

BEA WebLogic Server と Biz/Browser の融合による開発効率の向上について

沖電気工業株式会社

沖コンサルティングソリューションズ株式会社

公開日: 2004.3

OKI

ご注意

本ドキュメントの内容は、予告なしに変更されることがあります。

沖電気工業株式会社および沖コンサルティングソリューションズ株式会社は本ドキュメントに掲載されている技術的内容および欠落点について一切の責任を負いません。

本ドキュメントは、掲載されている製品等の技術的保証をするものではありませんので、本ドキュメントに従ったシステム実装および製品利用によって損害が生じた場合も、一切の責任を負いません。

Biz/Browser、および Biz/Designer はアクシソフト株式会社の商標または登録商標です。

BEA, BEA WebLogic, BEA Tuxedo, BEA WebLogic Server, BEA WebLogic Platform および BEA WebLogic Workshop は BEA Systems, Inc.の商標または登録商標です。

本ドキュメントに掲載されているその他の製品名およびソリューション名等の商標は各社の商標または登録商標です。

All trademarks are property of their respective owners.

© Copyright, 2004. OKI Electric Industry Co.,Ltd. All rights reserved.

© Copyright, 2004. OKI Consulting Solutions Co.,Ltd. All rights reserved

目次

Webシステムの課題とBEA WebLogic Platform	1
➤ Webシステムの課題	1
➤ BEA WebLogic Platformによる解決	2
BEA WebLogic WorkshopとBiz/Browser.....	3
➤ BEA WebLogic Workshopの特徴	3
➤ WebシステムにおけるBiz/Browserの特徴	5
➤ 両者の融合によるメリット	5
Biz/Browserの高い生産性について	7
➤ 生産性検証のモデル.....	7
➤ 検証方法.....	7
➤ 作成したシステムの生産性に関する考察.....	8
➤ JSPで作成した場合について	9
➤ Appletで作成した場合について	10
➤ Biz/Browserで作成した場合について.....	10
まとめ	11
参考：リッチシンクライアントの優位性.....	12
➤ 端末としてのシンクライアント	12
➤ リッチシンクライアントとは	12
➤ リッチシンクライアント適用の優位性	13

Web システムの課題と BEA WebLogic Platform

➤ Web システムの課題

クライアントサーバシステムは、クライアント側にロジックを保存するために発生する、次のような問題がありました。

- ・個々のクライアントへソフトウェアの配布が必要であり、大規模バージョンアップへ対応する場合に問題となる
- ・システム毎に複数のクライアントを使い分ける必要があるため、従業員の生産性を阻害する

上記のため、システム全体の維持管理コストや、従業員への教育を含む運用コストが課題となっていました。Web システムはこれらの課題を解決することが可能であるため、企業内の業務システムへの導入が進んでいます。しかし、Web システムも以下のように別の課題を持っています。

- ・ Web ブラウザによる制限
- ・ 構成要素が複雑
- ・ 効率的な開発手法が必要

このため、Web システムでは従来のクライアントサーバシステムを置き換えることができない場合がありました。

- ・ Web ブラウザによる制限

HTML に起因する操作性の低下、画面全体を書き換えることによるレスポンスの低下、帳票印刷などの機能の不足、これらは Web ブラウザの制限によるものです。この点については、アクシスソフト社のリッチクライアント製品 Biz/Browser などで解決することが可能です。Biz/Browser については、次章の「BEA WebLogic Server と Biz/Browser」以降で詳細に説明しますので、本章では割愛します。

- ・ 構成要素が複雑

近年、Web システムのアプリケーションプラットフォームとして、APS(Application Platform Suites)が提唱されています(APS とは、アプリケーションサーバ・インテグレーション製品・ポータル製品、など 3 つ以上の基盤を統合したスイート製品の総称です)。IT システムに対する要望の拡大に伴い、「複数の製品を組み合わせた複合ソリューション」が増加してしまいました。これらの各製品はベンダやアーキテクチャが異なるため、統合するには技術的な多くの課題をクリアする必要があり、これが新たな問題となってきました。またその一方で、TCO に占める複数ソフトウェア製品の統合コストの割合が高くなりつつあるという問題もありました。

