
開発支援フレームワーク 「オンライン処理方法選定指針」について

FUJITSU ファミリー会関西支部
2002 年度グループ研究会 A グループ

■ 執筆者 Profile ■



石原 博

- 1993 年 関電興業(株)入社
技術開発部情報システム課勤務 経理システム
(ホスト)運用
- 1998 年 工事業務支援システム
(クライアントサーバシステム)開発



丹羽 和弘

- 1999 年 荒川化学工業株式会社 入社
情報システム室所属 現在に至る



沼波 宏樹

- 1991 年 古河オートモーティブパーツ (株) 入社
システム部配属
受注・出荷システム開発を担当
生産技術システム開発を担当
ワイヤーハーネス設計CADシステム開発シ
ステム開発
生産準備効率化 PRJ. 参画

■ 論文要旨 ■

平成 14 年度ファミリー会関西支部分科会での活動が「Web 系オープンシステムにおける基幹システム構築」というテーマで行われた。我々A グループは9人のメンバで構成されており、それぞれの会社では次期システムの候補の一つとして Web システムの適用を模索していた。この論文では分科会での研究をもとに、今後各社でのシステム構築の基礎とするため、Web システムとはなにか・その方式にはどのようなものがあるのかを具体的な事例、長所及び短所を調査しまとめたものである。

■ 論文目次 ■

1. はじめに	《 4》
2. 今なぜWebなのか	《 4》
2. 1 メインフレーム	
2. 2 PC の出現	
2. 3 クライアントサーバシステム	
2. 4 Web システムの台頭	
3. Web システムの処理方法	《 6》
3. 1 2階層 Web システム	
3. 2 3階層 Web システム	
4. Web 開発事例	《 8》
4. 1 トラブル報告システム	
4. 2 介護サービス事業者(介護保険)向けASPシステム	
4. 3 出荷管理システム	
4. 4 各社事例のまとめ	
5. まとめ	《 10》
6. おわりに	《 13》

■ 図表一覧 ■

図1 2階層Webシステム構成	《 7》
図2 3階層Webシステム構成 (サブレット)	《 7》
図3 3階層Webシステム構成 (アプレット)	《 8》
表1 コンピュータの歴史	《 4》
表2 2階層Webシステム比較	《 10》
表3 3階層Webシステム比較	《 11》
表4 2階層型システムと3階層型システムの比較	《 12》

1. はじめに

平成 14 年度富士通ファミリー会関西支部グループ研究会の分科会活動は 9 名のメンバ(参考資料 1)によって約半年行われた。それぞれの会社ではホスト集中型のシステム、クライアントサーバ型システムのところ、Web システム導入済みと各社各様であったが、今後比較的規模の大きな基幹系システムの Web 化を模索していた。活動はほぼ 1 月に 1 回のペースで行われ、それぞれの調査結果を持ち寄り討議し研究を行った。

2. 今なぜ Web なのか

2. 1 メインフレーム

コンピュータの歴史はまだ 60 年ほどしかない。ここでは表 1 に従い、その歴史と Web システムに至る道を振り返ってみる。

表 1 コンピュータの歴史

1940 年代 コンピュータ発明 当初は軍用で弾道計算に用いられた。真空管やリレーを使用したものであった。
1950 年代 FORTRAN 開発 科学技術計算に用いられるようになった。
1960 年代 LISP, ALGOL, COBOL 開発 人工知能の研究や、事務処理にも使用され始める。
1970 年代 始め 4004(4 ビット CPU) 後半 8086(16 ビット CPU) 開発 電卓用の IC から始まったが、またたく間に処理能力を増加させた。 メインフレームを中心とした構成での集中型システムが主流となる。
1980 年代 IBM-PC 及びその OS である DOS 発売 現在の PC の始祖。さまざまな PC 上のアプリケーションが発売。 UNIX マシンの登場。PC・UNIX を中心とした分散型システムが主流となる。 1989 年にスイスの CERN において、Web が開発された。
1990 年代 Windows 発表・インターネット急激に拡大

