

図書館における Web アプリケーション開発と人材育成について

兵藤 健志[†] 井上 創造[‡] 牧瀬 ゆかり[§]

<抄録>

Web2.0 の時代に応じた図書館システムは、利用者の要求や、連携した他のシステムの進化に応じてシステムを柔軟かつ迅速に適応させる必要がある。しかし一方では、そのようなシステムの適応を実現できる知識を持った人材が不足しており、その育成についての具体的なカリキュラム開発もなされていない。本論文では、上記のような人材を育成する方法論を確立するという目的のもとで実施した Web アプリケーション勉強会についてその結果と課題を述べ、システムライブラリアン育成のカリキュラムへの一助となることを図る。

<キーワード> 図書館システム, Web アプリケーション, Ruby on Rails, 人材育成, システムライブラリアン

Web Application and Staff Development in Libraries

HYODO Kenshi INOUE Sozo MAKISE Yukari

1. はじめに

近年の目覚ましい情報システムの普及に伴い、World Wide Web を用いたアプリケーション（以下 **Web アプリケーション**と呼ぶ）が図書館において主要な位置を占めてくるようになってきた。Web アプリケーションの世界においては現在、Web2.0 というキーワードに代表される大きな変化が起こっており、図書館で用いられる情報システム（以下、**図書館システム**）に対する影響も少なくない。

Web2.0 の主な特徴としては、

1. Wikipedia^[1]や Mixi^[2]のように利用者が自らコンテンツを作り上げていく利用者参加型である、
2. Amazon^[3]や Flickr^[4]のように、情報システム間で連携して Web アプリケーションを作り上げる、

ということがあげられる。1 に関しては利用者のユーザビリティ、つまり利用の容易さが重要になる。そのため、完成版の一步手前である βバージョンという版を与えたまま運用を開始し、利用者の声をもとにシステムを改変していく必要が出てくる。また 2 については、連携する他のシステムの変更や進化に応じて素早く図書館システムが適応する必要が出てくる。

これらの必要性に対し本研究では、Web2.0 時代における図書館人材育成の方法論を模索する。平成 19 年

12 月から平成 20 年 3 月の間に図書館職員及び大学職員や学生を対象として勉強会を開催し、人材育成の体制や内容としてどのようなものが必要なのかに関する知見を収集した。本論文では、この勉強会の概要と、題材として用いた Ruby on Rails について述べ、人材育成に効果の期待できるモデルカリキュラムを提案する。

2. 勉強会の背景

本研究では人材育成の具体的な方法として、勉強会形式で技術を磨く方法をとった。その理由は、業務の傍らの限られた時間の中で知識を習得するために参加者が刺激しあい、学びあいが成長することが効率がよいと考えたからである。情報技術のような専門知識の習得には、インターネットをキーワード検索しても用語そのものがわからないといった理由から、独習は難しい。そのため直接の情報交換により勉強するのが効率がよいと考えられる。また直接集まることにより参加者のコミュニティを形成し、勉強会の他の時間や期間終了後にも気軽に尋ねあえる人間関係を構築できることも向上の一助となるであろう。

このような理由で企画した勉強会だが、その達成目標は、「データベースシステムを用いた Web アプリケーションを開発できるようになる」という高い設定とした。今やほとんどの Web アプリケーションはデータベースシステムを使っており、その技術を習得するこ

[†] ひょうどう けんし 九州大学附属図書館医学分館受入目録係 E-mail: kenshi@lib.kyushu-u.ac.jp

[‡] いのうえ そうぞう 九州大学附属図書館研究開発室 E-mail: sozo@lib.kyushu-u.ac.jp

[§] まきせ ゆかり 九州大学附属図書館 e リソースサービス室 e リソースサポート係 E-mail: ymakise@lib.kyushu-u.ac.jp

(原稿受理 2008.6.4)

とは一般の Web アプリケーション技術を習得することとほぼ等価である。また次のステップとして「他の Web アプリケーションと連携したアプリケーション」も可能なら視野に入れることにした。