- ・効率的な開発手法が必要

最近のアプリケーション開発での問題点は、開発プロジェクトごとにさまざまなプログラミング言語、技術、開発プロセス、開発ツールがあり、プロジェクトごとにそれに精通した人間を確保しなくてはならないということです。J2EE を利用する場合、ビジネスロジックを実装するには J2EE に関する高い知識と技術が必要になるため、開発の効率化およびスキルの高い開発要員の確保が難しくなります。こういった課題に対して、IDE(統合開発環境)がある程度カバーしてくれる場合もありますが、ウィザードなどで自動生成されたものが要件に一致することはまれであり、多かれ少なかれなんらかのカスタマイズが必要となっていました。このため、実際のプロジェクトで使用するには十分とは言えませんでした。

➤ BEA WebLogic Platform による解決

BEA WebLogic Platform は、上述した課題を以下のように解決することができるアプリケーションプラットフォームです。

- ・統合されたプラットフォームにより複雑さを排除
- ・効率的な開発環境の提供

- ・統合されたプラットフォームにより複雑さを排除

BEA WebLogic Platform は、アプリケーションサーバ・インテグレーション製品・ポータル製品を統合した単一の製品として、すべての機能が統合された真の APS 環境を提供しています。このため、複数製品を統合するための余分なコストをかける必要がありません。また、お客様が最初から APS の製品すべてを必要とされない場合でも、アプリケーションサーバ(BEA WebLogic Server)から導入し、インテグレーション製品(BEA WebLogic Integration)やポータル製品(BEA WebLogic Portal)を既存プラットフォームにシームレスに追加ができるというメリットがあります。

- ・効率的な開発環境の提供

BEA WebLogic Workshop(以下 Workshop)は、開発スキルの問題を支援し、さらに実際のプロジェクトで実用に耐える効率的な開発環境であり、実行時のランタイムフレームワークです。しかも、開発環境としての Workshop は無償で提供されます。Workshop は、J2EE の初心者にはコード自動生成機能などで支援するのはもちろんですが、エキスパートは、さらにその価値を実感できます。これは以下の理由によります。

Workshop で開発するモジュールはすべて、イベント、プロパティ、コントロール、の 3 つの要素に集約されます。これらを Workshop で変更すると、それぞれ、Java ソースファイルのメソッド、Javadoc コメントによるコントロールプロパティ、コントロールクラス変数、にリアルタイムに出力されます。つまり、Workshop はこれまでの IDE と異なり、すべての成果物(ページフロー定義やビジネスプロセス定義など)を、編集可能な Java ソースに書き出します。この特長によりエキスパートレベルの開発者は、詳細なふるまいを自由に追加コーディングしていくことが可能です。

Workshop については、次章で Biz/Browser と共に詳細に説明します。

BEA WebLogic Workshop と Biz/Browser

➤ BEA WebLogic Workshop の特徴

前述のように BEA WebLogic Workshop (以下 Workshop) は、非常に効率の高い開発環境であり、実行時のランタイムフレームワークです。しかも、開発環境としての Workshop は、BEA WebLogic Platform 8.1 の無償開発者ライセンスと共に、無償で利用することができます。

Workshop は、従来の J2EE 開発で標準的に用いられる Struts ベースの MVC モデルを改良した、新しい開発手法です。Struts を始めとする従来の MVC モデルは秀逸ですが、次のような課題がありました。

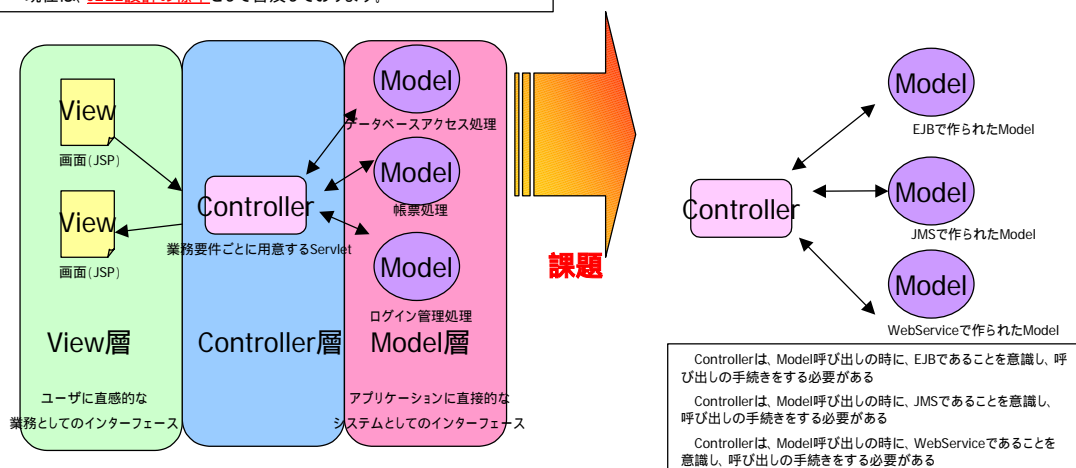
- 画面遷移のフローがわかりにくい
- Model 呼び出しの際、固有の技術を意識する必要がある