コンピュータが開発された当初、それは非常に高価な装置であった。複雑で繊細なハードウェアのために特別な空調付きの部屋に設置し特定の人しか触れることができなかった。また貴重な CPU タイムを無駄にしないように、入出力に紙テープはパンチカードを使用したバッチシステムであった。1970 年代に TSS(タイムシェアリングシステム)が開発され複数の人が同時に使用できるようになったが端末としてはダム端末や、VT100 に代表されるような(カーソルコントロールができる)インテリジェント端末であった。これがメインフレームシステムである。1970 年代から 1980 年中頃までは、すべてのデータ管理・業務処理・事務処理をメインフレームで行い、処理機能のないダム端末でデータの入力や操作をするという集中型システムが主流であった。

2. 2 PCの出現

しかし別な方向からの変化があった。1970年代に端を発するPCの台頭である。はじめは電卓に毛の生えたようなものではあったが、AppleIIのVisiCalc(スプレッドシート)の成功によってPCがホビーだけでなくビジネスでも使用できるということがわかった。これはMultiPlanやLotus123, MS-Excelと引き継がれ今でも広く使用されている。

ユーザはメインフレームシステムとPCに落差を感じていた。かたや白黒の画面のメニュー形式であり、かたやカラフルな画面、ユーザフレンドリなプログラム(スプレッドシートやワードプロセッサに代表されるシステム)である。後者は市場規模からも複数メーカーがしのぎを削り開発しており、初心者を含めたユーザのニーズに合致したものが次々と発表された。更に1人に1台のPCが割り当てられ、メールやグループウェアが広まる状況で、ダム端末やインテリジェント端末は時代遅れと認識されていった。

このように、表計算ソフト、ワープロソフトなどの普及によって、それまで高価なメインフレームで行っていた処理を、比較的安価なPCに分散させて行う分散型システムが普及するようになった。小型、安価でコストパフォーマンスのよいシステム構築を行うダウンサイジングの動きは、80年代後半から90年代前半、不景気であった米国企業において情報処理コストを削減すること、データベースの共有・分散によるシステムの効率を向上する目的で進められ、日本企業でもこの考え方が多く導入されるようになった。ダウンサイジングの具体例として、CSS(クライアントサーバシステム)が登場することとなった。

2. 3 クライアントサーバシステム

CSSはUNIXサーバ・UNIXクライアントに端を発し、クライアントはよりグラフィカルなWindows3.1へと移行していった。そして1990年代にはマイクロソフトがNTサーバを発表し、Windowsもバージョン95でTCP/IPを標準で使用できるようになった。これによってネットワーク化が急激に進行した。これらのシステムは元々個人使用のPCをネットワークで接続し、データやプリンタを共有することから始まったものであるため、ファイルサーバ・プリンタサーバ・データベースサーバなどを中心のサーバとするCSS(クライアントサーバシステム)となった。CSSのメリットとして、クライアント上でアプリケーション層の一部、プレゼンテーション層が動作するため、その修正、メンテナンスなどの対応がスピーディーかつ柔軟に行えるようになったことがあげられる。

しかし一方ではこのようなCSSには問題があった。比較的複雑で重いユーザインターフェースを中心にアプリケーションの一部をクライアント側で処理するため、クライアントの台数が増えれば、プログラムのインストールの手間・バージョンアップの手間がかかること、特にWindowsは日に何度もハングアップしたり、あるいはユーザの操作ミスによってクラッシュするような不安定なOSであるため全体の信頼性が著しく劣化することなどである。「リース費用や保守費用が高額なメインフレーム集中型システム」から、「安価なPCで利用できるオープン系資源が充実し、多種多様なニーズへの対応可能なCSS」というのが流行となったが、実際には管理コストの増大を招くことになってしまった。

