

キーワードで理解する 最新Web開発動向 2006

昨今、耳にすることの多いWeb 2.0、Ajax、EoD、Web API等々...さまざま
なキーワード、BuzzWordになりがちなこれらキーワードの本質を、皆
さんは理解していますか？
これらキーワードを通じて、本セッションでは、Web開発の最新動向に
ついて紹介します。



山田祥寛 (YAMADA, Yoshihiro)

CQW15204@nifty.com

<http://www.wings.msn.to/>

IT世界にますます氾濫するBuzz Word

■ 踊るのか、踊らせるのか Buzz Word

- Web開発の世界でもBuzz Word(= 流行語、空騒ぎ) はますます増えている
- Ex. Web 2.0、Ajax、Web API、SOA、EoD、アジャイル開発などなど
- 単にBuzz Wordに踊らされるだけでは、その技術の本質を見失ってしまう
流行になってしまうと、なかなか他人には聞きにくくなってしま
う
なんとなく聞いているうちに解った気になってしまう
よく理解しないままに使ってしまう(Ex. 「これってWeb2.0的だよね」)。
なんとなくキーワードだけが流行する悪循環

主要なキーワードの本質を把握してみましょう

Keyword-00) 本当に理解していますか？ Web 2.0 (次世代インターネットの方向性)

■ そもそもWeb 2.0とは...？

- 技術や仕様のバージョンではない Cf. Tim O'Reilly氏「What is Web 2.0」
Web活用のための次世代のコンセプト / 方向性 (技術 / ユーザ / マーケティング)
- Web 1.0: 静的なHTML。更新頻度も稀
- Web 1.5: CMS等を利用した動的なページ。サイト内で完結したサービスを提供

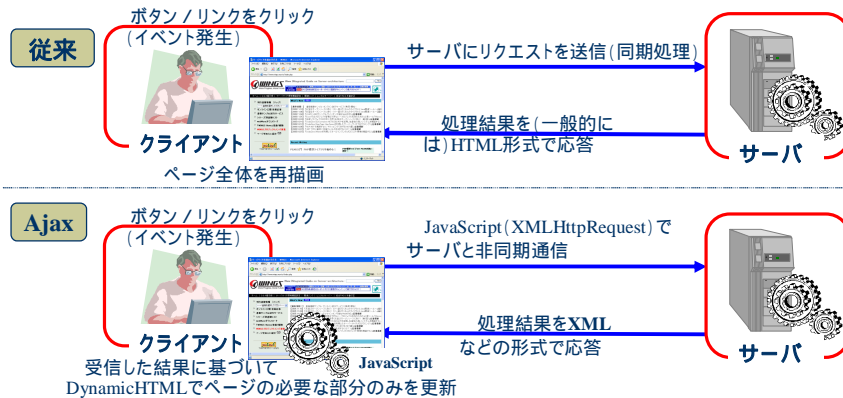
■ Web 2.0を構成する要素

- プラットフォームとしてのWeb
コンテンツ同士のシンジケーション / リミックス (Web API / RSS)
- ユーザ参加型のアーキテクチャ
ユーザの参加によってデータの価値が増す (集合知 / 性善説)
Ex.) ブログ / SNS / Amazon Review / WikiPedia
- サービス中心のアーキテクチャ
サービスにアクセスするための「直感的なUI」 (Ajax / XML)
- 永遠のベータ版 (日常的なフィードバックとバージョンアップ)
リリースして終わり、ではない。ユーザは常にテスター。
軽量プログラミングモデルの必要性 (EoD / REST)

Keyword-01) 枯れた新しいUI革命 Ajax (Asynchronous JavaScript And Xml)

■ Ajax (Asynchronous JavaScript And Xml)

- JavaScriptでサーバとの非同期 (Asynchronous) な通信を行い、その処理結果に基づいて、ページ上の必要なコンテンツだけを動的に更新する手法



What is Ajax ?

～ Ajaxは枯れた技術の集合体 ～

■ What is Ajax?

- JavaScript(XMLHttpRequest) + DHTML + XML
枯れた技術の集合体(新しい要素はない)
- Ajax技術を使うことで、たとえばこんなアプリが...

