

リッチクライアントによる Web アプリケーション開発の実際

野村総合研究所
福岡システム基盤技術部 上級テクニカルエンジニア

浅辺 公彦（あさべ きみひこ）

情報技術本部にてオープンソースを中心とする最先端技術を活用したシステム開発に従事する IT エンジニア。最近注目しているのは Ajax により実現されるシンクライアント向けサービス API の効果的な活用技術。福岡での高度 IT システム開発を可能とするネットワーク作りを目標とし活動している。



1. リッチクライアントの現状	54
2. 開発事例 ～ JAL Flash 版国内線予約サービス～	57
3. 事例にみる開発の実際	60
4. 実績・効果	68
5. 今後の課題	69
6. リッチクライアント開発のポイント	70
7. 最後に	71
8. 謝辞	72

要旨

リッチクライアントは、Web アプリケーションシステムのクライアントである HTML のユーザービリティの低さを解決する技術として注目されている。しかし、業務システムのクライアントとしてリッチクライアントを採用する場合、いくつかの課題が残っている。本稿は JAL Flash 版国内線予約サービスの開発に至るまでの動機と背景を解説した後、本開発の経験より得たシステム開発におけるリッチクライアントの課題である「設計・生産性・テスト」のポイントを整理し、導入後の実績と効果、及び、今後の課題について述べる。

キーワード：リッチクライアント、Flash

Rich Client has been watched with keen interest as a technology to resolve low HTML usability. However, there are still several issues left to go on to utilize Rich Client as clients for business systems. This article describes motivation and background that drives us to develop the Domestic Reservation Service System using Flash for JAL. And it also summarizes key points on its design, productivity, and verification that had gained from this development experience, along with its result and effect and future issues after the installation.

Keywords : Rich client, Flash

1. リッチクライアントの現状

(1) リッチクライアントの特徴

ホストからオープンシステムへとアプリケーション開発の中心が移って以来、クライアントアプリケーションはファットクライアントからリッチクライアントへと進化してきた。

1990年代前半、オープンシステムの黎明期には、クライアントサーバと呼ばれる2層構造のシステムアーキテクチャが中心であった。

クライアントサーバ型のシステムではビジネスロジックはクライアントに配置され、アプリケーションが必要とする機能のほとんどをクライアントで実現した。これに対し、サーバはデータの検索・更新を受け持つ程度であり複雑なアプリケーション機能の実装は行われなかった。

この時代のクライアントをファットクライアントと呼び、豊かな操作性を持ったユーザインタフェースと複雑なビジネスロジックを実行可能な優れた実行環境が特徴であった。

しかし、ファットクライアントはクライアントPCにインストールが必要なEXE形式で実現しなければならず、アプリケーションの配布にコストがかかることが欠点であった。

1990年代後半になると、MVCモデルとよばれるシステムアーキテクチャが注目されるようになり、クライアント開発にWeb技術が利用されるようになった。

この時代のクライアントをシンククライアントと呼ぶ。シンククライアントはHTML+

Script言語により実現され、ブラウザ上で動作する。

ファットクライアントにおいてクライアントに配置されていたビジネスロジックはサーバに配置され、シンククライアントはユーザからの入力受付と簡単な入力値チェックと入力制御を行うに留まった。

シンククライアントは起動時にサーバからブラウザに常にダウンロードされるという特徴を持っており、ファットクライアントの欠点である配布の問題を解決した。

この配布が容易だという特徴からインターネットに代表される大規模ネットワーク・大規模ユーザ（特にコンシューマ）向けのクライアントとして爆発的な普及を遂げた。

しかし、シンククライアントではファットクライアントが持っていた豊かな操作性は実現できず、豊かな操作性を必要とする業務には利用できない、もしくは利用者に我慢して使っていただくという選択を強いることとなった。

	利点	欠点
ファットクライアント	豊かな操作性 高度な機能	配布にコストがかかる
シンククライアント	配布が容易	貧弱な操作性 高度な機能が実現できない

表1 利点と欠点

ファットクライアントとシンククライアントは表1に示すとおり互いの欠点は補完するものの、互いの利点は犠牲にしており、課題の残るクライアント技術であるといえる。

これに対する1つの回答としてリッチクライアントが誕生した。

リッチクライアントはファットクライアントが持つ豊かな操作性・高度な機能と、シンクライアントが持つ配布の容易さを実現しているのが特徴である。

(2) 主要なリッチクライアント製品

現在、リッチクライアントと定義されるクライアント技術はいくつかあるが、本稿では以下の特徴を持つものをリッチクライアントと定義する。

- 高い操作性を実現可能
- 複雑なビジネスロジックを実行可能

- 高い描画・表現が実現可能
- 配布が容易

これらの特徴を持ち、実用化されている代表的なリッチクライアント製品を表2に示す。

リッチクライアントはさまざまな技術で実現されているが、ここではマクロメディア社のFlashを例にリッチクライアントとはどのようなものか具体的な説明を行う。

Flashはブラウザにプラグインとしてインストールされた実行環境Flash Player上で動作する。

Internet Explorerをはじめとする一般的に普及しているブラウザは、あらかじめFlash Playerをプラグインとしてインストールし出

製品名	提供ベンダー	稼働環境	特徴
Flash	マクロメディア	Flash Player	ホームページ上で映像やアニメーションを活用した動的かつインタラクティブなコンテンツを作成する技術としてデファクトスタンダードな製品。本来は、映像やアニメーションを作成することにフォーカスした製品であったが、ActionScriptの進化により業務利用可能なリッチクライアント製品としての利用が進んでいる。
Flex	マクロメディア	Flash Player	MXMLと呼ばれるマークアップ言語にてFlashの画面及びプログラムを記述することを可能にした製品。Flashの実行ファイルであるSWFファイルをサーバ側で実行時に動的に生成しダウンロード可能。Flashに欠けていた業務利用を視野に入れた開発環境・実行環境の整備が進んでいる。
.NET	マイクロソフト	.NET Framework	.NET Framework上で動作するWindowsアプリケーション。 .NET Frameworkがインストールされているブラウザであれば、ブラウザで動作させることも可能。クライアントサーバ時代にデファクトスタンダードであったVisualBasicと同様の非常に優れた開発環境・実行環境を持つ。
Biz/Browser	アクシスソフト	Biz/Browser	Biz/Browser上で動作するWindowsアプリケーション。あらかじめBiz/Browserをインストールする必要がある。プラグインとしてブラウザ上での動作も可能。Chain Reflection Script (CRS) と呼ばれるスクリプト言語にて画面及びプログラムを記述する。業務利用にフォーカスした製品。
Curl	カールアジア・パシフィック	Curl Surge RTE	実行環境であるCurl Surge RTEをダウンロードしインストールして実行する。プラグインとしてブラウザ上での動作も可能。Curlと呼ばれるオブジェクト指向言語にて画面・プログラムを開発する。