2. 4 Webシステムの台頭

このころまた別の方向から発展があった。1990年代始めにネットワーク上のコンピューターの情報共有を目的としてWWW(World Wide Web)システムが開発されたのである。

これはコンピュータに保存する情報を記述するための言語（HTML）、情報をやり取りするための通信手段である HTTP プロトコルから成っている。それまではインターネットに接続できるのは、大学や大企業の研究機関のみでありメールや FTP が中心であった。それ以外のユーザは通信といっても NiftyServe、PC-VAN などの BBS でメールやテキスト中心の掲示板程度であった。しかし Windows95 を使用することで比較的容易に TCP/IP でネットワークに接続できるようになったこと、インターネットプロバイダの出現によって個人でもインターネットに接続できるようになったこと、WWW によってグラフィカルな画面が非常に簡単にみられるようになったことなどによってインターネットの利用が急激に広まることになった。

これによって WWW をみるためのソフトであるブラウザの巨大な市場が現れた。そのため Mosaic に端を発し Netscape Navigator や Internet Explorer と言ったブラウザが開発され低価格化・高機能化・高速度化が進んでいく。今や Windows 系の OS にはブラウザが標準でインストールされている状態である。

このブラウザを一種の端末として使うことを考えたのが Web システムである。標準化され Windows でも UNIX でも使用できる上に安価(ほぼ無料)でありながらグラフィックベースでマウスを使えるブラウザ、そして広く使われることで洗練され安定な動作をする Web サーバ(例えば Apache)をプレゼンテーション層とし、従来のホスト的な業務を行うアプリケーション層と接続することでクライアントの負担を減少させ、管理コストを下げるのが目的である。つまりメインフレームの集中管理と CSS の柔軟性をもつシステムといえる。次章からこの中の Web システムを中心に報告を行う。

3. Web システムの処理方式

3. 1 2階層 Web システム

Web システムはブラウザ・Web サーバ(プレゼンテーション層)及びロジック部分(アプリケーション層)、データベース(データベース層)から構成される。最も単純なのが一つのサーバで、すべての処理をおこなってしまう 2階層構造である(図 1)。

WEB サーバに JAVA サーブレット・JSP、または ASP を置き、クライアントのブラウザと HTTP プロトコルにて会話をを行う。2階層システムでは Web サーバから呼び出されるアプリケーションが画面上のどの場所にどんな情報を表示するのかを HTML によって明示的に指定する必要がある。プレゼンテーション層とアプリケーション層が分離していないため動作は比較的単純で導入は容易ではあるものの、その反面逆に処理量に応じてシステムを構築していくといった拡張性に乏しい。このような特徴から 2階層型は小規模なシステム、WEB から DB 情報の照会システムなどに向いているといえる。