MVCモデルとは

- MVCモデル (Model, View, Controller アーキテクチャ) では、処理を以下の3層に区分します。
 - 画面を描画する処理層 (View)
 - 画面遷移を管理する層 (Controller)
 - 実機能を処理する層 (Model)
- MVCモデルを用いると、開発のしやすさ、メンテナンスのしやすさ、業務と処理の関係が直感的になるといったメリットを得ることができます。
- 現在は、**J2EE設計の標準**として普及しております。

MVCモデルの課題

- MVCモデルの実装は、Struts (Jakartaプロジェクト) が有名ですが、画面遷移やビジネスプロセスの構成などの設計図とアプリケーションの実体とが乖離しがちという問題が課題です。
- 画面遷移のルールはXMLファイルで管理され、設計者が手で編集する必要があり、設定上の間違いが起きることが課題です。
- 論理的な区分は出来ても、実装時には、Modelごとの初期化処理や呼び出し手続きを行なう必要があります。ビジネスロジックとはちがう、わずらわしい手続きが課題です。



Workshop は、画面フローの課題を Java Page Flow (以下 JPF) という技術で解決します。JPF 自体は XML 形式の定義情報として管理され、設計者及び開発者は GUI を通して編集作業を行います。

提供される GUI は、当該 Web システムが利用者に対して提供する最初のページや、各種ページのフォームから呼び出される処理ロジック (Struts でいう Action) との関係、またページとページとの関係を見やすくわかりやすい図で表現し、開発者は簡単に画面遷移を設計することができます。

さらに、画面へのインターフェースとなる JSP に対しては豊富なタグ(netUI)が用意されていますので、Workshop を利用することで画面設計も効果的に開発することができます。

新技術【JavaPageFlow】で画面遷移を設計、管理が簡単

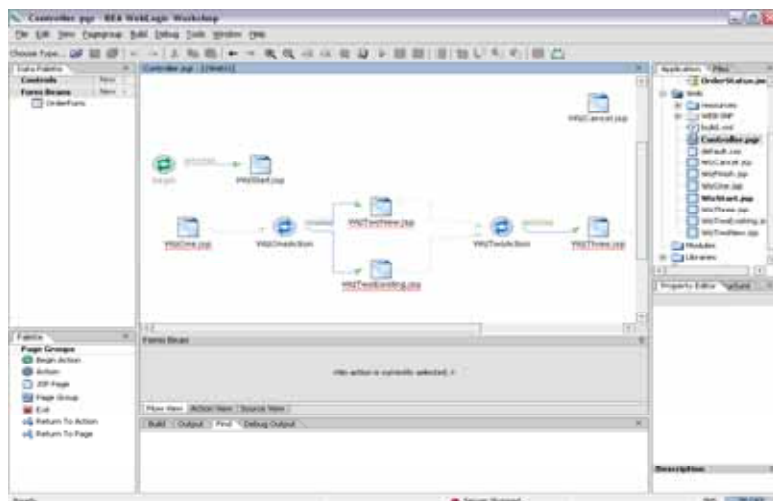
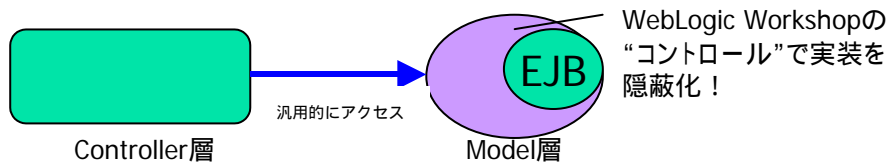


図 1 Java Page Flow 解説図

Workshop は、Model 層呼び出しの課題を Java コントロールという技術で解決します。Java コントロールは、EJB や JMS、JCA など技術要素を隠蔽し、単純なメソッドとして使うことを実現する技術です。