表2 リッチクライアント製品

荷しているものが多く、一般のインターネット利用者がFlashを使用する場合、改めてFlash Playerのインストールを行う必要はほぼない。

また、Flash Playerのバージョンアップ等が発生した場合にも、簡単なクリック操作のみでバージョンアップ可能な環境が提供されており、従来のファットクライアントで経験したようなセットアッププログラムをダウンロードし実行するといった手間も必要としない。



図1 マクロメディア提供のFlashPlayerインストールサイト

Flashはホームページ上に貼られたリンクをクリックすることで起動可能で、起動時、SWFファイルと呼ばれる実行ファイルがブラウザにダウンロードされる。(図1)

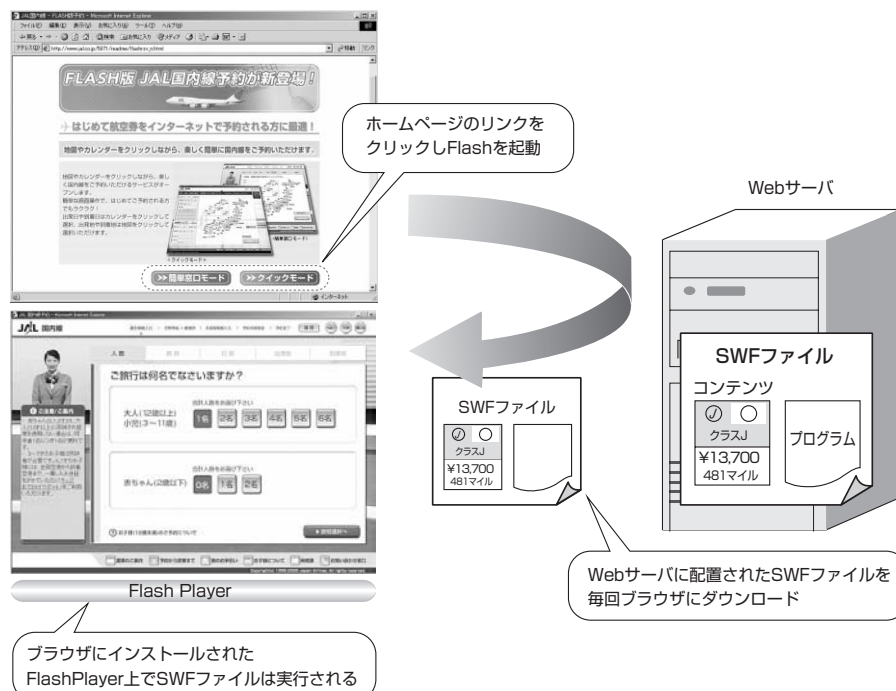


図2 Flash 実行イメージ

SWF ファイルにはリッチクライアントを構成する画面部品、映像、アニメーションといったコンテンツに加え、それらを制御する Action Script と呼ばれる Script 言語で記述されたプログラムが含まれている。

SWF ファイルはこれらのコンテンツやプログラムを効率的に圧縮するため、ファイルサイズが小さくなり、ダウンロードの所要時間を短縮している。

Flash は映像やアニメーションといった動きのあるコンテンツを作成するのに適しており、また、Action Script による高度なプログラミングも行えるため、HTML では実現できなかった高い操作性と機能を持つユーザインタフェースを実現できる。

さらに、Flash Player のインストール環境と SWF ファイルのダウンロード機能により

配布についても HTML 並みの容易さを実現している。(図2)

ホームページ上で映像やアニメーションを活用した動的かつインタラクティブなコンテンツを作成する技術としてデファクトスタンダード的な製品。

本来は、映像やアニメーションを作成することにフォーカスした製品であったが、Action Script の進化により業務利用可能なリッチクライアント製品としての利用が進んでいる。

2. 開発事例

～ JAL Flash 版国内線予約サービス～

開発事例として 2004 年 10 月にリリースされた JAL (株式会社日本航空) の Flash 版国内線予約サービスについて紹介する。

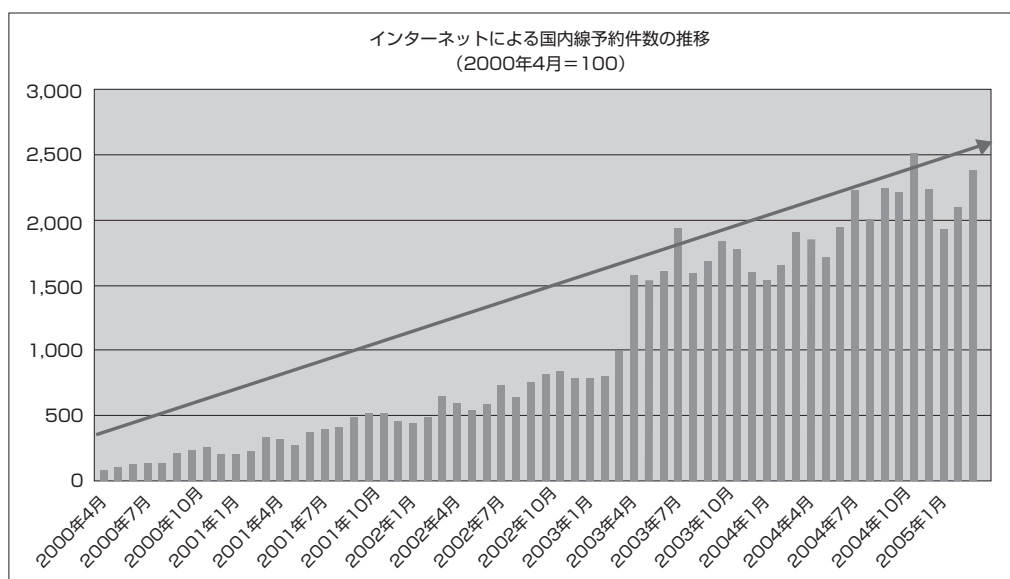


図3 国内線予約件数推移

引用：第15回 ソフトウェア開発環境展・リッチクライアントによる Web アプリケーション

(1) 動機と背景

JALの国内線におけるインターネットによる国内線予約者数はここ数年増加の一途をたどっている。(図3)