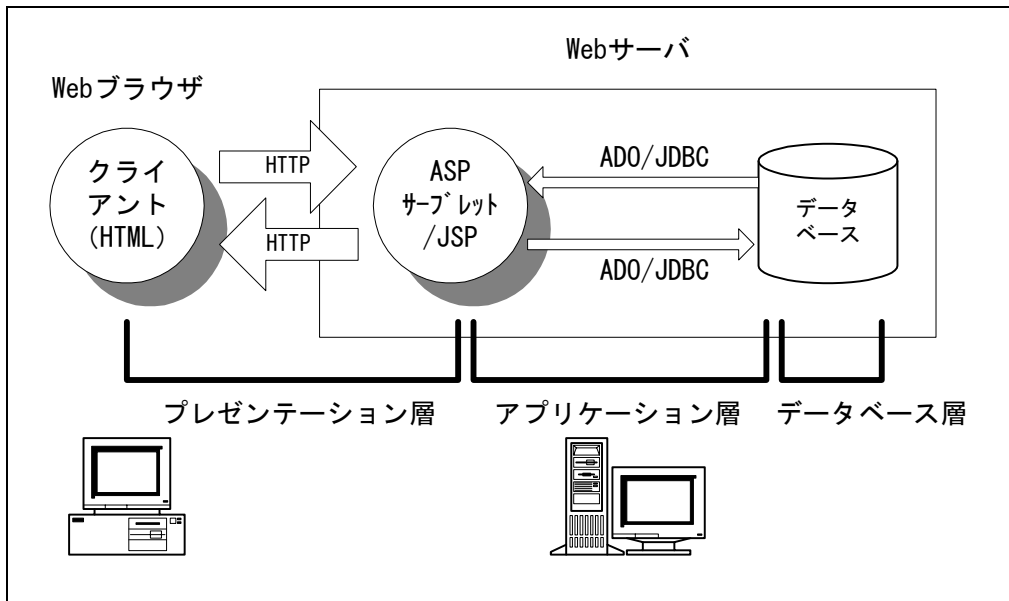


図 1 2階層 Web システム構成

3. 2 3階層 Web システム

これに対しプレゼンテーション層とアプリケーション層を分けたものが3階層型のシステムである。これはロジック部分と画面上の表示部分を分離することでプログラム作成時の作業分担などのメリット、ロジック部分の変更がインターフェースに影響を及ぼしにくくなるというメリットがある。(図2)

WEB サーバでは JAVA サブレット・JSP によってプレゼンテーションロジックのみを処理し、アプリケーションロジック、データベースロジックは分離されたアプリケーションサーバで処理されることになる。このため処理量に応じてシステムを構築していくといった拡張性が高く、複雑な WEB トランザクション処理も可能となっているのが特徴である。このタイプは、インターネットトレーディングシステムなどのパッケージ化されたサーバアプリケーションや、既存システムとの組み合わせも可能な効果的な形態となっている。

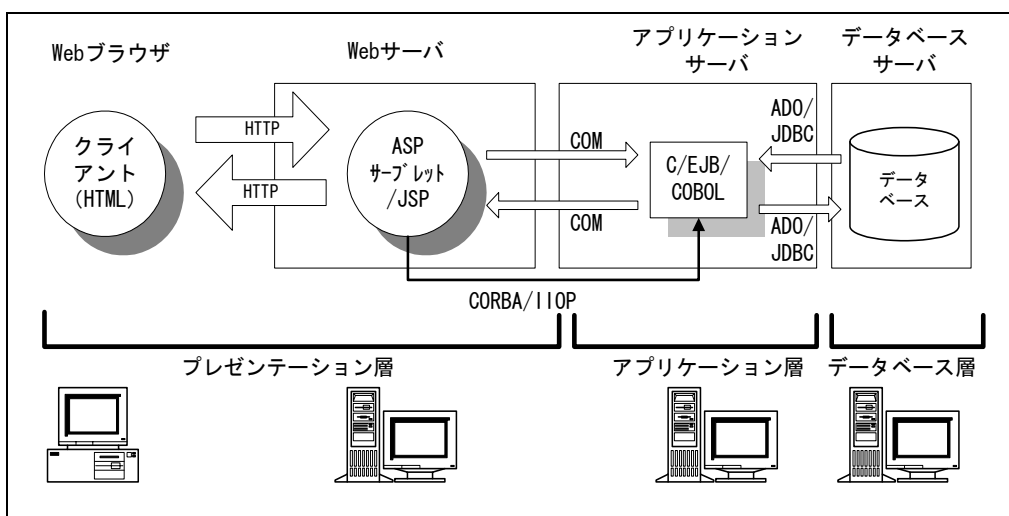


図 2 3階層 Web システム構成 (サブレット)