サイト名	URL
Google Maps	http://maps.google.co.jp/
マウスによって表示位置を移動したり、拡大 / 縮小率を変更できる地図情報サービス	
Windows Live	http://www.live.com/
RSS/Ajaxをフル活用したパーソナライズ可能なポータルページ(ニュースや天気、検索機能などを提供)	
Amazon.com Diamond Search	http://www.amazon.com/gp/search/finder/?productGroupID=loose%5fdiamonds
ダイヤモンドの商品検索システム。形状やカラット数、価格などの条件に応じて商品を動的に表示	
The Web Word Processor	http://www2.writely.com/info/WritelyOverflowWelcome.htm
ブラウザ上で動作する英文ワードプロセッサ	



Ajax技術の導入効果

■ ユーザビリティの向上

- ページ内の必要な箇所のみをリフレッシュ
Webアプリにありがちな通信ごとの画面のチラツキはなし
- サーバ処理の間もクライアント側では操作を継続できる
Asynchronous(= 非同期)な処理によって作業の中断を抑制
- 定期的なサーバサイドとの通信が可能になる

■ パフォーマンスの改善

- 画面全体を書き換える必要がない(必要なコンテンツのみを更新)
クライアント-サーバ間のトラフィック量を抑制
- サーバサイドへの負荷をクライアント側に分散

■ アプリケーション開発におけるアドバンテージ

- 特別なプラグインを導入することなく、リッチコンテンツを実現できる
- 標準的なJavaScript(DHTML)やXMLの知識のみで開発が可能
開発者が新しい技術を習得する必要がない

Ajax技術は本当に使えるか？ ～ Ajaxプログラミングの問題点～

■ クロスブラウザ問題

- ブラウザ間でのJavaScript/CSSの挙動の違い
旧来に比べると状況は改善しているが、クライアント依存は存在

■ 従来のWebアプリケーションと挙動がかけ離れていないか

- 現在の状態をブックマークできない(画面完結型なので、URLが変化しない)
- 現在の通信状態がユーザの目には分からない
読み込み中なのか、処理が完了したのか、更新されたのはどこか？

■ 開発生産性の問題

- JavaScriptの統合開発環境は不在(複雑なコードをどこまで記述するのか)
- XML DOMによるデータ処理はコードが冗長になりがち

■ その他

- セキュリティ上の課題(よりクライアントに裁量が委ねられている)
XSSやSQLインジェクションには要注意
- 複雑なJavaScriptはライブラリダウンロードと初期化処理によるオーバーヘッド

現実的なAjax技術活用のために (1)

■ データ交換言語をXMLにこだわる必要はない

- YAML (YAML Ain't Markup Language)
記号とインデントで構造を表現(汎用性はXMLに劣るが、可読性で有利)

```
Books:
-
  title: XMLデータベース入門
  publish: 翔泳社
```

- JSON (JavaScript Object Notation)
JavaScriptの(連想)配列形式。JavaScriptとの親和性、高速性
ただし、複雑な構造を扱うには不利

```
{"books": [
  {"title": "XMLデータベース入門", "publish": "翔泳社"},
  {"title": "書き込み式SQLのドリル", "publish": "ソシム"}
]}
```

- プレーンテキスト/HTMLでも可(ただし、再利用性は落ちる)

現実的なAjax技術活用のために (2)

■ Ajaxは入力 / 操作支援の目的で部分的に採用するのが効果的

- Ajaxはアプリケーション開発の一手法にすぎない
- 本当になにからなにまでAjaxが必要なのか？
Google Mapsのような例は、特別なケースであると考えべき
- 開発生産性 / パフォーマンスの理由から、Ajax技術でアプリケーション全体を構築するのは非効率
- やりたいことはなにかをもう一度考えよう

現実的なAjax技術活用のために (3)

■ Ajax対応の統合開発環境も続々登場

- 汎用IDEに主にAjax対応コンポーネントを提供 / 同梱したものから、Ajax開発に特化したもの、JavaScriptデバッガのような特定目的の開発環境まで

製品名	URL
Oracle JDeveloper 10g Release 3	http://www.oracle.co.jp/tools/jdeveloper/
Sun Java Studio Creator 2	http://developers.sun.com/jscreator/
Ajax Builder	http://www.hows.ws/product/
Venkman JavaScript Debugger	http://www.mozilla.org/projects/venkman/

