

Java フレームワークによる古文書データベース・システムの開発

A Development of Ancient Document Database System by Java Framework

高井正三 † 喜多啓太 ‡ 米田恭章 ‡

Shoso Takai † Keita Kita ‡ Yasuaki Yoneda ‡

† 富山大学総合情報基盤センター ‡ 富山大学工学部知能情報工学科

† Information Technology Center, University of Toyama

‡ Department of Intellectual Information System Engineering, University of Toyama

〒930-8555 富山市五福 3190

3190 Gofuku, Toyama-shi, Toyama 930-8555 Japan

† takai@itc.u-toyama.ac.jp ‡ s0370224@ems.u-toyama.ac.jp ‡ s0370281@ems.u-toyama.ac.jp

概要

筆者らは 2005 年度にそれまで提供してきた日本現存朝鮮古書を対象とする古文書の書誌情報データベース DOKB (Database of Old Korean Books existed in Japan) を全面的に再編成し, Unicode ベースの Java データベース・システムを開発, 再構築を実施してきた. 今回の古文書データベース・システムを開発するに際し, Java フレームワークの一つである JSF (Java Server Faces) を使用した. 一方, 企業での Web アプリケーション開発には, Java による Application Server 開発の業界標準である J2EE Platform が使用され, Struts 等の Java フレームワークによるシステム開発が進められている. 本稿では, Java フレームワークについて, その歴史と現状を概観しながら, 筆者らの古文書データベース・システム開発の経緯を通じて, Java による WebDB アプリケーション開発の推進, 標準化の方法, 問題点などを解明し, Java フレームワークの生産性の高さを提案する.

キーワード

Java フレームワーク JSF Struts Light Weight Java WebDB アプリケーション開発 古文書データベース

1. はじめに

筆者らは富山大学人文学部教授の藤本幸夫氏が 1910 年以前の朝鮮古書を対象に全国から収集してきた「日本現存朝鮮古書」に関する 28 項目にわたる書誌情報をデータベース化し, インターネット上に公開してきた. この間 3 回に渡って, 旧字体漢字や外字の入力, 蓄積, 検索, 表示および印刷方法を改善するために, データベース・システムを改訂してきた. 2005 年には使用している文字コードを全面的に Unicode 化するため, Unicode をサポートするデータベース管理システムへの切り替えと, システム開発言語の Java 化を進め, 第 4 世代目の WebDB システムを開発, 再構築してきた[1].

今回この古文書データベース・システムを開発するに際し, Java フレームワークの一つである JSF を使用した. JSF は次世代 Java Web アプリケーション開発の標準になりつつあるという情報を得て, JSF の実装には Apache MyFaces を, O/R マッピング・フ

レームワークには Hibernate を, DI/IoC コンテナには Spring Framework を使用した. また, このシステムの開発環境には Eclipse3.1 を使用した.

一方, 企業では Java による Web アプリケーションの開発に, J2EE (Java 2 Enterprise Edition) が使用され, Java フレームワーク Struts が広く使用されている. 筆者が所属しているメーカー系の IT 研究会では Struts を勉強しながら, 今までの WebDB システムを Struts で再開発してきた[2].

本稿では, WebDB アプリケーション開発の標準となりつつある Java フレームワークについて, その歴史と現状を概観しながら, 今なぜ Java なのか. Java フレームワークとは何か. なぜ JSF や Struts を使うのか. Java フレームワークを使うためには何が必要で, どのような知識や技術が必要か, 等について, 筆者らの古文書データベース・システム開発の経緯を通じて, Java による Web 情報システム開発の推進, 標準化の方法と問題点を明らかにし, Java フレームワークの生産性の高さを提案する.

