ Web2.0 化しませんか？」