インターネット予約による売上額は2005年度で2,500億円に達する見込みであり、国内線に搭乗する約50%がインターネットにて予約を行っている。

JALはインターネットによる国内線予約の推進がJALとお客様にとって表3に示すようなWin&Winの関係を生み出すと分析しており、さらに利用者の拡大を推進する方針である。

インターネット予約の持つ価値	お客様のメリット	JALのメリット
- 適切かつ適時な情報提供 - 空席及び運賃の検索・比較機能の提供 24時間のサービス提供 - 予約スピード - セルフサービス	- 適切かつ適時な情報取得 (ex 台風時の欠航情報等) - 利便性(時間/移動)の向上 (ex 窓口業務時間外の予約等) - 価格・サービスの向上	- 費用削減 - 需要の増加 - 企画部門とお客様のダイレクトコミュニケーション実現

表3 インターネット予約がもたらすメリット

JALではこのような背景のもと、インターネット予約の利用率拡大を目的に新たなサービス提供の検討を開始した。

検討当時(2003年頃)、JALが提供していたHTML版の国内線予約は、予約窓口でオペレータが使用しているホスト端末が持つ機能、画面遷移をほぼそのまま移行したもので

あった。そのため、HTML版国内線予約はお客様の使い勝手にフォーカスして作成されたものではなかった。

よって、国内線予約を行うのに必要十分な機能提供は成されているものの、PC初心者や航空機の予約を初めてする人にとってはわかりにくいものとなっていた。

インターネット予約の利用率を拡大するためにはこのようなPC初心者や航空機にあまり乗らない人たちへも裾野を広げる必要がある。

そのため、新たなサービスが実現すべき価値は従来型のHTMLで作成された予約ページでは実現できなかった、「わかりやすさ」「直感的」「安心」「気持ちよさ」「快適さ」といったエモーショナルなキーワードへと集約され、実現手法としては映像やアニメーションを駆使できるFlashの採用が決定した。

(2) ターゲットとコンセプト

Flashにより実現される新しいサービスがターゲットとするお客様はHTML版でカバーできていなかった以下のお客様層に定められた。

- PC、インターネットに不慣れなお客様

【特徴】

PCやインターネット自体に不慣れで、航空機の予約方法以前に予約画面自体の操作方法に対する十分な説明を必要とするお客様。

● 航空機のご利用に不慣れなお客様

【特徴】

PC やインターネットは日常的に利用しており、HTML 版の予約画面であっても操作方法は理解できると想定されるお客様。ただし、航空機の利用は少なく予約自体に不慣れであるため、電話窓口等、対人での予約を利用している。

以上の特徴からわかるとおり、新しいサービスがターゲットとするお客様は必要としているものがそれぞれに異なり、企画の早い段階でそれぞれの特徴にあわせた2つの予約画面を提供することが決定した。(図4)

● 簡単窓口モード (図5)

「PC、インターネットに不慣れなお客様」をターゲットに提供し、電話窓口やカウンターで行っている対人による一問一答式のわかりやすいインタフェースを目指す。特に、このモードはわかりやすくあることを最重要とし、クリック数や予約完了までの所要時間が増加したとしても文字・アニメーションを駆使して十分な説明を行う。

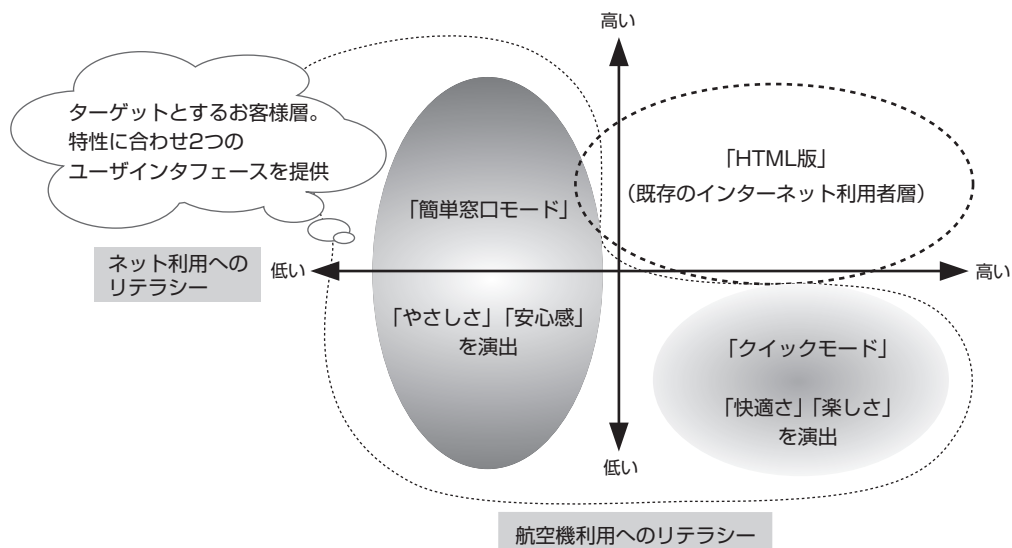


図4 ターゲットコンセプトの明確化

引用：第15回 ソフトウェア開発環境展・リッチクライアントによる Web アプリケーション



図5 簡単窓口モード

●クイックモード (図6)