2. Java フレームワークとは

2.1 サーバ・サイド Web アプリケーション開発

サーバ・サイドの WebDB アプリケーションの開発にはこれまで、CGI (Common Gateway Interface) /Perl, PHP, ASP.NET, JSP/Servlet, そして J2EE 等のサーバ・サイド・テクノロジーが使用されてきた。プログラミング言語には Visual Basic, C/C++, C#, Java, Visual J#などが、Script 言語には VBScript, Perl, Ruby, PHP, JavaScript 等が使用されてきた。最も Web Browser の機能追加とともに、様々な開発言語と API (Application Program Interface), Application Software の開発環境が整備されてきたので、今日のような WebDB アプリケーションが普及してきたとも言える。この論文誌 JACN でも ASP[3][4]や PHP[5], Java Servlet[6]などのシステム開発が報告されている。

2.2 なぜ Java なのか

Java については、当初 Netscape Navigator 上に Java 仮想マシン JVM が実装され、Java Applet で初めて動きのある Web コンテンツが公開されたのが衝撃的であった。その後は同じような言葉の JavaScript が一時盛んに使用されたが、やはり廃れ、CGI/Perl や PHP によるサーバ・サイド Web(DB) Application が作成された。この間、Java は進化を遂げ、1999 年 6 月に J2SE (Java 2 Standard Edition) など 3 つの Platform が発表され、その後 EJB (Enterprise Java Beans), Java Servlet, JSP (Java Server Pages), JDBC (Java DataBase Connectivity) 等が整備されてきて、システム開発の生産性を向上させたオブジェクト指向プログラミング言語 Java が、Application 開発において標準で使われるようになった。

Java の最も大きな利点は、Platform に依存しない実行環境を提供する実行ライブラリであり、一度作成した Java コードは Windows, Solaris などの UNIX, Linux, MacOS 等、JVM のある OS ならどれでも使用可能である。“Write Once, Run Anywhere” の言葉の通り、「一度書いたら何処でも動く」ようになっている。また、Java はオブジェクト指向プログラミング言語 (OOL: Object Oriented Language) として、言語仕様が整備されている。

筆者らが Java を使用する根拠は Unicode が標準でサポートされていることによる。これまでは PHP で WebDB システムを作成していたが日本語データは EUC コードを使用していた。旧字体漢字や外字の多い古文書データベースのアプリケーションには、最近 Extension C までの CJK コードがサポートされている Unicode が最も適しており、PHP では Ver.6 以降のサポートとなっているため、また、企業の Web

情報システムの大部分が Java で開発されているという状況から、この Java を選択する方がベターだと判断せざるを得なかったのが本音である。

2.3 Java フレームワークとは何か

Web Application 開発における一般的な Framework (骨組み) の定義は、Application の構造 (=骨組み) まで定義した形で実装された、再利用可能な Run-time Components (実行時の構成部品) である。Application 非依存の Run-time Components を指す場合が多く、Framework が Application の Components を呼び出す形式が一般的である。即ち、Framework は共通の目的を持つ Application を対象に、共通する機能や構造をもった「半製品」を提供し、Framework から呼び出される Application の「固有処理」を Add-in することによって開発を進める。このような動作の方式は「制御の逆転 (Inversion of Control)」とか「ハリウッドの原則: 連絡はこちらからするから、勝手に連絡してこないで、The Hollywood principle: Don't call us, we call you.」と呼ばれる所以である。

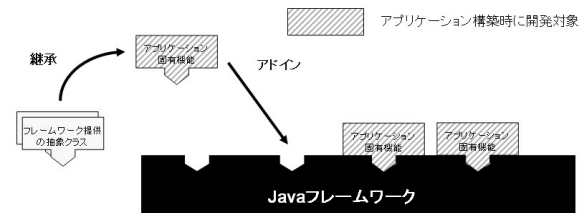


図1 フレームワークへの固有機能のアドイン

図1に示すように、一般的な開発では、開発者は Framework の Documents や Templates (雛形) に従って、固有部分の実装を記述することになる。Java をベースとした Framework の場合には、基底クラスが提供されていて、それを拡張して固有機能を実装することが多い。Framework には汎用的な Utility 機能や、これに開発補助 Tool を加えた Package で提供されるものもある。