BEA システムズ社では、J2EE の技術要素だけではなく、RowSet、Database アクセス、File アクセス、Mail など企業システムに必要なと思われる汎用的なコントロールを多数用意しています。これらのコントロール群は、開発者を技術習得の手間から開放し、生産性向上に寄与します。

【コントロール技術】でサービス指向開発



【コントロール技術】のイメージ

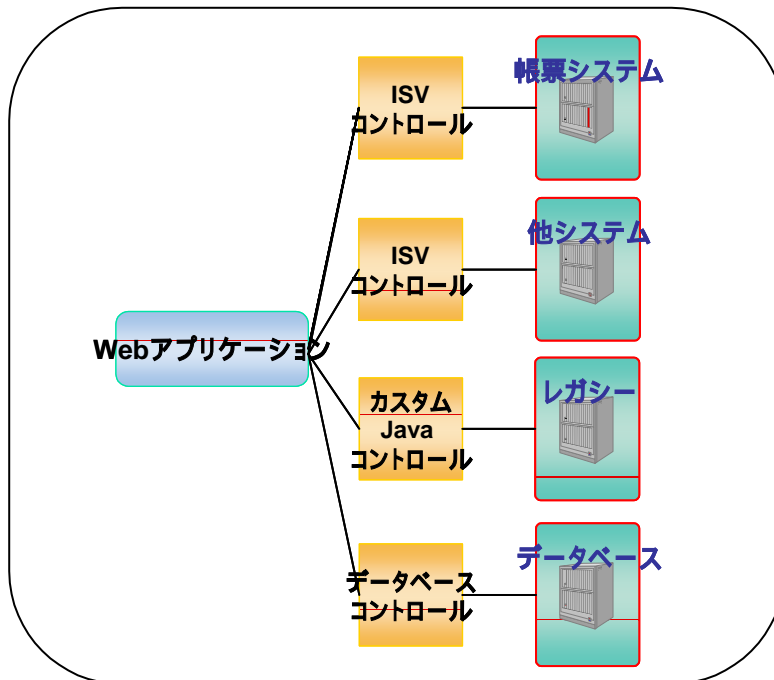


図 2 コントロール技術解説図

➤ Web システムにおける Biz/Browser の特徴

Biz/Browser では、従来の Web システムが実現し得なかった、高度な GUI を提供しています。その GUI は、キー入力の取り込みや入力文字 (IME 操作) を備え、Web システム隆盛前夜に普及していたクライアントサーバモデルの専用クライアントとほぼ同等の操作感を提供します。

また Biz/Browser は、サーバに標準の HTTP プロトコルでアクセスしますので、サーバサイドではその通信クライアントが Web ブラウザか Biz/Browser かを意識することなく汎用的に開発することが可能です。

➤ 両者の融合によるメリット

Workshop などを用いた WebLogic Server は、ページが変わってゆく画面遷移の管理 (水平) が容易であり、Biz/Browser はクライアントサーバとしての専用クライアント画面管理 (垂直) を得意としています。

現在の Biz/Browser にはクライアントへ事前に組み込む必要があること、OS は windows に限定されることの 2 点の制限がありますが、両制限が許容されれば、両者を組み合わせることで強力な Web システムを

開発できます。

例えば、メニュー画面から詳細な機能画面へ順次遷移する部分は JPF で設計し、サーバに対してクライアントサーバ的な動作をする画面は Biz/Browser で設計するという組み合わせを提案します。

両者の融合からは、次のようなメリットが期待できます。

- 画面遷移が JPF で表現されているため、わかりやすい
- 専用画面では、必要データのみを送受信となるためネットワーク負荷が軽減される
- クライアント画面では、ブラウザにはできない便利な GUI を提供できる

沖電気では、WebLogic Server と Biz/Browser の融合を検証し、協調動作することを確認しました。

Workshop や Biz/Browser は、それぞれ得意としている機能に特化して作りこまれている為、このような両者の組み合わせは弱いところを相互補完し、生産性が高まります。