- Microsoft Visual Studioも将来的には対応の予定 (Atlas Framework)

■ Ajax対応ライブラリ&フレームワークも積極的に活用しよう

Ajax対応フレームワーク & ライブラリの分類

■ 機能特化ライブラリ型

- オートコンプリートやタブペインなどの特定機能をAjaxで実現するためのライブラリ
- Ex. AjaxTags, AjaxFacesなど

■ 汎用ライブラリ型

- Ajax開発を支援する、より汎用的なライブラリ/フレームワークを提供 (Ajaxを主目的においたライブラリ)
- XMLHttpRequestオブジェクト(ブラウザ依存)のラッピングやデータ交換の支援、テンプレートエンジン、更新時のエフェクト効果提供など
- Ex. Atlas Framework, Ajax.NET, AJASON, PEAR::HTML_Ajax, DWR, JKL.Hina+JKL.ParseXML, Prototype.jsなど
- Cf. Selenium, Log4JavaScriptなどテストツール、ロギング支援ライブラリも

■ 汎用フレームワーク型

- Ajax開発に対応した汎用フレームワーク(Ajaxは一機能にすぎない)
- Ex. Ruby On Rails, JSF(JavaServer Faces)

Ajax対応の主なフレームワーク & ライブラリ (1)

分類	名称	URL
	概要	
Java	AjaxTags	http://ajaxtags.sourceforge.net/
	JSP(JavaServer Pages)上で利用可能なAjax対応タグライブラリ。特別なフレームワークに依存せず、サーバサイドJava環境全般で利用できる	
	AjaxFaces	http://www.ajaxfaces.com/
	JSF(JavaServer Faces)上で利用可能なAjax対応UIコンポーネント。JSFとの親和性に優れる(正式版の利用は有償)。他にもAjax対応IDEで対応コンポーネントを提供	
	DWR(Direct Web Remoting)	http://getahead.ltd.uk/dwr/
	JavaScriptとJavaの間を繋ぐRMI(Remote Method Invocation)。特定フレームワークに依存しないため、既存システムへの導入にも有利	
PHP	PEAR::HTML_Ajax	http://pear.php.net/package/HTML_AJAX
	PHP上でAjaxアプリケーションを動作するためのシンプルなフレームワーク	
	AJASON	http://ajason.sourceforge.net/
	JSON形式でクライアント-サーバ間のデータ形式を交換	

Ajax対応の主なフレームワーク&ライブラリ (2)

分類	名称	URL
	概要	
ASP.NET	Atlas Framework	http://atlas.asp.net/
	Ajax機能をASP.NETに実装するためのJavaScriptライブラリとHTTPハンドラのセット	
	Ajax.NET	http://ajax.schwarz-interactive.de/
	ASP.NET上でAjaxによるサーバメソッド呼び出しを可能にするHTTPハンドラ	
Ruby	Ruby On Rails	http://www.rubyonrails.org/
	Ruby上で動作するMVCデザインのアプリケーションフレームワーク(設定要らずが特長)	
JavaScript	JKL.Hina+JKL.ParseXML	http://www.kawa.net/works/js/jkl/hina.html
	XMLHttpRequestによる通信とJavaScriptテンプレートエンジン機能を提供。クロスブラウザを吸収する、XML解析 / 出力のコードを簡素化し、保守性を向上	
	Prototype.js	http://prototype.conio.net/
	XMLHttpRequestのラッピングやJavaScriptによる開発全般にフレームワーク。YellowFadeのようなAjaxアプリケーションに有用な特定機能も	

Ajax対応の主なフレームワーク&ライブラリ (3)

■ フレームワーク&ライブラリの選び方

- Ajax開発上の問題を手軽に解決したい
JavaScript系のテンプレートやライブラリ
- 既存のサーバサイドアプリケーションにAjax機能を付加したい
フレームワークに依存しない汎用ライブラリ
(DWR, PEAR::HTML_Ajax, Atlas Framework)
- Ajax機能を使って実現したいことが想定されている
特定機能を提供するタグライブラリ(AjaxTags, AjaxFaces)
- Ajax機能を含むアプリを一から構築したい(フレームワークを自由に選択できる)
Ajax対応のフレームワーク(Ruby On Rails, JSF, ASP.NET)

参考) リッチクライアントの代表格 Flashとの比較

■ リッチクライアントといえば...