Java Framework には一番メジャーな MVC (Model, View, Controller) アーキテクチャーの Framework である Struts と、今後中心となると言われている JSF がある。

なお、余談ながら Java 以外のフレームワークでは、Microsoft が進める .NET Framework があり、一大勢力を有している。

2.4 ソフトウェアの再利用と MVC モデル

これまでソフトウェア開発には生産性、メンテナンスの容易性、信頼性・品質の確保などが求められてきたが、特定の Technology を使用したからといって、これらの要求を実現できないのが本音である。

しかし、これを実現するための一つの方法にソフトウェアの再利用を高めるためのアプローチがある。今まで「再利用性の高い処理部品」としてライブラリが幅広く用いられてきたが、メイン・プログラムを作成する必要性があり、保守性や品質の確保が大変で、開発コスト、維持管理コストが高かった。

このような中で、オブジェクト指向プログラミング向けの Application Architecture として MVC モデルが登場し、頻繁に発生する User Interface の変更/追加要求に対し、機能の追加に対応する拡張の確保、動作の変更にたいするメンテナンス性の確保、そしてソフトウェアの品質を確保するために役立っている。MVC モデルは図 2 に示すように、Application を Model (Application のデータとロジックをカプセル化)、View (Model の可視化)、Controller (Model と View の制御) の 3 つの機能に分類して構成している。特に Model を Bean で、Controller を Servlet で、View を JSP で実装するものは「MVC モデル 2」と呼ばれている。例えば、Application に新たな画面や機能を追加する場合は、主に Controller の修正を行い、他の画面や機能の修正を最小限に抑えることができる。

また、構成する Components の役割が明確に分離できるので、図 3 に示すように複数人で分担し、並行に Components の効率的な開発が可能になった。

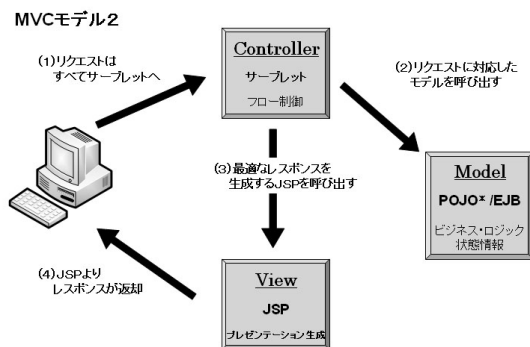


図 2 MVC モデル (MVC モデル 2)

Model	View	Controller
POJO/EJB	JSP	Servlet
Java Programmer	Web Designer	Java Programmer Site Architect

図 3 MVC モデルの開発分担

2.5 なぜ JSF や Struts を使うのか

JSF は MVC モデルの Framework で、2003 年 3 月に Java の標準化団体 JCP (Java Community Process) によって標準化され、J2EE 標準 Framework となっている。更に、Web User Interface の Components 化、ページ間遷移の Graphical なツールによるサポート

を行っており、EoD (Ease of Development) の実現によって、高い開発生産性とリッチな UI を実現している。従って、JSF は J2EE における標準サーバ・サイド UI Components Framework といえることができる。

一方、Struts は Craig R. McClanahan によって開発された Open Source の Web Application Framework で、Apache Jakarta プロジェクトに寄贈され、現在は Apache Struts として開発されており、Version 1.2.9 まで公開されている。現在、McClanahan は「Struts と JSF を直接統合してしまおう」という考えの下に、Shale という Framework を策定中である[7][8]。