また、3階層型には、クライアント側の JAVA アプレットで処理するタイプもある。
 (図3) これもまた同じく機能分離型だが、WEB サーバのサーブレットの代わりにクライアントにアプレットを配置している点が異なる。プレゼンテーションロジックは、クライアントの JAVA アプレットで処理され、WEB サーバは HTML と JAVA アプレットの格納サーバになり、クライアントは必要に応じて JAVA アプレットを WEB サーバからダウンロードして実行する。実行中のアプレットは、アプリケーションサーバに処理要求をあげ、結果を受け取るクライアントサーバ形態となる。JAVA アプレットは、クライアントのブラウザで必要なときに WEB サーバからダウンロードされ、実行されるので、CSS 型に見られるようなクライアントの資源配布を考える必要はない。JAVA アプレットは、凝った画面設計を要求されるシステムを構築でき、かつ3階層タイプなので、複雑な WEB トランザクション処理も可能となっているのが特徴である。このタイプは、クライアントサーバシステムの発展形態で、WEB 活用の新営業店システムといった、高度な GUI を要求されるシステムに欠かせない形態となっている。

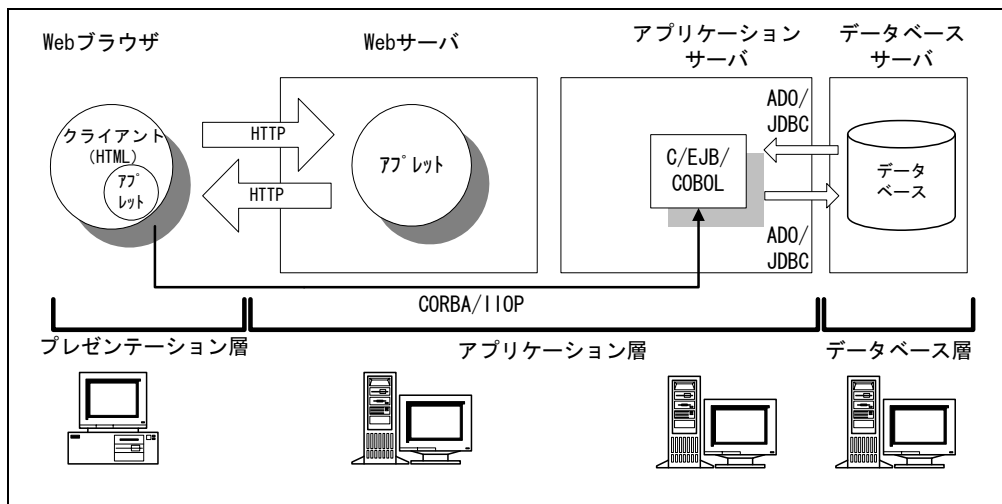


図3 3階層 Web システム構成 (アプレット)

4. Web 開発事例

インターネットでは(Web での情報発信が主であること、性能や安定性、セキュリティが何にもまして重要視されることもあり)UNIX+Apache+Perl がデファクトスタンダードといえる。しかしイントラネットと呼ばれる社内で限定されたネットワークの範囲内で公開されるものについては社内の運用の問題などから Windows が使用されることもある。ここでは実際の各社事例の調査を行った。我々のグループにおいては基幹系システムを Web にて構築した事例はなく、以下の3事例はすべて2階層 Web システムでの事例である。以下 1) システム概要、2) 使用した Web サーバ、3) Web アプリ、4) データベース、5) 選定理由、6) その他特記事項、をそれぞれの事例に紹介する。

4. 1 トラブル報告システム

- 1) トラブル発生時に報告，進捗管理を行うシステム．機能として報告内容に対し，技術的な回答，部品手配，施工報告の登録及び実績の検索が可能となっている．
- 2) Windows NT+IIS
- 3) ASP
- 4) MS SQLServer
- 5) 開発が短納期であり更に ASP を使用した開発に前例があったため
- 6) 規模は，同時接続台数 30 台程度．

