

JSP/サーブレット型 Web アプリケーション専用の リムーバブル式開発環境構築支援システムの考案と実装

平河内 竜樹†

本論文では、従来煩雑であった JSP/サーブレット型の Web アプリケーションにおける開発環境の構築とコンテキストの形成を支援するシステムについて述べる。このシステムを利用すると、プログラムを一つ起動するだけで開発に必要な環境を整えることができる。また、設定ファイルを編集することなく作成した Web アプリケーションを実行することが可能になる。前者は仮想ドライブと環境変数の動的生成を利用することで、後者は入力情報をデータベースとして管理することで実現した。システムを初学者に適用した結果、開発環境の構築時間削減に大きな効果があることを確認できた。

1. はじめに

JSP/サーブレットの実行には、コンテキストが正確に構築されていること、作成したコンテキストの存在がコンテナへ通知されていること、コンテナが正しく起動していることが前提になる。具体的には JSP/サーブレットコンテナを用意し、仕様に沿ったディレクトリの構成、リソースの配置、設定ファイルの編集を経てコンテキストを構築し、形成したコンテキストをコンテナに登録しなければならない。図1にあるように、JSP/サーブレット型の Web アプリケーションは実行のために多くの作業が必要となる。

コンテキストの構築は Eclipse などの統合開発環境（以下、IDE と呼ぶ）でサポートされているが、現状では手動で行わなければならない作業が残される。またサーバのインストールや設定などの開発環境構築も、手順の説明されている Web ページや解説書を参照し、時間をかけて行う必要がある。これらの手順を自動化できるシステムを提案できれば生産性の向上に大きく寄与することができる。本研究では JSP/サーブレット型の Web アプリケーションにおける開発環境構築とコンテキスト構築の双方を支援するシステムを考案し、作製した¹⁾。

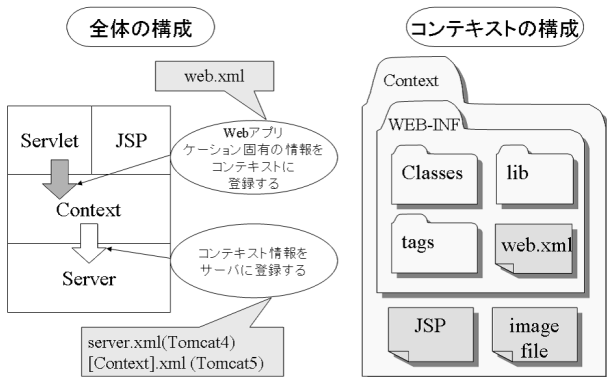


図1 JSP/サーブレット型 Web アプリケーションの構成

2. 本研究のアプローチ

2.1 開発環境構築の支援

開発に必要なソフトウェアを事前に選出・一元化し、基盤となる一つのプログラムの起動によって開発環境構築の全てを賅うことで、従来煩雑であったインストール・設定の手順を全て省略することができる。この視点から開発環境構築を支援するシステムを考案した。

2.2 コンテキスト構築の支援

従来の開発の流れと自動化が見込める部分を図2に示す！開発過程において自動化できる部分は全て自動化する」という視点でコンテキストの構築を支援するシステムを考案した。各工程について自動化できるかどうかを検討し、可能なものをシステムの機能として順次加えていった。

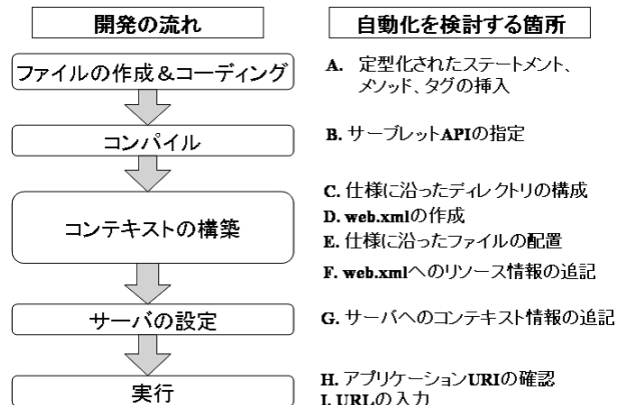


図2 従来の開発の流れ（図左）と自動化を検討する箇所（図右）

3. 課題とその解決法

3.1 パスや設定に依存しない開発環境の作成

OS には Windows2000/XP, JSP/サーブレットコンテナには Apache Tomcat (以下、単純に Tomcat と呼ぶ) を想定した。Tomcat をシステムに組み込むために解決しなければならない問題として、次の二点が挙げられる。