Struts の特徴は、シンプルな機能を提供する軽量の Framework で、クライアントには JSP で生成した HTML (+ JavaScript) を使用し、拡張性を確保している。Struts は MVC Architecture を採用し Controller と View の一部を提供している。Controller は、Action Servlet, Request Processor, Action (開発対象)、例外ハンドラー (開発対象)、Struts 構成ファイル (開発対象: Struts-config.xml)、デプロイメント・ディスクリプタ (開発対象: web.xml) など構成され、View は、ActionForm (開発対象)、JSP (開発対象)、Struts タグ・ライブラリ、Validator、Validator 設定ファイル (開発対象) で構成されている。Model のビジネス・ロジックは使用者の開発対象となっている。

2.6 Java Framework を使うためには何が必要か

(1) オブジェクト指向

まずはオブジェクト指向のできる脳(通称オブ脳)を作ることが必要不可欠である。Web システムの開発生産性を向上させるための OOL は以下の主要な機能を有している。これを理解することが最初である[9]。

- 1) クラス = データ (変数名) + メソッド (機能)
- 2) 継承 ==> (extends, for 差分開発)
- 3) 隠蔽 ==> (Private, Protected, Public)
- 4) カプセル化
- 5) ポリモーフィズム

(2) オブジェクト指向の原則の理解

- 1) 単一責任の原則
- 2) オープン・クローズドの原則
- 3) リスコフの置換原則、他

(3) Framework の特性に合わせた設計

Struts や JSF を十分使いこなすには、以下のような理解する項目が多々ある。

- 1) まずは書き方を覚える
- 2) どのような仕組みで機能が実現されているか
- 3) Framework の特性に応じて行わなければならない処理にはどのようなものがあるか
- 4) 特性に合わせた設計とはどのようなものか

2.7 どのような知識や技術が必要か

では, Java Framework を使った WebDB アプリケーション・システムを開発するにはどのような知識や技術が必要か, 以下に洗い出してみた.

(1) プログラム言語/スクリプト言語/HTML

Java, JSP/Servlet, Java Applet, HTML, JavaScript, CSS, HTTP

(2) Web アプリケーション開発と Framework

JSF, Myfaces, Struts, EJB(Enterprise Java Beans), J2EE, J2SE, JDBC, Ajax

(3) データベース

RDBMS(DB2, UDB, MySQL, PostgreSQL), SQL, XML

(4) 統合開発環境の使い方

最低でも Eclipse 等の統合開発環境が必要

(5) オブジェクト指向開発分析・設計手法

牛尾剛著「オブジェクト脳の作り方」を借りれば, 以下のような分析・設計手法が不可欠

- 1) 画面遷移イメージの作成
- 2) ユースケース図(要求分析図)の作成
- 3) シナリオの作成
- 4) クラスの抽出
- 5) オブジェクト図の作成
- 6) クラスを関連づけた分析クラス図の作成
- 7) アーキテクチャーの導入
 - (画面/ロジック/データに分析して設計)
- 8) 設計レベル・クラス図の作成
- 9) コードの作成(自分でコードを書くこと)
- 10) フィードバック

(6) その他

- 1) デザイン・パターンの習得[9]
- 2) Java コードの書き方の習得[10]

3. Java Framework による古文書 DB 開発

3.1 Light Weight Java による WebDB 開発

筆者らは Java によるシステム開発を決定し, とりあえず Java に関する書籍や雑誌を大量に収集し, その中から Light Weight Java[11]を選定した.

JSF の実装に, Apache Incubator Project で開発されている Apache MyFaces を, O/R (Object-Relational) Mapping Framework には, 高い Performance と豊富な Tools のサポートによる, 使いやすさに定評にある Hibernate を, DI(Dependency Injection)/IoC(Inversion of Control) コンテナについては, 現在, 世界で最も普及している Spring Framework を使用した.

また, 実行環境として Web サーバに Apache を, Web コンテナには Tomcat を, RDBMS には Unicode をサポートしている Freeware の PostgreSQL 使用し

た. 開発環境は Eclipse3.1 で, サーバ・システムには Windows 2003 Server を使用した(図4, 5).