「航空機の利用に不慣れなお客様」をターゲットに提供され、提供される予約画面を一目見て「使ってみようかな」と思わせるインタフェースを目指す。画面の構成は一枚の申込用紙をイメージし、リッチクライアントならではの表現力により、動きの気持ちよさ、楽しさ、かっこよさを実現する。



図6 クイックモード

3. 事例にみる開発の実際

Flash 版国内線予約サービス開発プロジェクトは2004年10月のリリースを目標に、2004年5月に設計フェーズはスタートした。

ここからは、Flash 版国内線予約サービスの開発を経験して得た設計、生産性、テストにおけるそれぞれのポイントについて解説する。

(1) 設計

～プロトタイピングの重要性～

Flash 版国内線予約サービスにおいて開発側に求められたものを集約するとFlashならではの「動きによるわかりやすさ」と「動きのある表現」であった。

ここで既にリリースされているFlash 版国内線予約サービスにて「動きによるわかりやすさ」「動きのある表現」がどのように実現されているかを説明する。

簡単窓口モードは必ず次の順序で画面表示が行われる。(図7)

- ① お客様に対する問い合わせの表示
(常に画面左上に表示する)
- ② 問い合わせに対する回答の入力欄の表示
(画面中央に表示する)
- ③ 回答の決定ボタンの表示に表示する
(常に画面右下に表示する)

人間は視覚から情報を取得する際、同時かつ複数表示されたものを理解するのは得意ではない。これに対し、順序だてて1つずつ表示される情報に関してはより早く、より深く理解するという特性を持っている。