(1) パスの一貫性
組み込んだシステムから Tomcat を呼び出すためには、システムをどのディレクトリにインストールしても Tomcat を起動できるという条件を満たさなければならない。Apache Web-Server, MySQL, PHP を組み込んでいる XSAS²⁾ は仮想ドライブを利用し常に一貫したパスを得ることで各ソフトウェアの起動を可能にしている。この動きを利用して上記の問題を解決した。

(2) 環境変数の動的生成
Tomcat は起動時に JAVA_HOME (Java の位置を示す環境変数) を参照するため、従来はこれを手動で OS に設定する必要があった。本研究の過程で行った分析の結果、環境変数のスコープは設定したプログラムから CALL したプロセス内でも有効であることを確認した。これを利用して、JAVA_HOME を設定した後 Tomcat の起動プログラムを CALL するバッチプログラムを作成することで上記の問題を解決した。

† 神奈川大学理学部情報科学科 (2005 年卒業)

本文中では Java クラスファイル (サーブレット) や JSP ファイルなどコンテキストの構成要素を指す。

本研究では Tomcat を www サーバ+JSP/サーブレットコンテナとして使用した場合を想定している。図中のサーバはコンテナの役割も兼ねる。

XOOPS Stand Alone Server の略。XOOPS は CMS の一種で動作に文中で述べた 3 種のソフトウェアを必要とする。

3.2 入力した情報の保存・変更・削除

実行に必要な設定ファイルやディレクトリの追加・修正は、JSP/サーブレット作成時にシステムを通じて入力された情報を元に行うことができる。また、各種設定ファイルの走査やオプション付きのコンパイルも Tomcat のホームディレクトリ (CATALINA_HOME) さえ保持しておけば対応できる。しかし、作成時に入力された情報を再利用する機能、例えば URL の作成などを実現するには、多量のデータを整理した状態で保存する必要がある。また、より充実した機能を付加するためには、登録したサーバや作成したファイルの情報を変更・削除できることが求められる。

この問題は専用の簡易 DB を用意することで解決した。作成時に入力された情報をサーバ、ホスト、コンテキスト、リソースの 4 階層に準えて各テーブルに蓄積し、必要に応じて参照する。例えば、URL の作成においては、サーバ DB からポート番号を、ホスト DB からホスト名を、コンテキスト DB からコンテキストパスを、リソース DB からマッピングパス (サーブレットの場合) 及びリソース名 (JSP の場合) をそれぞれ取得し、URL を形成してブラウザに渡すという処理を行う。

4. システムの仕様

実装は Borland 社より無償で提供されている IDE, Delphi6Personal を利用して行った。開発環境の構築を支援するシステムを Java Stand Alone Server, コンテキストの構築を支援するシステムを Java Web Application Manager と呼ぶ。主な機能は以下の通りである。

4.1 Java Stand Alone Server (JSAS)

Tomcat の起動

システムをインストールしたディレクトリや OS に設定された環境変数にとらわれず、起動と同時に Tomcat を呼び出すことができる。関連するオプションとして、ログの出力先を定める start/run 引数の選択、起動確認に用いる [ループバックアドレス: ポート番号] へのワンクリックアクセスなどの機能を付与した。

JSAS の基本動作は図 3 の通りである。図中の「設定ファイル」は JSAS が生成・参照するもので、仮想ドライブ生成に必要な情報も記入されている。これが無い場合は、同ファイルを処理 C で生成した後一度終了し、再起動する。

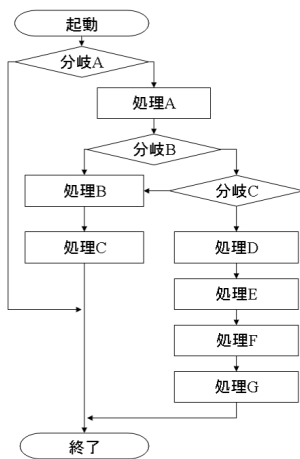


図 3 JSAS の基本動作

- 分岐A: 二重に起動していないか
 - ・している→終了へ
 - ・していない→処理Aへ
- 分岐B: 設定ファイルが存在するか
 - ・存在する→分岐Cへ
 - ・存在しない→処理Bへ
- 分岐C: 設定ファイルに書かれているディレクトリが妥当且つ存在するか
 - ・存在する→処理Dへ
 - ・存在しない→処理Bへ
- 処理A: 仮想ドライブを生成する
- 処理B: CATALINA_HOME, JAVA_HOME の入力をもとに起動する
- 処理C: 入力されたパスを設定ファイルに保存する
- 処理D: パスを引数として渡しバッチプログラムを起動する
- 処理E: Tomcat を起動する
- 処理F: (開発者が) 開発を行う
- 処理G: Tomcat を停止する