- Flashが代表的、ノウハウも多い(余談: Arax: Asynchronous RPC and XML)
参照) 「リッチ・クライアントから見るWebアプリケーション開発の将来」
<http://www.wings.msn.to/out.php?c=oa&id=375>
- AjaxとFlashとをどう使い分ければよいのか？

項目	Ajax	Flash
HTMLとの親和性	DHTMLによるサポート	比較的不得手
プラグインの要否	標準ブラウザで実行可能	プラグインの導入(普及しており、問題は少ない)
クロスブラウザ	ライブラリによってある程度吸収できるが、不得手	原則問題はない
オープン	XMLは標準技術、その他もオープンな技術である	Adobe社に依存する製品である
画像/動画	標準的なライブラリは備えない	強力に対応

- 画像処理や動画処理などの複雑なコンテンツはFlashで、HTML(入力フォームなど)との親和性を重要とする局面、テキスト中心の処理はAjaxで

Keyword-02) これからはあるものを使う時代 本格化する Web API の世界

■ 従来のWebアプリケーション開発といえば...

- サーバサイド技術(ASP.NETやJavaEE、PHP)やクライアントサイド技術によって一から構築
(長所) 自由度/柔軟性のあるアプリケーションを構築できる
(短所) サイト規模によっては膨大な開発工数と時間を要する
- ライブラリやアプリケーション(掲示板やCMS、Blogなど)のような既存のソフトウェアを導入して構築
(長所) 定型的な機能を一から作成する必要がない(工数の削減)
(多くの場合) レイアウトのカスタマイズが容易に可能である
(短所) 自サーバ上にインストールや環境構築が必要
データは基本的にサイト単位で管理

■ Web APIを利用することで...

- 他のサイトで提供しているサービスを利用したアプリケーションが構築できる
- では、既存のライブラリやアプリケーションとなりが違うのか？

What is Web API?

～従来のAPI / アプリとなにが違うのか?～

■ What is Web API?

- 従来のアプリケーションは人がアクセスするもの
Web APIはコンピュータ(アプリケーション)がアクセスするもの
- 従来のAPIはOSの機能にアクセスするためのもの
Web APIはWeb上のサービスにアクセスするためのもの
- 従来のライブラリが提供するものは、データを操作するための「道具」のみ
Web APIが提供するものは「道具 + (ネット上の膨大な)データ」



Web APIの分類 (1)

■ SOAP Webサービス型

- クラシカルなWebサービスの実現手段
- 下位プロトコルに依存しない拡張性(HTTP、FTP、SMTPなどで利用可)と高機能性
- .NET Frameworkの「.asmx」ファイル、Apache Axis、Soap関数(PHP5)などのライブラリを活用することでSOAPによるアクセスも手軽にできる環境が整いつつある
WSDL (Web Services Description Language) からクラスを自動生成
アプリケーションからはローカルAPIにアクセスする要領でサービスを利用
- × プロキシクラスを介する分、パフォーマンスは低い
- (一般的に)SOAP対応のライブラリが必要となる
- Ex. Google WebAPIs、Amazon Webサービスなど



SOAPリクエストの構造

Web APIの分類 (2)

■ REST(RePresentational State Transfer) Webサービス型

- SOAPの高機能性は必要ない、もっと手軽に！(マルチプロトコルは不要)
- クエリ情報経由でパラメータを送信、XMLで結果を取得(プロキシクラスは不要)
シンプルでパフォーマンスにも優れる
`http://api.search.yahoo.co.jp/WebSearchService/V1/webSearch?
appid=xxxxx&query=山田祥寛`
- 一般的なXMLパーサさえあれば、REST APIは利用できる(一般的にライブラリは不要)
- (厳密な議論はあるが...)SOAPを利用しないHTTPのみを使ってXML文書を送信するしくみ SOAPよりも人気があるアーキテクチャスタイル
- Ex. Amazon Webサービス、
Yahoo! Japan Webサービスなど