一方で対象者は、ワープロや表計算を使ったことがある程度の初心者を含む一般の職員及び学生という、非常に間口の広いものとした。

このように一見無理難題と思える高い設定と間口の広い対象者を設定したのは、

1. 最新の開発フレームワークの実力と効果を測るため、および
2. 人材育成の方法論について可能な限り多くの課題抽出をするため

という狙いがあったためである。

また、勉強会の会場として、古くて性能の低い、ソフトウェアのインストールができないインターネット接続端末しか望めない状況だった。そのため、このような環境でも Web アプリケーション開発の勉強会が実施できる開発環境を用意できるか検証する必要がある。

3. 関連事例

図書館システムをめぐる今後の方向性については種々の議論がなされてきた。その1つは、硬直的なパッケージシステムを導入することによる、機能拡張の難しさであり、オープンソースソフトウェアの活用^[5]や、API (Application Programming Interface) の公開^[6]を示唆している。また文献[7][8]では、実際にオープンソースソフトウェアの開発プロジェクトを紹介している。

他方、このような環境に対応できる人材育成の必要性も指摘されている。文献[8][9]においては、ソフトウェアを取り巻く人材コミュニティの重要性を指摘しているし、情報システムに限らない人材育成というテーマにおいても、情報技術に明るい人材が明言されている^[10]。また文献[11][12][13]においては情報システムに明るい図書館員であるシステムライブラリアンの資質や環境について述べている。また文献[14]では Web アプリケーションに詳しい者が直接影響を与えた事例を示している。しかしこれらのいずれも、いかにして Web アプリケーションに強い人材を育成するかという点についてカリキュラムのレベルまで実際化して議論はしていない。

4. Ruby on Rails

本勉強会では、題材として、近年注目を浴びている Web アプリケーション開発のフレームワーク、つまり

開発様式である Ruby on Rails (以下、単に **Rails**)^[15]を取り上げた。Rails は Web アプリケーションを、データを扱う**モデル**、画面デザインと遷移に相当する**ビュー**、その他のロジックを記述する**コントロール**という3つの階層に分割した MVC (Model/View/Control) アーキテクチャを持つフレームワークである。またその基本精神として、1. 同じことを繰り返さない(DRY: Don't Repeat Yourself), 2. 設定よりも規約 (CoC: Convention over Configuration) を掲げている。1は例えば、モデル層のプログラムを記述すれば自動的にそれに対応するデータベースが自動的に生成されるというように無駄な繰り返しをできるだけ減らすものである。また2は、標準的な設定は決まったルールに従い、そうでない部分の設定のみを行うようにすべきという考えである。

このような Rails 環境を勉強会の題材として取り上げた理由は、上述のようにできるだけ無駄を省いた設計思想であるため、時間的に制約のある勉強会において迅速に開発を体験することができるであろうことと、システム導入などに関わって画面遷移図やデータベース定義といったドキュメントに慣れた職員にとっては MVC アーキテクチャになじみやすいと考えたからである。

5. 勉強会の実施

5.1. 実施体制

勉強会の立ち上げについてはまず、Web アプリケーションに関心のある図書館の有志5名(教員1名、事務職員4名)で実施チームを組み、勉強会の名称を「みんなで創ろう Web アプリケーションゼミ(通称: Web アプゼミ)」として企画書を作成した。その後、事務責任者の承認を得た上で、学生には広報チラシを配布して勉強会の参加者を募り、図書館職員には正式な研修の一環として案内メールを流した。勉強会実施中、教材作成など内容に関することはチーム内の教員が中心となって準備を行い、連絡調整・会場確保など事務的なことに関しては他のメンバーで分担してその役割を担った。

5.2. 開発環境の整備

勉強会のために、情報サロンのようなソフトウェアのインストールができない Windows 端末や自宅でも参加者が開発を続けられるように、主要な環境はサーバ上に配置し、それ以外のツールは参加者が持参する USB メモリ上に配置する構成をとった。具体的には以下のような開発環境を用意した。

サーバ：

- ハードウェア：Intel Xeon 2GHz×4 個，メモリ 2GB
- OS：OpenSUSE Linux 10.2
- Rails：Rails 1.2.6
- データベース管理システム：MySQL Version 14.12

参加者ごとにユーザアカウントを作り，SSH ポートおよび，Rails に付属の Web サーバが用いるポート 3000 番以降数十個を開放して参加者ごとに 1 つずつ使用した。