4.2 Java Web Application Manager (JWAM)

コンテキストの作成、リソースの作成の順を経て実行 IDE を用いない従来の開発では JSP/サーブレットを作成した後、各種設定を行い実行環境を構築していた。JWAM では逆に、最初に Web アプリケーションの外観を形成し、その後構築要素であるリソースの作成を行う、という形式を採用した。具体的な手順を以下に述べる。

- (1) 使用するサーバ情報などの登録
- (2) コンテキストの作成とコンテキスト情報の登録
(自動化: 図 2-C,D,G)
- (3) リソースの作成とリソース情報の登録
(自動化: 図 2-A,B,E,F)
- (4) 実行
(自動化: 図 2-H,I)

示した手順では、コンテキストの作成とリソースの作成それぞれの段階で、作成と同時に登録作業を行う。この手順を踏めば、リソースの作成が終わった時点で必要な登録作業は完了し、実行可能な状態になる。実行に必要なコンパイルや URL の入力においても手動による文字入力の手間を省略している。また DB 参照時にデータを発見できなかったときの例外処理として、現在の登録情報を基に各処理を行う。これより JWAM を用いずに作成した JSP/サーブレットも、JWAM を通じて実行することができる。

同じくコンテキストの構築をサポートしている Eclipse (Ver3.01 + Sysdeo Eclipse Tomcat Launcher) や、実行環境としての Apache Ant と比較すると、以下の点で優れている。

- サーブレットの作成時に web.xml へのマッピング情報の追記を自動で行うことができる
- 生成されるファイルに JSP/サーブレットの定型的なテンプレートを埋め込むことができる
- ハードウェアリソースの消費が少なく動作が軽い
- 環境変数の設定を行う必要がない
- ビルドファイルを作成する必要がない

さらに、コンテキストの作成と同時に Eclipse のプロジェクトファイルを生成する機能を持っている。これより JWAM で作成したコンテキスト・リソースは、Eclipse に引き継ぐこともできる。

5. システムの検証

本システムを用い、インストールからブラウザに「Hello」の一文を出力するだけのテストプログラム (サーブレット) の作成・実行までにかかった時間を測定した。IDE などを用いず教科書とプリントだけで行ったものと比較した結果、表 1 のようになった。二つの実験は被験者群の違いなどにより厳密には比較できないが、今回作成したシステムが大きな効果を出していることは示された。これは実験後のアンケートで聞いた箇所として挙げられた、環境変数の設定や各種設定ファイルの編集、リソースの配置などの作業から解放されたためと推測できる。

	サンプル数	平均時間 (分)	標準偏差 (分)
従来の方法	15	250	101
本研究の方法	6	10	4

表 1 検証結果

6. まとめ

本研究で提案した「必要なソフトウェアをシステムに組み込み一元化する」という手法は、JSP/サーブレット型 Web アプリケーション開発環境構築の労力削減において大きな効果があることを確認できた。また、従来の IDE でサポートされていたコンテキストの構築においても、更なる自動化の余地とその改善方法を示すことができた。

本システムはその実現手法 (環境変数の動的生成) により可搬性を持たせられたことも特徴として挙げられる。USB メモリなどのリムーバブルメディアにインストールすれば「開発環境」を持ち運べ、Windows マシンさえあればどこでもすぐに開発を始めることができる。

7. 今後の課題

今後の課題として、以下の内容が挙げられる。

- セキュリティを考慮した仕様の変更
 - 起動時のユーザ認証
 - DB の暗号化
- 開発に求められる機能の追加
 - リソース作成時に提供するタグ等の追加

謝 辞

本研究を進めるにあたって多数のご助言を頂いた、神奈川大学理学部桑原恒夫教授に心より感謝致します。

参 考 文 献

- 1) [本論文初出] 平河内竜樹: JSP/サーブレット型 Web アプリケーション専用のリムーバブル式開発環境構築支援システムの考案と実装, 平成 16 年度神奈川大学理学部情報科学科 卒業研究発表会要旨集, pp.261-262 (2005)