また、文書等の視覚情報を取得する場合、視線を下から上へと動かすより、上から下へと動かす方が慣れている。

簡単窓口モードの画面はこのような人間の特性を考慮した画面の動きを行うことで、HTML版にはなかったわかりやすさを実現している。

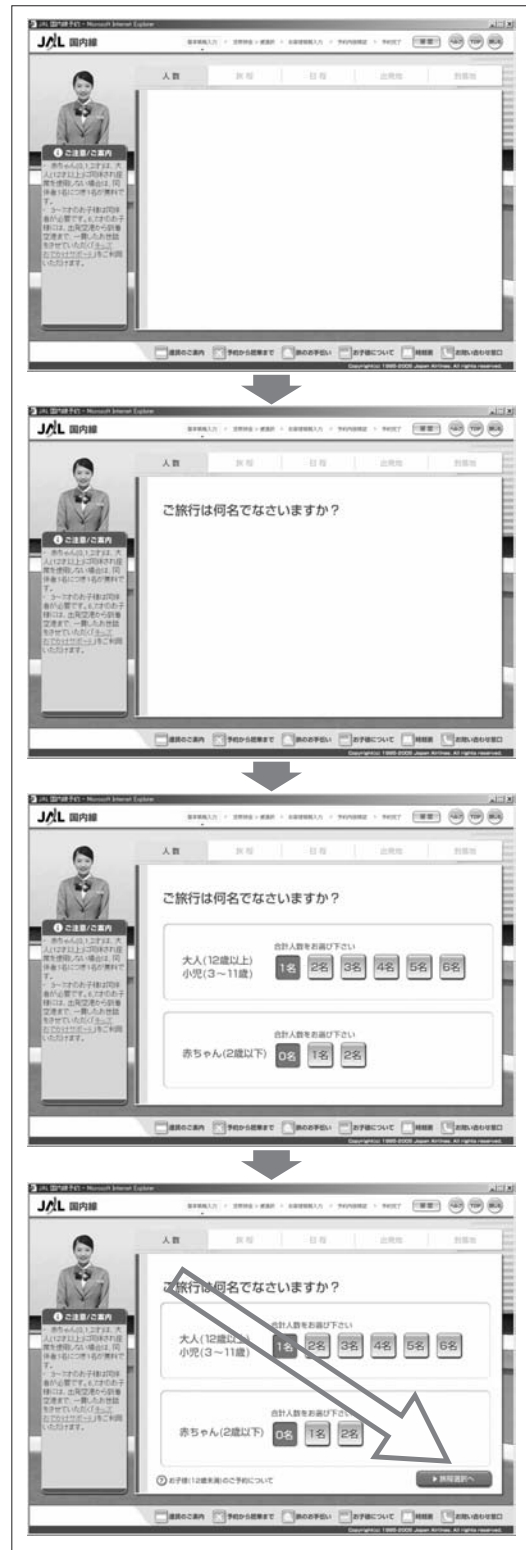


図7 簡単窓口モードの画面表示順序

Flash 版国内線予約サービスの設計フェーズでは、このような動きを設計し JAL の担当者に理解していただく必要があった。

通常、SI (System Integration) における画面設計は図8に示すような画面設計書にて行う。

開発者は画面設計書を元に画面の構成や機能・動作要件等をレビューし、承認をいただく

というステップを踏みながら設計を進めていく。

この手法は HTML 等による静的な画面では有効であったが、動きのある画面の設計には不十分であった。

動きのある画面の設計には設計書自体が動きを持つ必要があり、プロトタイプによる設計が不可欠となる。



機能一覧		
1	大人/子供人数入力欄	・ 1名～6名の人数を指定するトグルボタンを表示する。 ・ 押下されたボタンが指定人数となる。 ・ 1つのボタンを押下すると、他のボタンは非押下状態になる。 ・ 初期時 (Flash起動時) は1名が押下された状態で表示される。 ・ 初期時 (最初からやり直す押下時) は前回押下された人数ボタンが押下された状態で表示される。
2	幼児人数入力欄	・ 0名～2名の人数を指定するトグルボタンを表示する。 ・ 押下されたボタンが指定人数となる。 ・ 1つのボタンを押下すると、他のボタンは非押下状態になる。 ・ 初期時 (Flash起動時) は0名が押下された状態で表示される。 ・ 初期時 (最初からやり直す押下時) は前回押下された人数ボタンが押下された状態で表示される。
3	次画面遷移ボタン	・ 旅程画面へ遷移する。 ・ 基本情報入力内容表示の人数欄に入力された人数を表示する。 ・ 「最初からやり直す」押下時の初期表示に使用できるよう入力内容を記憶する。
4	ヘルプ	・ 画面毎に定められた個数のアイコンを表示する。 ・ ヘルプアイコン押下によりヘルプ文言が表示される。 ・ ヘルプ文言は外部ファイルとして管理する。 ・ ヘルプ文言用外部ファイルはswfもしくはテキスト形式とする。 ・ テキスト形式とした場合には限定されたHTMLタグ (リンク・色・サイズ・太さ) が使用できる。 ・ 外部ファイルの期限管理は行わない。 ・ 外部ファイル名称はファイル毎にあらかじめ固定のファイル名称をつけるものとし、変更・追加はできない。 ・ 外部ファイルの内容を変更するとその時点より変更された内容が表示される。
5	TOP画面へボタン	・ オープニング画面へ遷移する。
6	閉じる	・ Flashを閉じる。
チェック一覧		
1	大人/子供人数入力欄	幼児>大人のボタンを押下された場合、コンファームを表示する。
2	幼児人数入力欄	幼児>大人のボタンを押下された場合、コンファームを表示する。
3	次画面遷移ボタン	幼児>大人のボタンを押下された場合、エラーとする。

図8 画面設計書

Flash 版国内線予約サービスの設計フェーズでは、当初、画面全体の設計は従来型の画面設計書で行い、一部画面のみプロトタイプを作成するという手法を取った。

我々は一部のプロトタイプで実現されている動きから Flash 版国内線予約サービス全体として「動きによるわかりやすさ」が実現されていることを説明しようと試みた。

しかし、JAL の担当者は多くの画面の動きを「想像」により理解しなければならず、設計は確定せずスケジュールは遅延していった。(図9)

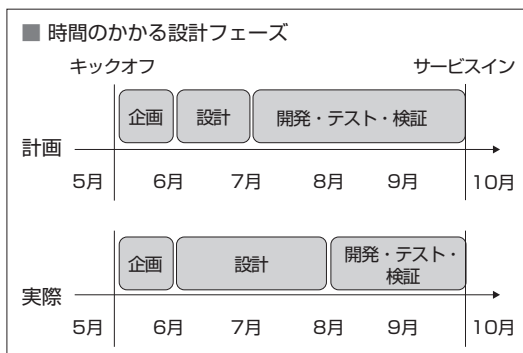


図9 設計スケジュール

引用：第15回 ソフトウェア開発環境展・
リッチクライアントによる Web アプリケーション

当初の設計スケジュールの終了期間を迎える頃、我々は画面設計を全画面についてプロトタイプを作成する方針に切り替えた。

ただし、これまでの設計でデザイン込みプロトタイプの作成には多くのコストがかかることを経験していたため、以下のステップでプロトタイピングを進めるものとした。

- ① 動きのみを表現する枠組みのプロトタイプ作成
- ② 画面のデザインは従来通り画面設計書(絵)で確定
- ③ 2で確定したデザインをプロトタイプに付与

図10はクイックモードのプロトタイプを

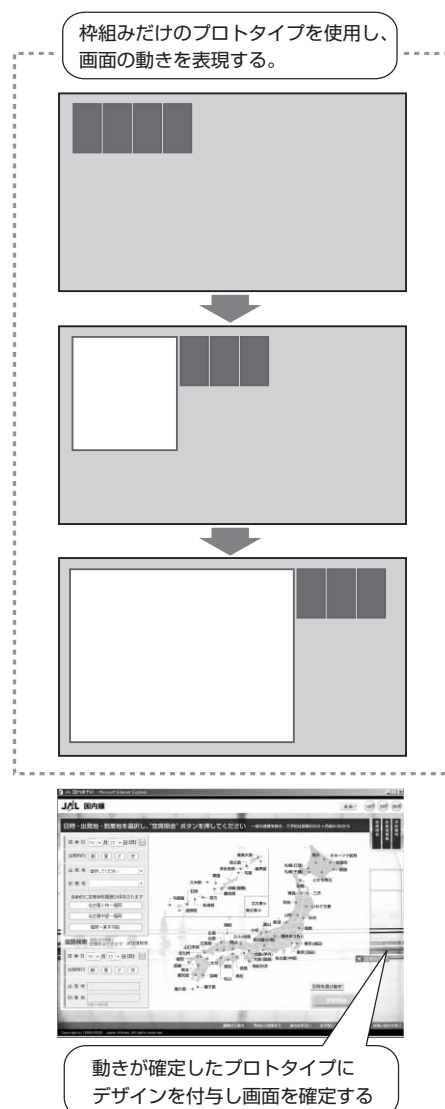


図10 クイックモードのプロトタイプ

例示したものである。

クイックモードの画面は予約のステップを画面上部の黒い四角で示しており、各ステップの操作を行う段階で黒い四角が拡張し操作画面となり、表示される。

この動きを枠組みのみのプロトタイプで表現した後、画面デザインを付与した。

この枠組みのみのプロトタイピングを導入することで遅延していた設計が進捗し、開発フェーズを開始することができた。

以上のように、Flashにおける設計では画面の見た目（デザイン）、機能、動きを明確に分け、それぞれに適した設計手法を適用することがポイントとなる。

今回我々が最終的に行き着いた枠組みのみのプロトタイピングを使用した設計手法は効果的であると考える。

（２）生産性 ～部品化～

ここからはFlashにおける生産性のポイントについて説明する。

Flashによる開発と、HTMLやVisual Basicによる開発の生産性を比較した場合、Flashの生産性は数分の１～数十分の１に悪くなると考える。

この原因は開発範囲の違いにある。

HTMLやVisual Basicでは画面を構成するボタンやテキストボックスといった部品が標準で提供される。

さらに、これらの部品にはサードパーティ製品も多く、ロウソク足チャート等の業務色をもった特殊な部品であっても、比較的容易かつ安価に入手することができる。

また、これらの部品はWindows標準の操作性、動き、表現を最初から実装しており、画面設計書に従い部品を配置することで画面作成が行えると共に、Windows標準のユーザインタフェースが得られる。