参加者用 USB メモリ：

- ファイル転送ソフトウェア：WinSCP
- 仮想端末ソフトウェア：Putty
- エディタ：TeraPad

TeraPad を WinSCP に登録することで，参加者は WinSCP のサーバのファイル一覧からファイルをダブルクリックすると TeraPad で開き，保存すれば直接サーバにアップロードされるようになる。

さらに Putty およびアカウント情報を WinSCP に登録することで，参加者は USB メモリを端末にさして WinSCP を起動するだけで Putty が開き，Rails のコマンドを入力することができるようになる。これにより，利用者はあたかもローカル環境で開発を行う感覚で作業が可能となる (図 1)。

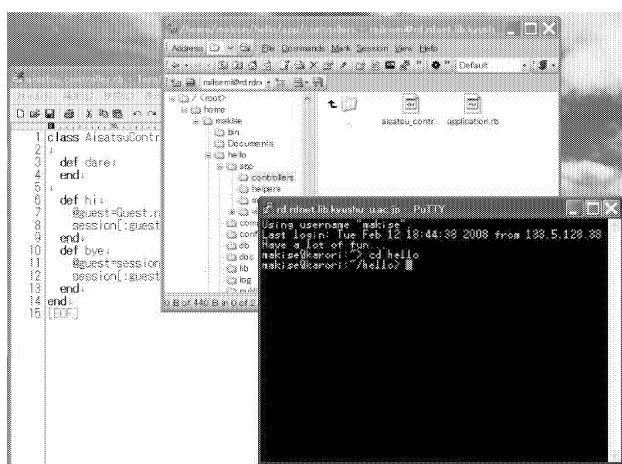


図 1 開発画面

5.3. 実施内容

勉強会は平成 19 年 12 月から平成 20 年 3 月までの 4 カ月にわたって実施された。基本は隔週ペースとした

が，参加者の都合に合わせてスケジュール調整を行い，毎回 2 時間程度，合計 6 回の勉強会を行った。また，講師はチーム内の教員が務めた。

内容としては，毎回講義と演習を組み合わせながら，第 1～3 回目までの勉強会で Web アプリケーション・データベース・Ruby 言語について基礎的な事項を取り扱い，第 4～5 回目で Web アプリケーションの開発を行い，第 6 回目で HTML・CSS の基礎を学ぶというような構成となった。その詳細と参加人数を表 1 に示す。

表 1 勉強会各回の内容・参加人数

回(日付) 参加人数	内容
第 1 回 (12/4) 16 名	<p>【講義】</p> <ul style="list-style-type: none"> ・ Web アプリケーションとは ・ Ruby on Rails の特徴 ・ プログラミング言語/フレームワーク/MVC モデル ・ プログラムとは ・ 手続き/データ/クラス/オブジェクト指向 <p>【演習】</p> <ul style="list-style-type: none"> ・ 挨拶を返すアプリケーションの作成 ・ プロジェクト作成→モデル作成→コントローラ作成→ビュー作成→緒準備→サーバ起動→利用→サーバ終了
第 2 回 (12/18) 13 名	<p>【講義】</p> <ul style="list-style-type: none"> ・ 前回のプログラムの説明 ・ Web の基本/MVC モデルとは/プログラムとは/ソースコード ・ リレーショナルデータベース ・ サーバー上のコマンド習得 <p>【演習】</p> <ul style="list-style-type: none"> ・ ブログアプリケーションの作成 ・ プロジェクト作成→データベース作成準備→データベース作成→マイグレーション準備→マイグレーション→雛形の作成→サーバ起動→利用→サーバを終了
第 3 回 (1/15) 13 名	<p>【講義】</p> <ul style="list-style-type: none"> ・ 前回の復習 ・ Web アプリケーションとは/プログラムとは/データベース ・ Ruby の文法の実習 ・ 数のプログラム/文字列/変換/変数/分岐/while ループ/メソッド/局所変数/クラス/インスタンス変数/配列/ハッシュ ・ 前回作成したプログラムを文法的側面から復習 <p>【演習】</p> <ul style="list-style-type: none"> ・ 学習した文法を使ったプログラムの作成 ・ 6 つの候補から新年会のお店をランダムに決定するプログラムの作成
第 4 回 (1/29) 10 名	<p>【講義・演習】</p> <ul style="list-style-type: none"> ・ これまでの学習を応用し，自分の作りたい Web アプリケーションを作成 1. 作りたいものを発表 ・ メニューリスト・絵本データ・ToDo リスト... 2. 紙に書く ・ データベースの構造を表にする/画面イメージ 3. プログラミング