4. 2 介護サービス事業者(介護保険)向け ASP システム

- 1) 介護サービス事業者が利用者に対して行ったサービスに対して請求処理などの事務処理を行うシステム．中小サービス事業者向けのシステムであり，高価なパッケージソフトを購入しなくてもWeb環境が整備されていれば利用できるようなASPで構築された．
- 2) Windows2000+IIS
- 3) ASP
- 4) MS SQLServer
- 5) Windows 2000 Server と MS SQLServer があれば即座にシステム構築ができる点，更に使用開発言語が JScript, VBScript で記述することによって，言語の習得も早く，開発が容易であったため
- 6) 規模は，同時接続台数でピーク時最大 30 台程度．

4. 3 出荷管理システム

- 1) 製品出荷情報（出荷予定日・出荷日・出荷製品の詳細内容・出荷状況（進捗管理など）を，条件検索して情報閲覧ができる参照システム．
- 2) Apache
- 3) JAVA サブレット，JSP
- 4) DB2
- 5) 既存の CSS で開発された出荷管理システムのデータベースを WEB システムでも使用し，CSS のクライアントプログラムがインストールされていないクライアントでも特定の情報をみることができる参照系システムをユーザから要求された点，更に将来的には更新系システムを含め，CSS のすべての機能を WEB システムでも実現するため，3階層システムへの移行，WEB トランザクション処理の導入といった将来の拡張性も考慮したため
- 6) アプリケーションサーバに Websphere を適用．

4. 4 各社事例のまとめ

各社でのシステム構築のコンセプトとなっているのは，何れも“より早く”“より安く”“よりよいもの”をである．これらのコンセプトを実現するために，I I S ・ A S P を用いた2階層システムでの開発は手軽であり，開発のしやすさ（難易度），安価・スピード対応，に適していると各社で判断されたため構築実施となった．これらのシステムは各

社で細かな問題を抱えながらも有効に活用されており、事例で考察する限り、参照系システムでかつ小規模システムであるなら2階層システムでも構築・運用は可能であるという目安になる。

5. まとめ

2階層型と3階層型の特徴を整理するための比較表を作成した。表2では代表的な2階層型システムの中から、サーバでASPやPHPなどスクリプトで処理する「サーバサイドスクリプト型」と、サーバでJAVAで処理する「サーバサイドJAVA」とを選び比較を行った。

表2 2階層 Web システム比較

	2階層		
	サーバサイドスクリプト		サーバサイドJAVA
	ASP	PHP	JSP/Servlet
特徴(開発性)	・HTMLにスクリプトを埋め込めるので作成が容易 ・ソースを変更するとすぐに反映されるので、デバッグが容易		・HTMLにスクリプトを埋め込めるので作成が容易
特徴(処理)	・スクリプトを実行するので、複雑な処理は、おそくなる		・コンパイル結果を実行するので、スクリプトに比べ処理が高速である
HTTPリスナ	IISのみ	Apache, IIS	Apache, IIS
プラットフォーム	Windows	UNIX, Windows	UNIX, Windows
使用できる言語	ASP	PHP	JAVA

2階層ではサーバでASPやPHPなどスクリプトで処理する「サーバサイドスクリプト型」と、サーバでJAVAで処理する「サーバサイドJAVA」とに分けて比較した。

サーバサイドスクリプト、サーバサイドJAVAともにHTMLにスクリプトを埋め込めるので作成が容易である。処理速度はサーバサイドスクリプトではスクリプト言語で記述されているため比較的遅く、サーバサイドJAVAはコンパイル結果を実行するため速い。またサーバサイドJAVAでは使用言語にJAVAを使用しているため、後に述べる3階層型への移行が比較的スムーズにできるというメリットがある。

その他の特徴としてはPHP・JSP/ServletではHTTPリスナとしてApacheとIISが使えるのに対して、ASPではIISのみに限定される。このためプラットフォームはPHP、JSP/ServletではUNIXもWindowsも使用できるが、ASPに関してはWindowsに限定される。