そのため、ユーザインタフェース開発者は機能面の開発に注力すればよく、ユーザインタフェースのデザインや動きといった部分をほとんど意識する必要がない。

これに対して、Flashにおける画面開発はHTMLやVisual Basicでは開発範囲外となる部分も全て開発しなければならない。

Flashでは統一規格の画面を作成する事はほとんどない。

開発依頼側の要望や用途、ターゲットとするユーザ、実現する機能に応じて自由にデザインし作成するのが一般的であり、それがFlashを用いる意義でもある。

そのため、Flashの画面開発では、ボタンの形はどうするか、スプレッド内の情報をどう配置するか、すべていちから創造していく必要がある。



図 11 画面部品

図 11 に Flash 版国内線予約サービスで開発した画面部品を例示する。

Flash 版国内線予約サービスでは、用途や機能に応じて、色や形、押下されたときの動きが異なるボタンを数種類作成している。

また、1つのセル内に複数の情報を表示する空席状況表用のスプレッドも、よりわかりやすく情報を伝達できるようデザインを行っている。

これらの部品はあらかじめ提供されているものではなく、すべて Flash 版国内線予約サービス専用の部品としていちから作り込んでいる。

以上のように Flash では、HTML や Visual Basic ではほとんど必要とされなかった、画面部品のデザインや操作性の設計・開発が別途必要となり、その開発工数が生産性を引き下げる最大の要因となる。

Flash はもともと映像やアニメーションといったエンターテインメント的なコンテンツを作成することにフォーカスを当てた製品である。

よって、SI における生産性向上のための常套手段である、「部品化」や「フレームワーク化」といった分野が製品として十分に整備されていない。

例えば、Flash にもヘイローと呼ばれる標準部品が存在する。しかし、その種類、機能、完成度は Visual Basic で提供されている部品群に遠く及ばない。

Flash 版国内線予約サービスの開発でもヘイローを使用しているが、スプレッドシートのスクロールバー等、画面デザインと親和性を取りやすく、単純な機能しか要求されない部分に留めている。

現時点で Flash 開発の生産性を向上するためには独自で「部品化」や「フレームワーク化」を推し進める必要である。

この「部品化」や「フレームワーク化」は開発工程中、外部設計段階であるプロトタイプフェーズから着手することが望ましい。

プロトタイプフェーズにおいて、常に画面を構造化し「デザイン」と「機能」に明確に分離することを心がける。

プロトタイプフェーズ初期においては機能面の部品化・共通化を進め、プロトタイプフェーズ後期で個別のデザインを付与していくというステップを踏む。

また、機能設計が完了したタイミングで、サーバサイドとの通信機能、取得データの解析機能といった画面のバックグラウンドで動作するアプリケーション機能を部品化することも重要となる。

このような手順を踏むことで、開発フェーズにおける生産性を高め、仕様変更にも柔軟な対応がある程度は行えるようになる。

しかし、Flash の新規開発にて十分な生産性を得ることは非常に難しい。

新規開発ではやはり開発物の全体感（デザインや動き）を決定することに多大な工数を割く必要があり、プロトタイプ時に十分な部品化を進める時間を割くのは困難になりがちである。

よって、可能であれば2次開発、3次開発を踏まえて生産性を考えた方がよい。

新規開発で作成した画面部品はコンテンツ

のデザインテイストを統一するため、2次開発、3次開発でも流用可能となるケースが多い。

新規開発で十分な部品化を進め、2次開発、3次開発でトータルの生産性を上げるとするのが現実的であると考ええる。

また、先に述べた部品の「デザイン」「機能」の分離に留意し、他の開発案件においても「機能」部分を流用することでFlash 開発案件全体の生産性を向上していくことが重要である。

（3）テスト ～赤くならないボタン～

ここからはFlash におけるテストのポイントについて説明する。



図 12 空席便の選択画面

Flash 版国内線予約サービスでは、表示画面のすべての操作が完了し、次画面に進める状態になると、画面右下のボタンが赤くなり、次画面へ進めることを示す。

リリースを間近に控えたテスト期間中、空席便の選択画面（図 12）にて、このボタンが3日に1度くらい赤くならないという障害が発生した。

今回の開発は非常に切迫したスケジュールにて開発を行ったため、数多くの障害が発生したのだが、その中でもこの障害が Flash におけるテストのポイントを示す象徴的な障害となった。

障害発生当初、我々はプログラムバグによるものであろうと安易に考えていた。

しかし、再三のプログラムチェックでも原因が判明せず、また再現性も長期間に渡り確認することができなかった。

我々は当該画面に対するモンキータイプテストを繰り返し「ボタンが赤くなる瞬間にクラスJのバナーをフラッシュ」させると再現するということを発見した。しかし、その原因については究明することができなかった。

両処理を記述したプログラムには本障害が要因となる記述は見当たらなかったのである。

最終的に、我々は根本的な障害対応をあきらめ、この2つの処理が重ならないよう制御するプログラムを追加することで対象療法的に不具合を回避せざるをえなかった。

本開発において、この障害以外にも原因を特定できない障害が多数現象が多数発生した。

我々はそのつど、徹底したモンキータイプテストで再現方法を確認し対症療法を施すというトライアンドエラー的なステップを繰り返し、品質をあげていった。

通常の SI ではテストケースを事前に設定し、テスト担当者がテストケースに従いテストを消化していくのが一般的である。

しかし、Flash における開発においてこの手法では、満足のいく品質を得られないということが今回の開発でわかった。

Flash で実現されたユーザインタフェースの操作には、例えばボタンを押すという操作だけでも、ボタン上へ移動するマウスの経路やタイミングといった、文書で書かれたテストケースでは表現できない数多くのバリエーションが存在している。

Flash によるユーザインタフェースのテストではこれらのバリエーションを消化していく必要があるのである。

今回のテストフェーズにおいて、経験のあるテスト担当者や、開発物をよく理解しているスキルのある担当者は、単なるテスターより効率的に障害の再現性や対症療法を発見するという傾向が見られた。