次章にこの生産性を検証した結果を示します。

Biz/Browser の高い生産性について

➤ 生産性検証のモデル

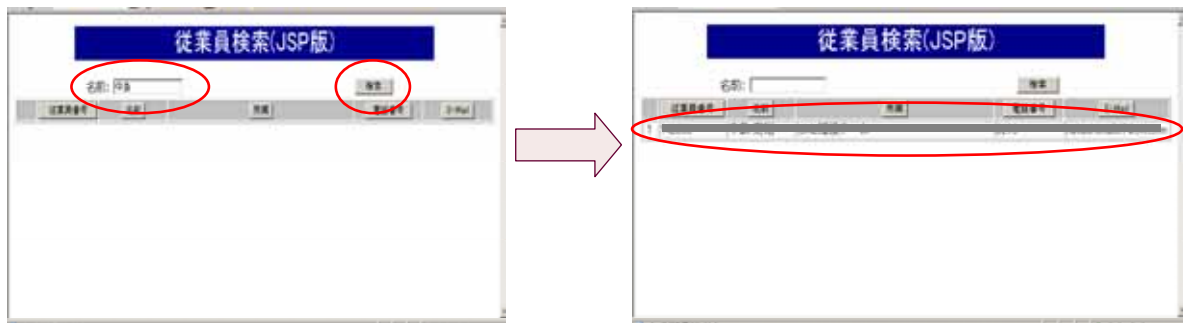
沖電気では、Biz/Browser の生産性がどの程度高いのかを客観的に検証しました。比較対照は J2EE システムにおいて、リッチクライアントを実現するのによく用いられる Java Applet および JSP(+Java Script)としました。

➤ 検証方法

JSP, Applet, Biz/Browser の 3 つの技術で、同じ機能を実装し、開発に要する時間や工数、ステップ数などで生産性を検証することとしました。

評価は 2 段階で行なうこととし、最初はスクラッチビルドの開発、次に仕様変更のケースを想定した影響を検証しました。

- **画面から入力された条件を基に、従業員DBから該当する従業員情報を検索し、結果の一覧を表示する**
- **条件を入力せずに検索ボタンを押すと、入力を促すダイアログを表示する**
- **結果一覧テーブルのカラム名をクリックすると、昇順にソートされる(追加開発)**



(対象機能)

- 従業員データベースに対して、従業員番号で検索をかけ、その結果を一覧で表示する機能

(評価対象技術)

- JSP
- Applet
- Biz/Browser

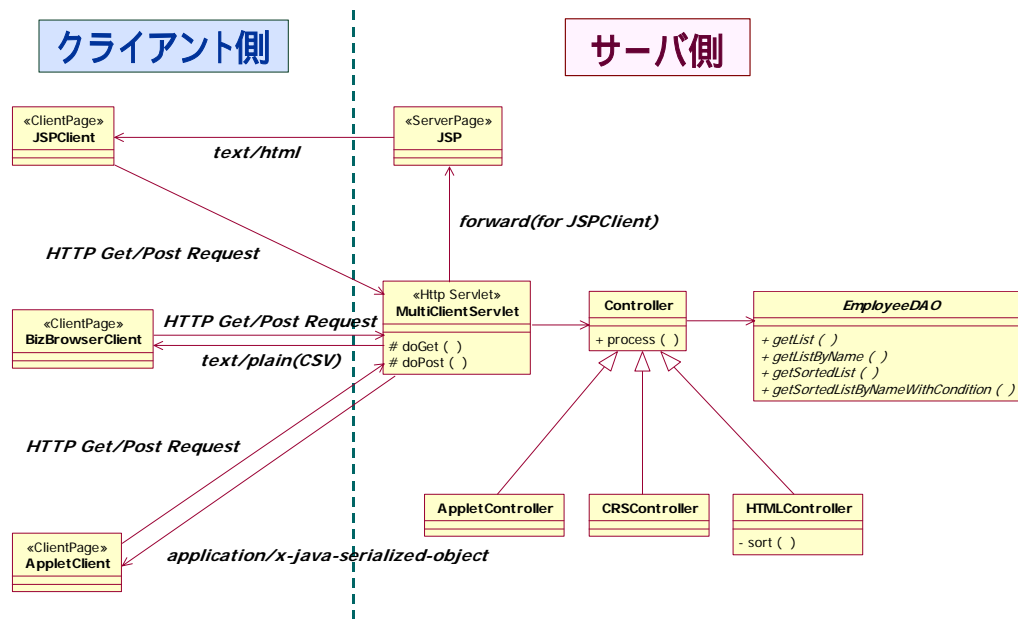
(評価指標値)