第5回 (2/21)	<p>【講義】</p> <ul style="list-style-type: none"> ・2日目（データベース作成）の復習 データベースと Scaffold の作成の簡単な方法 ・1日目（ページ遷移を作成）の復習 <p>【演習】</p> <ul style="list-style-type: none"> ・前回の反省点、未完成な部分、発展させたい点等について発表 ・プログラミング
第6回 (3/10)	<p>【講義・演習】</p> <ul style="list-style-type: none"> ・デザイン HTML 入門/スタイルシート (CSS) 入門 ・複数のデータベースを作成 テーブル間の関係を指定

この表 1の内容は、立ち上げ当初から決定されたものではなく、参加者の反応を見ながら毎回の試行錯誤を通じて、その都度決めていったものである。この他にも、支援体制として、普段の勉強会だけでは内容が身につかない、これまでの復習をしたいが独学では難しい、という参加者のために、皆で集まって自習を行う日を別途設けた。

また、参加者との連絡調整ツールとして、同時に SNS でこの勉強会のコミュニティを立ち上げた。採用した SNS は本学図書館と連携を行っている VARRY^{[16][17]}である。具体的に SNS 上で行った活動としては、ロゴの作成 (図 2)、スケジュールのアナウンス、出席確認、勉強会のまとめ、感想・意見の収集、参考文献の紹介であった。



図 2 勉強会ロゴ

6. 成果と課題

6.1. 成果

勉強会の実施により得られた成果としては以下のよう
な事項を列挙することができる

内容について

(ア) 講師の側にとっては、Web アプリケーションの開発にあたって、HTML さえも知らない人に教えるべき最低限の要素が何であるかということについての確認ができた。具体的には、Web ページ構築のための HTML や CSS、Ruby やオブジェクト指向といったプログラミング言語、リレーショナルデータベース、

Ruby on Rails や MVC モデルといった Web アプリケーション開発のフレームワークの知識である。

(イ) 演習については、1 回目の勉強会で画面遷移のある Web アプリケーション、2 回目でデータベースを利用した Web アプリケーションの作成を行った。このように、初回の勉強会からでさえ、一定の手順に従えば、ほとんどの参加者が講師の用意した Web アプリケーションを再現することができた。

(ウ) 参加者のうち数人は、以下のような簡単な Web アプリケーションを完成させることができた。

- ① フランスの哲学者に関する図書の分類記号管理アプリケーション (図 3)
- ② 飲食店の料理メニュー公開アプリケーション (図 4)