Software Configuration of Web Database System named DOKB

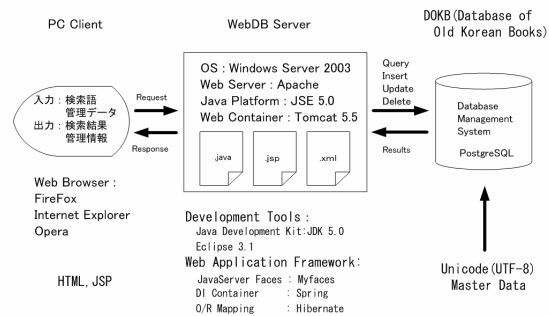


図4 Software 構成図

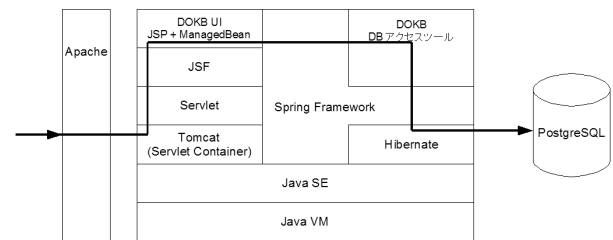


図5 DOKB の構成図

3.2 Layers パターン多層アーキテクチャー

更に, WebDB システム開発には, ベスト・プラクティスの Layers パターンによる多層アーキテクチャーを採用している(図6). 即ち, プレゼンテーション層は JSF(Myfaces)をベースに Managed Bean, JSP を使用して, クライアント(Web Browser)からのリクエストを処理し, ビジネス・ロジック層と通信し, その結果に基づいて画面表示の制御を行う. ビジネス・ロジック層は Spring のサービス・コンポーネントを使用して業務ロジックを実行する. インテグレーション層(データ・アクセス層)は O/R マッピング Hibernate の DAO (DataAccessObject) パターンを使用して RDB へのアクセスを行い, この全体を Web コンテナ Jakarta Tomcat 上で稼働させている. 最後の EIS(Enterprise Information System)層では RDBMS の PostgreSQL を実行している.