- 開発工数
- ステップ数
- ステップ数(非自動生成成分)

(評価の為の実装モデル)

WebLogic Server では、従業員 DB を検索する機能、画面遷移を管理する機能などを共通のアプリケーションとして J2EE で実装し、それぞれの技術にあわせたクライアント用インターフェースを用意しました。

JSP, Applet, Biz/Browser の各クライアント側アプリケーションでは、WebLogic Server から受け取ったデータを元に一覧表などの GUI を生成し、表示することとしました。



(*) 上図は、主要部のみの簡略化したもので、詳細部分は省略している

➤ 作成したシステムの生産性に関する考察

同一機能を Applet や Biz/Browser で実装比較した場合、Biz/Browser の生産性が高いことを確認しました。

表 1 開発に要した工数

	初期開発 (H)	ソート機能追加 (H)
Biz / Browser (CRS)	3	0.5
JSP & JavaScript	2	2
Applet	14	3.5

表 1 は、アプリケーションの作成に要した開発工数の比較です。

この表からは、Applet の生産性が目立って低いことがわかります。JSP と Biz/Browser ではほとんど類似の値でした。

また、仕様変更を模擬的に起こった“ソート機能の追加”については、Biz/Browser に適切な GUI 部品があったことから、0.5H という優れた生産性を確認することができました。

表 2 開発詳細比較表

		ファイル数	総ステップ数 (行) ^(*1)	コーディング 量(行) ^(*1)	コーディング 比率(%) ^(*2)
Biz / Browser (CRS) [*.crs]	初期開発	2 ^(*3)	86	14	16
	機能追加後	2 ^(*3)	93	19	20
JSP & JavaScript [*.jsp]	初期開発	1	62	62	100
	機能追加後	1	71	71	100
Applet [*.java]	初期開発	2	218	74	34
	機能追加後	4	343	199	58

表 2 は開発時のステップ数に関する比較を表にしたものです。この比較表では、Applet 開発環境や Biz/Browser の開発環境 (Biz/Designer) が自動生成した部分と、実際に人手でコーディングした部分の比率についても算出しています。

検証の結果、JSP (いわゆる Struts による MVC モデル) 開発の総ステップ数と Biz/Browser の総ステップ数は近似し、開発量はさほどかわらないこと、及び、Biz/Browser で実際にコーディングする比率は 16% と大半が自動生成されることが寄与し、生産性が高いことがわかりました。

Applet については、総ステップ数が Biz/Browser の約 4 倍、JSP の約 5.5 倍と総量に違いが生じることもわかりました。Applet はコーディング比率も Biz/Browser 比べて 34% と高く、ソート機能の追加をした際はコーディング比率が 58% とより高くなることがわかりました。

Applet によるリッチクライアントでは、機能追加、機能変更に対して既存のソースに影響が出やすいということがこの評価からわかりました。

今回は、さほど込み入ったリッチクライアントとしての機能を想定しなかったことと、表の表示機能が Biz/Browser の得意分野であることが Biz/Browser の高い生産性発揮につながった可能性があることを注意事項として上げておきます。

➤ JSP で作成した場合について

JSP はプログラムロジックと HTML の部分が混在するため、GUI のデザイナーが JSP を全て作成することは

難しいと考える。また、JSP 開発では、以下の点がハードルとして存在します。

- J2EE のスキルなどが要求される
- ビューとロジックの分離が不十分
- JavaScript はデバッグが困難

プログラムとしてある程度自在の表現が可能な反面、作りにくいという特徴があります。

Biz/Browser が実現するクライアント側の処理は、JSP ベースの場合、JavaScript で実装しますが、この技術はエラーメッセージの意味がわかりにくく、デバッグが困難です。エラーメッセージが出ずに何も反応しないこともあります。

動的な処理には JavaScript を使いますが、ブラウザによって挙動が異なることや、ソート機能はクライアント上で実現することが困難だったため、サーバ側のロジック（実際には RDB の機能）で実現せざるをえませんでした。この点は Biz/Browser が部品を用意している点と比較し、デメリットになります。

➤ Applet で作成した場合について

Applet の開発では、GUI ツールが補完するのは基本的なレイアウト・デザインのみ（テーブルの中身などは確認できない）で大半を実装する必要があることを確認しました。ツール自体の使用感もあまり直感的ではない（ツールに原因があるのではなく、Java の制約によるところが大きい）