おそらく彼らは意識的に、もしくは無意識のうちに不具合の発生しやすいバリエーションでテストケースを消化していると思われる。

ただし、今回の開発でもそうであったが、そのような経験・スキルを持った人材ばかりでテストチーム・開発チームを構成するのはほとんど不可能である。

よって、Flashによるユーザインタフェースのテストではモンキータイプのテスト項目を多数設定し、それを複数人で消化することで無意識の操作バリエーションを増やし、障害発見につなげることが重要である。

4. 実績・効果

Flash版国内線予約Flashサービスは2004年10月にリリースされ1年が経過している。

リリース後、2次開発として2005年4月に座席指定機能の追加を行い、また中部国際空港の追加やいわて花巻空港の名称変更等のメンテナンスも実施している。

ここでは、導入後、1年間で見られた実績及び効果について述べる。

(1) Flash版国内線予約サービスの利用層

HTML版の国内線予約サービスの利用層の中心は出張等で航空機を利用する業務利用者である。

よって、利用者の多くをサラリーマンが占める傾向にあり、男女比・年齢層で見した場合、男性中心となっている。

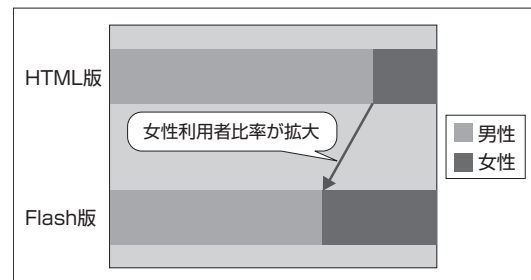


図13 利用者の男女比

引用：第15回 ソフトウェア開発環境展・
リッチクライアントによるWebアプリケーション

図13にFlash版とHTML版の利用者層の男女比を比較したグラフを示す。

具体的な比率を示す数値は提示できないが、Flash版国内線予約サービスでは女性の比率が拡大していることがわかる。

今回、JALがターゲットとした利用層は定性的に航空機を利用する人々ではなく、年に数度の旅行で利用するような、航空機利用のリテラシーが低いお客様である。

今回の結果は、男性中心の業務利用者ではなく、これまでHTML版を利用していなかった女性を取り込むことに成功しているといえる。

(2) 利用時間帯別の予約動向

次に利用時間帯別の予約動向を図14に示す。HTML版は一般的な営業時間帯、午前9時から午後5時までの予約比率が高いのに対し、Flash版国内線予約サービスの利用者は夜間の予約比率が高くなっていることがわかる。

これは家庭で Flash 版国内線予約サービスを利用するユーザが多いことが推測され、この結果からも、HTML 版の利用層とは異なる利用層を獲得できていることがわかる。

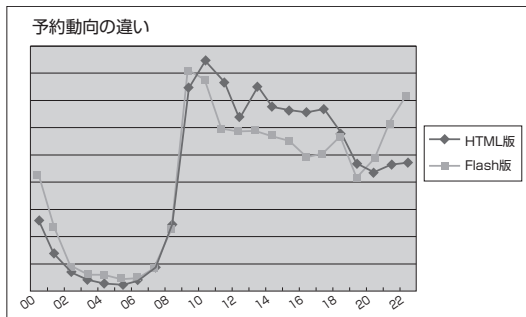


図14 時間帯別の予約動向

引用：第15回 ソフトウェア開発環境展・
リッチクライアントによる Web アプリケーション

その他の統計情報として、1つの予約における搭乗人数の増加や、往復予約比率の増加、またチケットの購入比率の増加が Flash 版国内線予約サービスでは見られた。

JAL が想定した Flash 版国内線予約サービスの利用シーンとして「航空機利用、PC 及びインターネット利用に対するリテラシーの低いユーザが、旅行等で利用する航空機の予約を行う」ことを想定していた。

先に述べた実績は、複数人で往復旅程の予約を行い、チケットまで購入していることを示しており、まさに想定していたシナリオでの利用が成されていると考えられ、一定の効果を上げているといえる。

5. 今後の課題

本章では、Flash 版国内線予約サービスの開発と、1年間のメンテナンスを経験することで見えてきた課題について述べる。

(1) メンテナンスにおけるコスト効率

Flash 版国内線予約サービスでは2回の空港追加と名称変更による改修を経験している。

HTML 版の国内線予約サービスにとっては、空港は文字列で構成されるデータとして管理されている。

よって、空港の追加や名称変更による改修はデータ変更にかかるコストで実施可能である。

これに対し、Flash 版国内線予約サービスにおける空港はデータであると共に、画像として管理されている。



図15 空港名称

例えば、図15の地図上に表示されている空港名はすべて文字データではなく文字の形をした「絵」である。

メンテナンス性をよくするためには、HTML版と同じく空港名はすべて文字データとしてFlash内で管理し、その文字データをそのまま地図上に表示すべきである。

しかし、文字データの表示イメージは非常に貧弱であり、地図画面のデザインにそぐわなかった。

デザインはFlashにおけるユーザインタフェース開発の最大のポイントの1つであるため、デザイン性が求められる画面では「絵」で文字を表現し、求められない画面では「文字データ」で管理するという使い分けを行った。

この手法はFlashでは一般的に用いられる手法であり、Flash版国内線予約サービスに特定のものではない。

この結果、空港の追加や名称変更が発生する度に空港名の絵を作成するコストが新たに発生することとなった。

Flashの成果物はシステムというより映像やアニメーションに近い。

この成果物をデータベースで管理されるデータ等と同様にフレキシブルにメンテナンス可能にするためには、開発方式とは別のデザイン環境や表現能力の格段の進歩が必要であると考ええる。

(2) ユーザインタフェース設計の方法論

開発の実際でも述べたが、Flash版国内線予約サービスの設計には多くの時間を要した。

その原因には設計手法の不適當さによる部分もあったが、開発に携わった全員がFlash版国内線予約サービスが必要とするユーザビリティを定めていく方法論を持っていなかったことも要因として挙げられる。