Yahoo Webサービスの応答メッセージ

```
<?xml version="1.0" encoding="UTF-8" ?>  
<ResultSet ... totalResultsReturned="10" firstResultPosition="1">  
<Result>  
<Title>サーバサイド技術の学び舎 - WINGS</Title>  
<Summary>...</Summary>  
<Url>http://www.wings.msn.to/</Url>  
<ClickUrl>http://wrs.search.yahoo.co.jp/...</ClickUrl>  
...中略...  
</ResultSet>
```

Web APIの分類 (3)

■ JavaScriptライブラリ型

- JavaScript上で利用可能なライブラリ
- クライアントサイド技術のみを理解していれば、すぐに使える手軽さが特徴
- Ex. Google Maps API, Google Homepage APIなど

■ RSS(Rich Site Summary)フィード型

- サイトの更新 / 新着情報を配信するための標準XML形式(特定ベンダに非依存)
- サイトが配信しているRSSフィードは、専用のRSSアグリゲータ(リーダ)、またはRSS対応ライブラリ、XMLパーサなどを利用することで自由に利用できる
- スパムメールの蔓延がRSSフィード普及を後押し
- 代表的なニュースサイトやブログツールなどもRSS配信に対応
- Windows Vista, Internet Explorer 7.0でRSSに標準対応の予定
- バージョンによる仕様差が大きい
Atom(コンテンツの配信や保存 / 編集に対応)を模索

参考) Web APIの挙動(比較)



Keyword-03) 高度な開発をより簡単に EoD (Ease of Development) の世界

■ What is EoD?

- 開発の容易性
 - もともとは「Javaの」開発をより簡単に、というスローガン
- 統合開発環境との統合などを目指した、単なる「簡易化」ではない
- より高度なアプリケーションをよりシンプルなコードで構築できる素地の提供
 - オブジェクト指向それ自体としての完成度を高めようという試み
 - ただし、実際にはツールやライブラリによる簡易化の文脈で語られることも多い
- 今までより簡単にコードが書ける、と同時に、コーディングの誤りを犯しにくくする

■ EoDはJavaの世界に留まらない

- Web 2.0の世界では、ソフトウェアは「永久にベータ版」
 - 新機能はリリースという形で、一括提供するものではない
 - 日常的に新機能を提供し、日常的にフィードバックを受け、日常的に改善を行う環境が必要
- Javaの世界に留まらず、すべての分野において「EoD」は必要とされている
- ASP.NET、PHPなどの環境における「EoD」にも注目！

サーバサイドJavaにおけるEoDの動き (1)

■ サーバサイドJavaの特徴

- デザインパターンなどの蓄積されたオブジェクト指向の開発 / セキュリティノウハウ
- 商業製品、オープンソース製品を問わず、ライブラリやフレームワークも多くの選択肢
Apache Jakarta Project (<http://jakarta.apache.org/>) などが有名
- 反面、コーディング手続きは煩雑、冗長な設定ファイルの山が問題

■ JavaSE 5/6におけるEoD

- JDK 5.0におけるEoD (Eclipse 3.1でJDK5に対応)
 - ✓ 拡張for文、オートボクシング、staticインポート(コーディングの容易化)
 - ✓ ジェネリックス(コーディングミスを減少)
- 見えてきたMustang (Java SE 6)
 - ✓ スクリプティング言語のサポート (JSR-223: Scripting for the Java Platform)
標準はJavaScript(ただし、言語は特定しない)
 - ✓ カスタムアノテーションの構築を容易化 (Pluggable Annotation Processing API)

サーバサイドJavaにおけるEoDの動き (2)

■ Java仮想マシン上で動作するスクリプティング技術

- スクリプト言語 = 動的型付け言語(実行時にデータ型を決定)のメリット
 - ✓ 型宣言に関わるコード記述を省力化
 - ✓ 継承関係に関わらず、同じメソッド名さえ持てば、ポリモーフィズムを利用できる
 - ✓ 実行時に動的にメンバの追加 / 削除が可能
 - 要は、設計時に想定しなかった部品の再利用が可能になる(変化を前提)
- 具体的なスクリプト技術
 - ✓ Groovy (<http://groovy.codehaus.org/>)
Javaの文法や標準ライブラリをそのまま利用できるのが特徴
 - ✓ JPython、JRubyなどPython、Rubyからの移植言語も
- もちろん、Javaを置き換えるものではない
ユニットテスト用のコード、ビルドコードを記述するなど補助的な局面に有用