Applet は基本的に Java プログラムそのものであるため、Java やオブジェクト指向に関する知識は必須である点がデメリットになると考えます。（要求されるスキルが高い）

また、OS、ブラウザ、Java Plug-in などの環境に動作がかなり左右されてしまうため、問題個所の切り分けが困難な場合があります。

しかし、ソート機能はクライアント側で対処でき、複雑なことで Java で実現できることは基本的に対処可能であり、高度なアプリケーション処理を要求される場合には、Applet が優位と考えます。

➤ Biz/Browser で作成した場合について

Biz/Designer は、直感的で分かりやすいため、デザイナーだけでレイアウト・デザインは簡単に行うことができると考えます。

コードの補完機能は使えるレベルにあり、工数削減が期待できます。通信部分に関しては、ある程度決まった形になるので、テンプレート化すれば高度な知識がなくても記述可能です。

ただし、デバッグに関しては、規模が大きくなるにつれ、困難になると予測されます。理由は以下の点です。

- サーバとの通信に関する詳細な情報が取れないため、どこで問題が起きているかを特定しにくい
- エラーメッセージが不親切
- 意味する内容が読み取れない
- マニュアルの記載が不十分
- 存在しないオブジェクトを使用しても、実行時にしか判明しない（文法チェッカでは検出できない）
- IE に依存するため、問題の切り分けが難しい

まとめ

今回の生産性に関する検証では、Biz/Browser の高い生産性を確認することができました。生産性が高い理由は次のような点にあると考えます。

- クライアントに必要な機能が部品として用意されている
- スクリプト自体は Microsoft 社の VisualBasic に似ているため比較的習得が容易
- Biz/Designer は直感的なツールで使いやすい
- サーバサイドとは単純な HTTP 通信でつながるので簡単

また、J2EE アプリケーションサーバとして高い性能を誇る WebLogic Server 8.1 には、Workshop と呼ばれる開発フレームワークがあり、Biz/Browser と組み合わせることで次のようなメリットが得られることを確認しました。

- MVC モデルで設計された Web による対話型の画面フローの中に、クライアントサーバシステムのような“きっちり”としたインターフェースが得られるメリット
- 画面上の入力制御(ダブルサブミット、画面への反映)を Biz/Browser に任せることで、WebLogic Server は業務処理、外部システムとの連携に集中できるメリット

参考：リッチシンククライアントの優位性

Web システムを構築する上でクライアントの選択肢の 1 つとして、ここではハードウェアシンククライアントとして Wyse 社の装置を紹介します。このシンククライアントに Biz/Browser をインストールすることにより、後述する「リッチシンククライアント」が実現可能です。

➤ 端末としてのシンククライアント

シンククライアントは一般に WBT(Windows Based Terminal)とも言われるディスクレスの端末であり、PC と比較して以下のような特徴があります。

- ・ハードディスクやフロッピーディスクがなくデータの持ち出しが不可能であるため、情報漏洩抑止の観点で優位性がある

- ・ディスクレス、ファンレスであり、駆動部分がないため故障が少なく、さらに消費電力も少ない

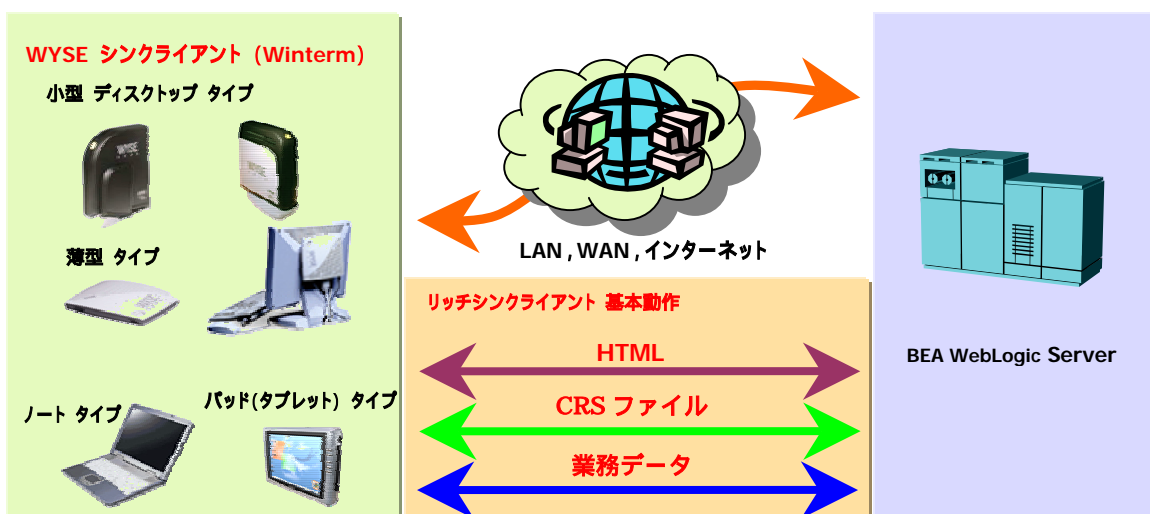
- ・サーバ側での集中管理/制御が可能であり、保守管理が容易になり TCO 削減にも繋がる