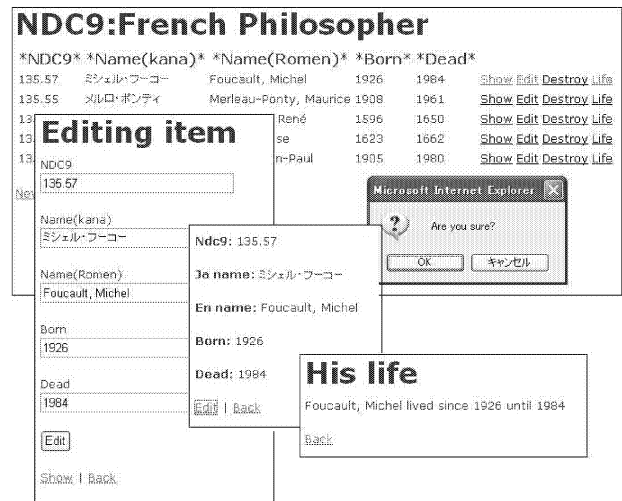


図 3 勉強会参加者が開発したアプリケーション 1

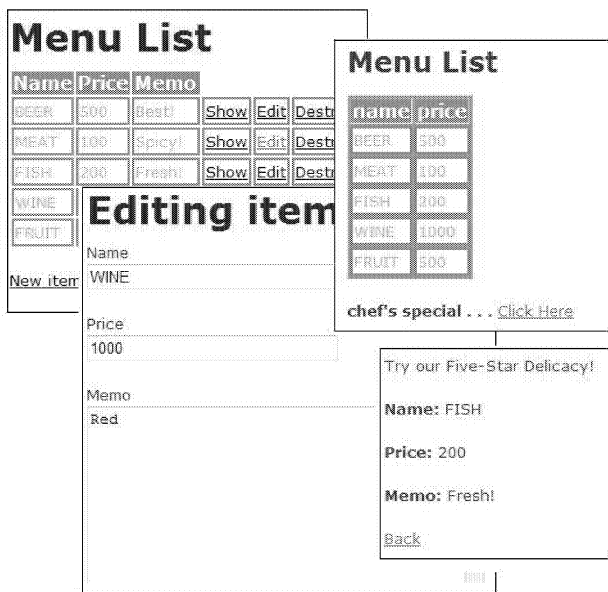


図 4 勉強会参加者が開発したアプリケーション 2

運用について

- (エ) 5.2節で述べた開発環境も問題なく使うことができた。Netbeans^[18]などの統合開発環境をローカルマシンにインストールするのに比べてエディタの補完機能などが無いのに、Putty というコマンドライン端末での操作をある程度覚えなければならなかったが、それを除けばスムーズに演習が可能であった。
- (オ) スケジュール通知や意見収集などの連絡調整ツールとして、SNS は問題なく機能した。また、勉強会に興味を持っているが本人の都合により参加できない人には、SNS 上のコミュニティのサイトにより活動内容を案内することができた。

6.2. 課題

課題としては以下のような事項を列挙することができる。

内容について

- (カ) Web サーバの立ち上げや開発環境の構築など、準備設定に係る技術的な体験を経ないため、参加者にとっては Web アプリケーション開発環境の整備について全体像を把握できない。
- (キ) 参加者の間で講義の理解度および演習の作業ペースに大きなばらつきが見られるようであり、講師の側としては偏りなく参加者全

員に内容を理解させる工夫が必要である。

- (ク) 内容を理解せずとも、一定の手順に従えば、Web アプリケーションが簡単に作成できてしまう。そのため、その仕組みを理解できていない人にとっては、いったんエラーがでてしまうと、何が間違っているのかの原因が全く分からず、その復旧が難しい。なお、これは Ruby on Rails のエラーメッセージが不親切であるということも要因の一つとしてあげられる。Ruby はいったんコンパイルされることなく動的に実行される言語であるため、型チェックなどのデバッグが難しい。オブジェクト指向を完全に理解しないと利用が難しいといった問題もある。

- (ケ) 参加者の漸次減少(表 1)が見られたが、その原因としては以下のようなものが考えられる。

- ① 複数回の勉強会を通して一つのプログラムの改良を繰り返す形式だったため、一度でも脱落してしまうと復帰が難しかったのではないかと。
- ② フレームワークという最新の方法が、現時点での業務の現場での必要性や速効性が実感できなかったため、動機づけが難しかったのではないかと。

運用について

- (コ) SNS は、連絡調整ツールとして問題なく機能したが、一方で多少の物足りなさも感じた。実際には、ドキュメントなどの共有機能や、参加者の開発手順のログを共有できる機能、デバッグのための Q&A といった知識共有機能の必要性が確認できた。

7. モデルカリキュラムの提案

今回の Ruby on Rails 勉強会実施の成果と課題を踏まえて、効果のありそうなモデルカリキュラムを提案する。今回と同様に、Ruby on Rails による Web アプリケーション開発を勉強会のトピックとし、対象者を Web プログラミングの初心者に設定する。

7.1. モデルカリキュラム

提案するモデルカリキュラムを表 2 に示す。

表 2 モデルカリキュラム

回	トピック 各回、講義+演習として2~4時間程度
第1回	Web ページ構築のための HTML や CSS
第2回	Ruby やオブジェクト指向といったプログラミング言語 数/文字列/変換/変数/分岐/while ループ/ メソッド/局所変数/クラス/インスタンス変数/ 配列/ハッシュ
第3回	リレーショナルデータベース 複数テーブル/テーブル間関係
第4回	Web アプリケーション開発のフレームワーク Ruby on Rails/MVC モデル
第5回	開発環境の構築 サーバ管理/インストール