■ ポストStruts、J2EEフレームワークの最新動向 (黒住氏セッション)

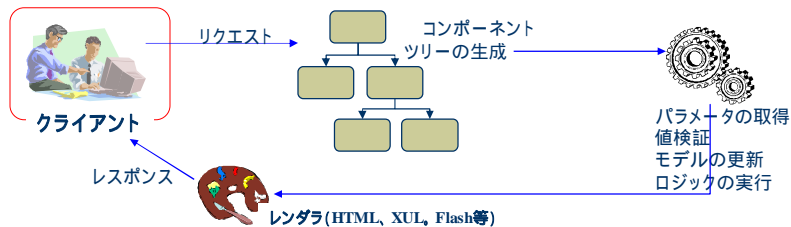
- JSF (JavaServer Faces)、Shale (ポストStruts)、Spring、Hibernate、EJB 3.0など

参考) JSF (JavaServer Faces)

<http://www.jcp.org/en/jsr/detail?id=127>

■ J2EE標準のアプリケーション・フレームワーク

- イベントドリブン型のアプリケーション開発 (VBライク)
- 画面の構築はUIコンポーネントで行う
直感的なユーザ・インターフェイスの構築が可能
- レンダラを切り替えることでHTML以外の出力も容易
- Oracle JDeveloper、Sun Java Studio Creatorなどの開発ツールも充実しつつある
- ただし、現時点で採用実績がまだ多くなく、Strutsとの住み分けもこれからの課題



ASP.NET 2.0におけるEoDの動き (1)

■ ますます充実したサーバコントロール

- コーディングレスで定型的なしくみを実現
ログインページなどは部品というよりページテンプレート

分類	概要
マスターページ/テーマ&スキン	サイト共通レイアウトやデザインの一元管理
データアクセスコントロール	データソース連携の一覧表や単票をコードレスで実現
ナビゲーションコントロール	ツリーメニューやパンくずリストを設定ファイルから生成 ()
セキュリティコントロール	ログインページやパスワード問い合わせ機能、ユーザ管理機能等
Webパーツコントロール	MyYahoo!のようなパーソナライズ機能 ()
リッチコントロール	Wizard機能やXML変換、カレンダー、マルチページなど

- ◆ ホーム
- ◆ ピアノ楽譜新刊情報
 - ◆ 改定新版 基礎エチュード
 - ◆ 独習ひまの
 - ◆ 10日で1曲マスター J-POPS編
 - ◆ Q&A掲示板
- ◆ ピアノ楽譜について
- ◆ 電子ピアノ
 - ◆ アップライトピアノ
 - ◆ グランドピアノ
 - ◆ 輸入ピアノ
- ◆ 鍵盤
- ◆ ペダル

Design

登録情報	ISBNコード	書名	出版社	価格	配本日
	4-7980-0945-8	今日かたのえるPOPシンガブル集 秀和システム	3150	2004/12/03	0:00:00
	4-7980-1270-X	Pocket辞書辞典 秀和システム	2520	2006/03/24	0:00:00

マイカランダ

登録情報	ISBNコード	書名	出版社	価格	配本日
	4-7980-0945-8	今日かたのえるPOPシンガブル集 秀和システム	3150	2004/12/03	0:00:00
	4-7980-1270-X	Pocket辞書辞典 秀和システム	2520	2006/03/24	0:00:00

