

情報学科 CS コース
情報システム (3年後期)
第6回
(田島担当分第2回)

田島 敬史

2012 年 11 月 14 日

Web サービスの様々なアーキテクチャ

SOAP

- Simple Object Access Protocol の頭文字 (だった) .
- 分散オブジェクト計算での RPC (Remote Procedure Call) に相当.
- クライアント: サービス名や引数の情報を XML で記述したものをリクエストとして送信.
- サーバ: リクエストに対する計算結果を記述した XML を返す.
- 通信には, HTTP や SMTP を使う.

「サービス」を提供するインフラとしての Web

静的なページとリンクからなる「ハイパーテキスト」

- ユーザ操作 = 閲覧とリンクによるナビゲーション



動的なページとリンクからなる「動的なハイパーテキスト」

- ユーザ操作 = 閲覧とリンクによるナビゲーション
フォームへの入力



任意のアプリケーションを「サービス」として提供するための枠組み

- ユーザ操作 = 任意の高度な GUI
- 人間ではなくプログラムがアクセスする「Web API」

Web サービスの様々なアーキテクチャ

REST

- サーバ側に「オブジェクト」のような状態を考えない.
(ステートレス, stateless)
- リクエストに関する全ての情報を URI に埋め込む.
- 全てのリクエストは, HTTP の GET で表現.
- ステートレスという点では関数型プログラミング的

Web サービスの様々なアーキテクチャ

Web サービスにおけるセッション

- もっとも一般的な実装方法
 1. ユーザの新規アクセスやログインのたびにサーバがセッション ID を生成
 2. クライアントへHTMLを返す時に、ID を Cookie として HTTP ヘッダ中に入れて送信
 3. クライアントは Cookie を取り出し、以降のアクセス要求に、この Cookie を付加
 4. サーバ側では、セッション ID ごとにセッションの状態を記録
必要なら、適宜、「タイムアウト」
- アクセスしてくるユーザを識別し、インターネット通販などの「複数のステップからなる処理」や「ユーザ毎の表示」を実現

4

Web 登場初期の Web サイトの構成

Dexter model との対応で言うと1.5層構造？

1. サーバが HTML ファイルを蓄積
 2. クライアント側のブラウザが HTML を表示（レンダリング）し、インタラクションを管理
- レンダリングの詳細はクライアントに任せる考え方
 - － スクリーンサイズ、ウィンドウサイズ等の環境依存の表示
 - ただし、
 - － 当初は、たいしたインタラクションの機構はなかった
 - － HTML の拡張に伴い、表示の詳細までサーバ側で制御

6

Web サービスの様々なアーキテクチャ

Ajax

- Asynchronous JavaScript + XML
- 「ページ遷移」を行わない。
- ページに埋め込まれた JavaScript がサービスを提供しながら、裏で非同期でサーバと通信し、XML をやり取りしページの一部を更新。
- REST による地図と Ajax による地図の例
- 参考：Java applet, Flash

5

現在の Web サイト or Web サービスのサーバの構成

Dexter model との対応で言うと三層構造？

1. バックエンドのデータベース
(関係データベース, XML データベース?)
2. サーバ側のプログラム
 - クライアント(ブラウザ, ソフトウェア)からのアクセスに基づき起動
 - サーバ側の状態を更新 or 初期化
(「セッション」を持つタイプのサーバの場合)
 - データベースからデータを検索し XML や HTML を生成
 - 生成されたデータ(とスタイルシート)をクライアントへ送信
3. クライアント側でのレンダリングと、JavaScript 等によるインタラクション

7