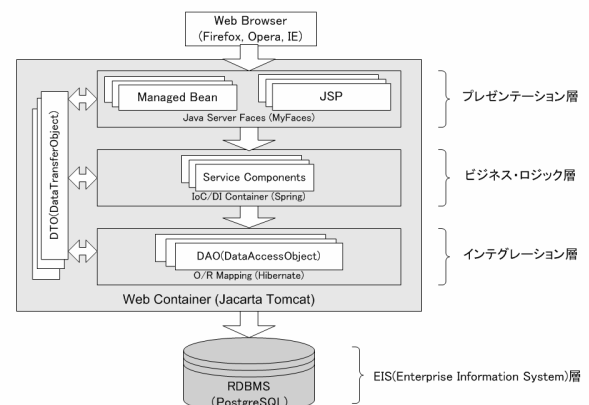


図6 Layers パターンによる多層アーキテクチャー

3.3 古文書データベースの画面遷移と画面

筆者らが開発した日本現存朝鮮古書データベース検索システムのトップ画面は図7の通りである。

図8は画面の遷移設計図、図9は検索語入力画面、図10は検索結果の表示画面、図11は文書単位の詳細表示画面である。



図7 日本現存朝鮮古書データベース検索のトップ画面

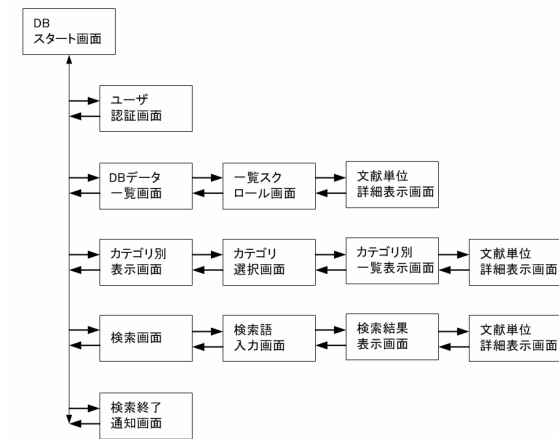


図8 検索画面遷移設計図



図9 検索語入力画面



図10 検索結果の表示画面

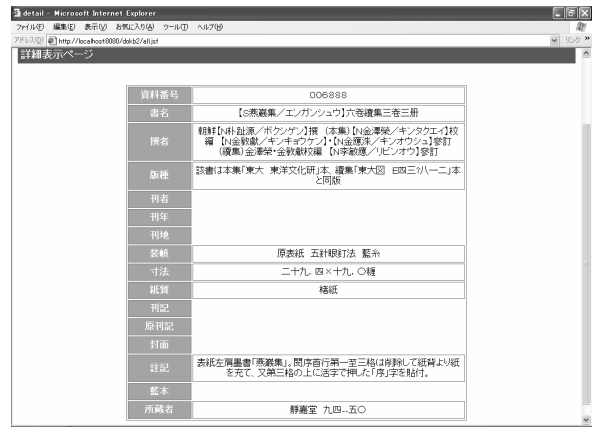


図11 文書単位の詳細表示画面

3.4 S-JIS 欠字の Unicode 化

現在まで使用してきた S-JIS 欠字は 1,079 文字であったが、Unicode の基本多言語面 BMP と第 2 面補助的表意文字面の Extension B まで調査したところ、Unicode で定義されていない文字は欠字の 3% 程度であった。S-JIS コードからの Unicode への変換は正規表現を使って 3% 以外を変換して、Unicode をサポートしている PostgreSQL の RDBMS へ再編成した。

4. 開発上の問題点

JSF Framework での開発の問題点は、マッピング用の xml 設定ファイルを記述するのが大変であることである。

1) WEB-INF\web.xml

FacesServlet を登録し、JSF Application へのリクエストを受け付けるように URL と Servlet のマッピングを行う。

2) WEB-INF\faces-config.xml

JSF Application のモデルである Managed Bean やクライアントからの入力値を検証する Validator、入力値の Format や型を変換する Converter、画面遷移のルールなどを設定する。

5. Java フレームワークの生産性

5.1 Java 統合開発環境

JSF や Struts Framework を使用して効率的に WebDB アプリケーションを開発する Java 統合開発環境には、以下のようなものがある。

- 1) Sun Java Studio Creator 2
- 2) IBM Rational Software Development Platform
- 3) Eclipse 3.1

1, 2) を使えば XXX-config.xml ファイルは殆ど自動的に作成してくれるようになっている。3) の Eclipse ではそこまでサポートしていないので、自分

でドキュメントを見ながら記述するより方法はないようである。

5.2 生産性

JSF や Struts を使った WebDB アプリケーションの開発における生産性は、素人ではこの開発技法を習得するのに多大の時間がかかるが、誰か 1 人以上のオブジェクト指向プログラミングのプロフェッショナルがいれば、開発プロジェクト・チームの相当の生産性が上がると言われている。参考文献[2]のような Web サイトの News&Topics システムは、Struts を知っているプロフェッショナルなら、一週間くらい集中すれば開発が可能で、利用者画面と投稿管理画面のシステムは一応動くようになると言われている。従って、プロフェッショナルにとっては非常に高い生産性を産み出すことが可能である。