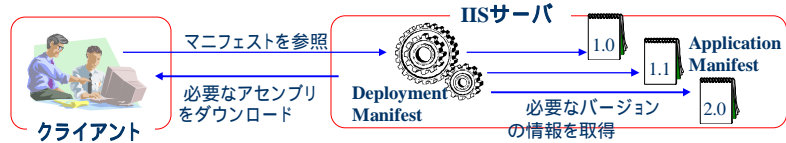
2006年

日	月	火	水	木	金	土	日
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	

参考) .NET Framework環境のリッチクライアント(1) ~ ClickOnce ~

■ Windowsアプリケーションを簡単に配置するしくみ

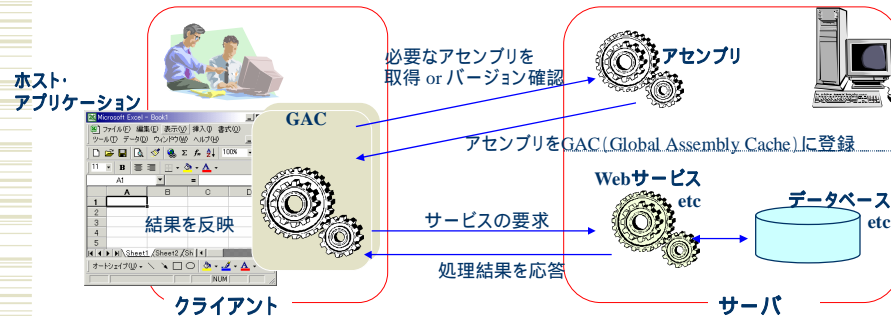
- Windowsアプリケーションのリッチクライアント対応
 - Windowsアプリケーションをサーバ上から動的に取得(常に最新版)
 - オフライン・サポート(2回目以降のアクセスはスタートメニューから可)
 - Windowsアプリを構築するのと同じ要領で構築可能、サーバ発行はVSで実施
 - アプリケーションの自動更新&必要に応じたロールバック
- Windowsアプリの使い勝手をそのままに、Webアプリの配布性を実現
- Windowsアプリ開発のノウハウをそのままに、リッチクライアントを構築できる
- セキュリティ面も.NET Frameworkが管理
 - SandBoxによるセキュアな実行(ファイルやネットワークアクセスの制限)
 - マニフェストへの署名(更新には発行者キーが必要)



参考) .NET Framework環境のリッチクライアント(2) ~ VSTO (Visual Studio Tools for Office) ~

■ Microsoft Officeをフロントエンドとしたリッチクライアント

- 常に最新のアプリケーションを動的に呼び出し可能
- VB.NET, C#などの.NET標準言語を開発に利用可能
Visual Studio .NETで一元的な開発が可能(CASも利用可能)
- コンピュータの苦手な人でもWord/Excelならば使えるという人は多い



PHP 5.0におけるEoDの動き

■ パーソナル用途からエンタープライズ用途まで幅広く対応

- 初学者にも分かりやすい平易な構文と直感的に使えるライブラリが特長
本来的なEoD。主にパーソナル用途に強み
- PHP 5ではオブジェクト指向構文、例外処理を大幅に強化
Javaライクなオブジェクト指向開発が可能に
- SOAP関数、SimpleXML、XMLReader/XMLWriterなどXML機能の強化
- JSR-223「Scripting for the Java Platform」
Javaアプリケーションにおけるフロントエンド開発言語の可能性
- 標準アプリケーションフレームワークか？ Zend Framework
Phrame、Mojavi、Agavi、Ethna、Mapleなどの乱立状態を解決？
- PHP 6の開発も開始
Unicodeネイティブ対応、コードキャッシュへの標準対応、
ユーザ入力チェックフィルタ(XSS、SQLインジェクション対応)、名前空間対応
- 大規模なシステム構築にも対応可能な基盤を強化
より幅広いアプリ開発が可能 (Javaや.NETとは違うものの、これもEoD)

まとめ

■ BuzzWord？それとも時代のKeyword？

- 新しい単語や技術が実際に定着するかどうかは、時間が経過しないと分からない
- 来年の今ごろはどうなっているか、誰にも分からない。
- 踊らされたくない。だから、BuzzWordには近づかない　それもひとつの方法
- でも、BuzzWordは時代の方向性の象徴であることも多い
- 消えてしまうBuzzWordでも、その本質を理解することは重要

BuzzWordと忌避せず、本質の把握に努めよう！

■ 今回、説明できなかったキーワード

- XMLデータベース
- SOA (Service Oriented Architecture)
- ロングテール
- アジャイル開発

ご静聴ありがとうございました

■「サーバサイド技術の学び舎 - WINGS」

- <http://www.wings.msn.to/>
- サーバサイド技術に関する記事や書籍、関連サイトなど最新の情報を日々紹介
- RSSフィードも配信中



サーバサイド技術の学び舎 - WINGS