JALの担当者も含め、いったいどのようなユーザインタフェースにすれば求めている効果、ユーザビリティを得られるのか、想像し模索しながら設計を進めた。

これが、設計が二転三転する1つの要因となった。

開発する画面のデザインは用途やターゲットとするユーザにより千差万別となるのは仕方がない。

しかし、例えば先に述べた「上から下への順序立てた情報表示」手法や、画面の配色による利用者の心象といった効果は統計的に把握可能である。

このような統計に裏打ちされた効果を整理しユーザインタフェース設計の方法論として確立し、普及させることが今後のリッチクライアント開発には求められると考える。

6. リッチクライアント開発のポイント

Flash版JAL国内線予約サービスの開発事例をもとに実際の開発と効果、課題について述べてきた。

このプロジェクトの経験から、リッチクライアント開発のポイントは先に述べた「デザインプロトタイピング」「部品化」「モンキー

タイプテストの徹底」の3つに集約されると考えている。(図 16)

これらリッチクライアント開発のポイントを、ユーザインタフェース開発の歴史に照らし合わせて考えて見ると、ファットクライアント開発の黎明期、開発環境（製品・ツール）が十分整備されていない頃に盛んに議論されていた開発方法論に酷似していることに気付く。

つまり、現在のリッチクライアント開発のポイントというのは、リッチクライアント特有の課題に起因するものではなく、開発環境、実行環境、テスト環境といった開発環境を構成する製品、ツールが未成熟であることに起因したものが多いいえる。

こういった状況下でスムーズに開発を進めるためには、やはりファットクライアントの黎明期に蓄積されたノウハウを参考にするのが最も効果的である。

現在、Flash 開発の中心となっている技術者の多くは、どちらかというと自由度の高い表現力を生かせる、デザイナーに近い技術者が中心である。

そういった技術者が先に述べたノウハウを熟知しているケースは少ない。

逆にかつてファットクライアント開発を経験しノウハウを知り尽くした技術者が Flash の表現力を生かせるスキルやセンスを持つケースは少ない。

今後、業務利用を目的とする Flash 開発の体制を考える場合、ノウハウを熟知した経験

豊富な Flash 開発の技術者を獲得するか、もしくは、ファットクライアント開発経験者がノウハウを提供し、Flash 開発技術者と力を合わせてプロジェクトを推進していける体制を作っていくことが重要だと考える。

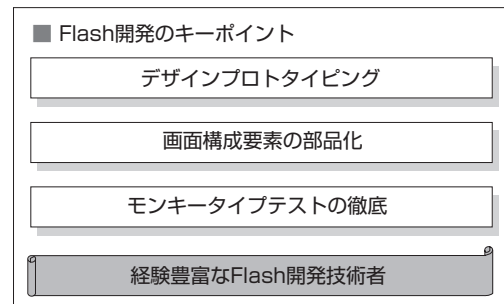


図 16 Flash 開発のポイント

引用：第15回 ソフトウェア開発環境展・
リッチクライアントによる Web アプリケーション

7. 最後に

Flash は映像やアニメーションといったエンターテインメント的要素を加えつつ業務的なリッチクライアントを開発できるソリューションとしてデファクトスタンダードの道を歩むであろう。

しかし、生産性に直結する開発方式・開発／テスト環境という視点からみた場合、映像やアニメーションを作ることを当初の目的に開発された Flash はまだまだ過渡期にあり、今後も大きく変わっていくと考える。

例えば、昨年夏から提供が始まり導入が進んでいる Flex は MXML と呼ばれる XML で Flash のプレゼンテーションとビジネスロジ

ックを記述できる等、今後の Flash 開発ツールの進むべき1つの道筋を示している。

また、リッチインターネットアプリケーションの世界にもオープンソースの波が押し寄せており、Flex と同じく XML ベースで Flash 開発が行える Open Laszlo 等、今後のデファクトスタンダードとなり得るソリューションも出現してきている。

今後、リッチクライアントの世界でもオープンソースが大きな流れとなることは間違いない。

しかし、繰り返しになるが、過渡期である今、どのソリューションを選んだとしても、本節で述べた開発ポイントを踏まえ図 17 に示す基本的な開発工程を確実に進めることが開発プロジェクト成功の鍵となるであろう。

8. 謝辞

本稿を執筆するにあたり、株式会社日本航空インターナショナルの小野陽弘様には、SODEC における講演資料のご提供、及び、査読等、多大なご協力をいただきました。

ここに厚く感謝の意を表します。

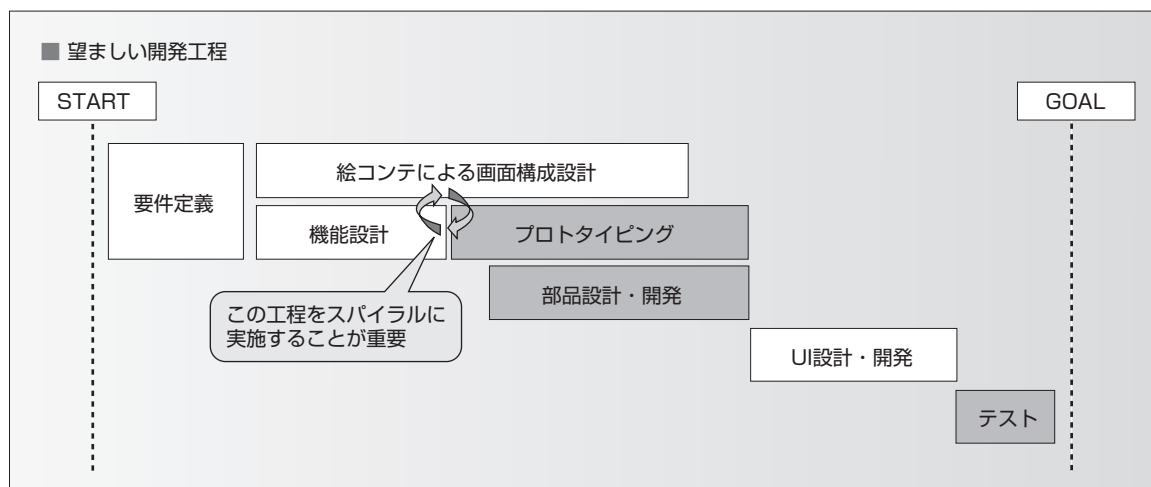


図 17 開発工程

引用：第15回 ソフトウェア開発環境展・リッチクライアントによる Web アプリケーション