- ・サーバセントリックな使用を前提としており、PC に比べ陳腐化しにくい

このシンククライアントを Web ブラウザ 端末として採用する考え方は以前からありましたが、シンククライアント製品の多くが Server based computing 環境下で使用されることを想定し、処理能力、搭載 OS などを重視しておらず、Web ブラウザ端末としてあまり普及していませんでした。しかし、シンククライアントのワールドワイドトップメーカーである Wyse 社は、カスタムアプリケーションターミナル(CAT)やリッチクライアントの市場をカバーするために、マイクロソフトの最新 OS、大容量フラッシュメモリを搭載した機種の販売を開始しました。

➤ リッチシンククライアントとは

Wyse 社製シンククライアント Winterm シリーズは製品のラインアップが豊富であり Windows XPe/IE6.0、Windows CE/IE5.0 搭載の機種などがあります。さらに、それぞれの機種で異なるタイプ(デスクトップ型、薄型、ノート型、タブレット型など)があり、用途に応じて最適な機種を選択することができます。



Winterm シリーズには、ローカルストレージとしてフラッシュメモリを搭載した機種がラインナップされています。このフラッシュメモリに、リッチクライアントとしての Biz/Browser をインストールすることにより、リッチクライアントとシンクライアントの優位性を兼ね備えた「リッチシンクライアント」を実現することが可能となります。プレインストールされたソフトウェアを含め、バージョンアップ時にはサーバからダウンロードしてフラッシュメモリに書き込むことも、集中管理による遠隔からのフラッシュメモリ一括更新も可能です。

➤ リッチシンクライアント適用の優位性

セキュリティの強固さ、保守管理の容易性、故障・障害の少なさを特長とするシンクライアントに、高機能リッチクライアントである Biz/Browser を実装した「リッチシンクライアント」は、PC と同等の高機能と使いやすいユーザインターフェースを両立させており、今後のクライアントサーバシステムの Web システム化におけるクライアント形態の主流のひとつになると考えられます。

また、Wyse 社ではシンクライアントの管理ならびに、シンクライアントと PC が混在する環境下での端末管理ツールも提供しており、これらのツールを活用することにより保守管理の負荷をさらに軽減することが可能となります。

お問い合わせ先

BEA WebLogic Server、その他 BEA 製品については、以下の沖電気工業営業窓口までお問合せください。

沖電気工業株式会社 ネットビジネスソリューションカンパニー ソリューション営業部

Tel: 03-3456-5196 E-mail: nbc-info@oki.com URL: <http://www.oki.com/netbiz>

Biz/Browser シリーズ、Biz/Designer シリーズまたはその他の Biz-Collections 製品の情報については、アクシスソフト営業窓口もしくは Biz-Collections 製品パートナー営業窓口にお問合せください。Biz-Collections 製品パートナー情報については以下のホームページでご覧になれます。

アクシスソフト営業窓口

アクシスソフト株式会社

営業本部 Tel:03-5950-3522 E-mail:product-info@axisoft.co.jp

Biz-Collections 製品パートナー営業窓口および情報

<http://www.axisoft.co.jp/biz/bizpartner/index.html>

また、Biz/Browser シリーズ、Biz/Designer シリーズまたはその他の Biz-Collections 製品情報については以下のホームページでご覧になれます。

<http://www.axisoft.co.jp/biz/>