しかしWindowsではWEBサーバであるIISが標準で付属するためASPで「とりあえず作ってみる」というのであれば非常に手軽であるといえる。

ただ、一方でWINDOWS系のOSは一般的にセキュリティ、安定性の面でUNIX系のOSよりも弱い。手軽なOSを採用するのか、安定性でUNIX系のOSを採用するか（UNIX系であればASPは必然的に選択肢から外れる）といったことも比較検討の一要因となる。

次に表3に3階層型の比較表を示す。3階層型では、サーバでJSP/Servletで処理する

「サーバサイド JAVA 型」と、クライアント側の Java アプレットで処理する「クライアントサイド JAVA 型」に分けられる。ここでは二つのタイプを比較している。

表3 3階層 Web システム比較

	3階層	
	サーバサイドJAVA	クライアントサイドJAVA
	JSP/Servlet + COBOL/C/EJB	Applet + COBOL/C/EJB
特徴(開発性)	・アプリケーションが独立しているので、複雑なWebトランザクション処理が可能	・アプリケーションが独立しているので、複雑なWebトランザクション処理が可能 ・Appletを使うので、画面操作性に優れる
特徴(処理)	・アプリケーションの処理を分離することにより、多数のトランザクション処理が可能 ・コンパイル結果を実行するので、スクリプトに比べ処理が高速である	・アプリケーションの処理を分離することにより、多数のトランザクション処理が可能 ・コンパイル結果を実行するので、スクリプトに比べ処理が高速である ・Appletのダウンロードに時間がかかる
HTTPリスナ	Apache, IIS	Apache, IIS
プラットフォーム	UNIX, Windows	UNIX, Windows
使用できる言語	COBOL, C, JAVA	COBOL, C, JAVA

サーバサイド JAVA, クライアントサイド JAVA とともに、アプリケーションの処理を分離することによって、多数のトランザクション処理が可能となる。クライアントサイド JAVA ではそれに加え、アプレットを使うため画面操作性に優れている。しかしクライアントサイド JAVA の場合はアプレットを最初にダウンロードするのに時間がかかる問題がある。またアプレットを利用するためブラウザの種類、バージョンによって動作しないなどの問題が起こる可能性もある。

また、サーバサイド JAVA, クライアントサイド JAVA とともに、コンパイル結果を実行するので、スクリプトに比べて、処理が高速である。HTTP リスナ、プラットフォーム、使用できる言語に関しては、いずれも広く一般的に利用されている HTTP リスナ、OS、言語で実現可能となっている。

ここでのポイントはプレゼンテーションロジックがサーバサイドにあるかクライアントサイドにあるかということである。クライアントサイドにあれば、それだけ複雑な画面制御が可能となりユーザに扱いやすい画面を作成することができるがアプレットのダウンロードに時間がかかるという問題、動作条件が厳しくなりがちといった問題がある。つまりクライアントサイド JAVA はユーザが限定されており、回線速度がある程度保証された環境であるならば推奨できるが、そうでない場合には少々厳しいものがあるといえる。

次にシステムの構成を決定する上で考慮すべきポイントを洗い出し、それぞれの構成について表4にまとめた。

表4 2階層型システムと3階層型システムの比較

	2階層			3階層	
	サーバサイドスクリプト		サーバサイド JAVA	サーバサイドJAVA	クライアントサイド JAVA
	ASP	PHP	JSP/Servlet	JSP/Servlet + COBOL/C/EJB	Applet + COBOL/C/EJB
システム規模 (同時接続台数)	△(～50)	△(～50)	△(～50)	○(～1000)	○(～1000)
導入コスト	○	○	○	×	×
開発コスト	○	○	○	△	△
運用コスト	○	○	○	△	△
開発難度	○	○	△	×	×
開発期間	○	○	○	△	△
サーバー負荷	△	△	△	○	○
拡張性	×	×	△	○	○
汎用性	△	△	△	○	○
更新の向き・不向き	△	△	△	○	○
レスポンスの安定性	△	△	△	○	○
ネットワーク負荷	○	○	○	○	△
クライアント画面の柔軟性	△	△	△	△	○