6 . 今後の課題

WebDB アプリケーション開発の生産性向上と維持管理コストの軽減を目標に、Java Framework が益々便利になるようとしている。複数人で共同開発するための Version 管理ツールには CVS (Concurrent Versions System) や SVN(Subversion) が使用され、EoD (開発の容易さ) を目指した Framework が整備されつつある。生産性ばかりでなくセキュリティ上の堅牢性も含めて、今後は UI に画面遷移を伴わない Ajax (Asynchronous JavaScript+XML) 技術が駆使された Web 情報システムが登場すると予想される。また、現在の書誌情報データベースに古文書の写真画像、刻手の印影画像を追加する計画も進んでおり、この画像検索と検索語類推入力に、Google Suggest や Google Maps/Local/Earth に使用されている Ajax 技術を導入していくことも計画中であり、より高度で進歩的な User Interface を持つ WebDB システムとして登場させることも十分期待できる。

一方では PHP が今後の主流になると予言する人もいるが、Java は今年 Java EE 5 が正式に公開され、EJB3.0 や JSF の仕様が盛り込まれ、EoD の実現という方針の下に、開発作業を容易にするための工夫がなされている。更に JSE、JME が整備されれば、Java による Web アプリケーション開発は一層進むものと思われる。

また、古文書データベースの文字コードにおいても、Unicode の Extension C の制定が進んでおり、これによって殆どの古文書にある漢字は表現できるようになると期待されている[12]。

筆者らのシステムは、今後、管理者モードの入力管理画面を追加整備して完成させる予定である。

謝辞

現在、筆者らがサービスしている PHP 版日本現存朝鮮古書データベースの構築に当たって支援を受けた同僚の布村紀男助教授、DOKB データベースの作成に当たって、書誌情報、データの分類方法、記法、索引の採取方法など、本データベースに全体に関する情報の提供と相談を受けた人文学部の藤本教授。これらのデータ入力と修正に献身的な努力をしてくれた越野、洲崎、葉山、木戸、竹澤の女性スタッフ。ここに記して感謝の意を表する。

参考文献

- [1]高井正三, 日本現存朝鮮古書データベースの国際対応化の方法, 富山大学総合情報基盤センター広報, Vol.3, 29-38, 2006
- [2]第 17 回 IBM ユーザー研究会連合会ホームページ <http://www.uken.or.jp/isuc/isuc17/index.shtml>
- [3]升屋正人, 鍵山茂徳, Web 履修受付システムの開発, 学術情報処理研究 JACN, No.5, 31-42, 2001
- [4]奥山澄雄, 伊藤智博, 立花和宏, 仁科辰夫, 鈴木勝人, 青木和恵, データベースを利用した教育・研究情報の収集・公開の試み, 学術情報処理研究 JACN, No.9, 109-112, 2005
- [5]江藤博文, 吉賀夏子, 只木進一, PHP による論文投稿システムの開発, 学術情報処理研究 JACN, No.9, 113-117, 2005
- [6]野村武志, 當房新一, 渡辺義明, 渡辺健次, 江藤博文, 只木進一, Java Servlet を用いたネットワーク利用者認証システムの開発, 学術情報処理研究 JACN, No.9, 85-89, 2005
- [7]黒住幸光, 芦澤嘉典, 最強フレームワーク Apache Struts 1.2 パーフェクトガイド, ISBN4-7741-2234-3, 技術評論社, 2005
- [8]阿部敬一, J2EE 及び Web アプリケーション・フレームワーク概説, 北陸 IBM ユーザー研究会特別セミナー資料, 2005.10.6
- [9]牛尾剛, オブジェクト脳のつくり方, ISBN4-7981-0418-3, 翔泳社, 2003
- [10]杉山貴章, 正しく Java のコードを書いていますか? - みんなで守ろう Java コーディングお作法 -, JAVA PRESS, Vol.47, 40-53, 2006.04.10
- [11]岡本隆史, 吉田英嗣, 金子崇之, 権藤夏男, Light Weight Java, ISBN4-8399-1777-9, 毎日コミュニケーションズ (MYCOM), 2005
- [12]師茂樹, 秋山陽一郎他, 多言語情報処理, 漢字文献情報処理研究, 漢字文献情報処理研究会編集, 第 6 号, 157-173, ISBN4-87220-098-5, 好文出版, 2005