システムの規模、同時接続台数では WEB トランザクション処理の可否によって2階層構成では50台ぐらいまでが限界であり、また、3階層構成では1000台ぐらいまでと3階層構成が圧倒的に有利である。同様にサーバ負荷、拡張性、汎用性、更新の向き・不向き、レスポンスの安定性といった点に関しても、アプリケーション層の独立している3階層のほうが有利となる。

導入コスト、開発コスト、運用コスト、開発難度、開発期間といった、コスト面、開発の容易さといった点では2階層のほうが有利で3階層は構成が複雑である為、コスト、開発面では不向きとなっている。なお開発難度の点ではサーバサイド JAVA は開発に JAVA を用いるため、スクリプトに比べると若干難しくなる。

またネットワーク負荷については2階層と3階層のサーバサイド JAVA では、クライアントと HTML で会話するので比較的負荷は軽いですが、3階層のクライアント JAVA は、アプレットのダウンロードや、クライアントがアプリケーションサーバと直接やりとりするため少し負荷は高くなる。

一方クライアント画面の柔軟性ではアプレットでは VB などで作成した従来の CSS 並の高度な GUI を組めるため、HTML のみに比べて有利となる。

結論として処理方式の選定には、サーバの OS (UNIX なのか Windows なのか)、開発者のスキル(使用言語の経験)、予算なども考慮する必要があるものの、基本的には同時接続台数が比較的少ない社内のデータ照会系のシステムなどでは2階層型が推奨され、また頻繁にデータのオンライン更新を行う基幹系システムや同時接続台数が多い BtoB, BtoC などの WEB システムでは3階層型が推奨される。

更にブラウザがある程度統一でき、データの通信速度が保証された環境化でかつ高度な GUI が要求されるような場合、3階層 Web システムの中でも特にアプレットを利用したものが推奨される。逆にそういった環境(通信速度・ブラウザの統一)が確保できない、または高度な GUI を必要としない場合にはサーバレットを選択した方がよい。このため、例

えばクライアントがある程度限定でき、画面内でのエラーチェックなどが要求される基幹系システムや、一部の BtoB システムではクライアントサイド JAVA が、逆に不特定多数をクライアントとする BtoC や BtoB のシステムではサーバサイド JAVA が向いているといえる。

6. おわりに

今回の研究にはインターネットでの検索を多用した。一昔前には考えられないような便利な環境である。しかしその情報には善し悪しは別に現実にバイアスがかかっており取捨選択は個人にまかされている。特に今回の例で言えば Apache, Perl, JAVA などの一般的な情報については玉石混合状態ではあるものの多量の情報が得られたが、3階層構造のシステムについては概要か企業の宣伝程度の情報しかなく苦勞した。これは基幹系の事務処理というインターネットでは一般的ではないものであり、また個人が使用するようなものではないためと思われるが調査不足は否めない。

しかし3階層 Web システムが一般的に広がるのはこれからである。今回の調査研究によって「今なぜ Web なのか」を我々なりに明らかにすることが出来たと思っている。今後自社の基幹系システムの Web アプリケーション化を検討する際には今回の経験を有効に生かして行きたい。

最後に、今回の分科会活動でご指導頂いた担当幹事 日比野様・森田様、アドバイザー 篠木様、事務局 尾初瀬様のご支援に対し深く感謝申し上げます。

参考資料 1.

平成14年度富士通ファミリー会関西支部グループ研究会分科会Aグループメンバ

・小林 浩	アトラス情報サービス株式会社
・丹羽 和弘	荒川化学工業株式会社
・石原 博	関電興業株式会社
・清水 純	株式会社セントラル情報サービス
・平井 裕	タイガー魔法瓶株式会社
・稲森 啓志	ニッタ株式会社
・大西 良和	フジテック株式会社
・沼波 宏樹	古河オートモーティブパーツ株式会社
・佐伯 幸史	ヤンマー情報システムサービス株式会社