表 2では、初心者でも開発ができるようになるための6.1節(ア)で示した最低限の要素を一通りそろえている。さらに、これらに加えて、6.2節(カ)に対応するために、サーバ管理やインストールの経験など開発環境の設定に関する時間も最後に追加している。また、これだけの要素では満足できない中級者向けの内容として、XML, API の利用, プラグイン開発, デバッグ, テスト, Verify や Paginate など Ruby の上級メソッド, セキュリティやバックアップなどサーバ管理の上級を組み合わせたのも有効であろう。

表 2の新カリキュラムが今回の勉強会と異なる点としては、その要素の教える順序を変更したことである。今回は、フレームワークという概念的要素から始めて、そのフレームワークを構成する個別要素へトップダウンに勉強していく方式であった。しかしこれは、あまり知識のない者にとって、いきなり複雑な要素を提示されるため、心理的な抵抗感が生まれ、参加者によっては6.2節(キ)(ク)のように内容理解があまり進まない側面のある方式であったように思われる。よって新カリキュラムでは、理解促進を図るために、初心者でも取り組みやすいHTMLやCSSを初回の講義とし、個別的な要素から次第に知識を積み上げていくボトムアップ方式の順序へと全体の構成を転換した。また、6.2節(ケ)①への対処として、全回参加を予定しない受講者でも取り組めるように、第1回から第3回までは1回完結の内容とし、他の回と依存関係のない構成としている。

7.2. その他改善すべき点

内容以外に改善すべき点としては、6.2節(キ)(ク)に

ついて参加者の理解促進を図るための措置が重要であろう。例えば、Ruby Business Commons^[19]という Ruby 言語に関する団体が開催している勉強会のように、

- ある程度の知識を持った参加者にチューター的な役割を担ってもらい、
- グループ制により自発的な協力体制を構築するという対策が考えられる。また、宿題や補講などにより参加者の理解をフォローアップする機会を設けたり、Web 自習環境を整えたり、Q&A を整理するのも有効であろう。

演習の方法としては、今回のように1からプログラムを作成して毎回改良を繰り返す形式では、6.2節(ケ)①の問題が発生することから、別のやり方も選択肢として考慮すべきである。例えば、既に完成されたサンプルプログラムをサーバにいくつか置いておき、そのプログラムを改造していくという方式を採用してもよいのではないだろうか。この方式の利点としては、それまでの演習の積み重ねをあまり必要としないため、途中の回だけ参加する者であっても取り組みが容易となることが挙げられる。また、既に完成した使えるアプリケーションの改良なのでその実効性が目に見えやすく、6.2節(ケ)②の対策ともなり、参加者の動機付けも高まるのではないかと。さらに、図書館の現場としては、1からプログラムを作成するというよりも、機関リポジトリの構築などに見られるようにオープンソース等の既存のプログラムを改良するという事例の方が多と思われる。そういう意味では、新しい演習方式の方が、実業務に近いためインセンティブも高まるであろう。

その支援環境としては、6.2節(コ)で述べたとおり、

1. 資料共有機能
2. 開発手順のログを共有できる機能
3. Q&A といった知識共有機能

を用意することが考えられるが、特に2のようなシステムは我々の知る限りないので、1の資料参照や、3のデバッグにおける連携なども含めて今後の課題となるであろう。

またこれに近い機能として Redmine というプロジェクト管理システム^[20]が Ruby on Rails で開発されているのも興味深い。本勉強会の中でも終盤になって使い始めたところである。

8. おわりに

このように、講義と実習という形で勉強会を行うこ

とにより、図書館における Web アプリケーション開発に対する人材育成の方法論について実際的な知見を得ることができた。人材育成の問題は、現在行われている CSI 事業^[21]においても種々のシステムが開発されていることを考えると、そのうち大きな問題となると思われるため、本研究のような知見を蓄積することはそれに応じて重要となってくる。

謝辞

最後に、本研究の勉強会を開催するにあたって、ご協力いただいた皆様へ感謝します。勉強会を盛り上げてくださった参加者の皆様、また、この活動を支持してくださった図書館の皆様、どうも有り難うございました。

参考文献

- [1] Wikipedia, <http://www.wikipedia.org/>, (参照 2008-06-02).
- [2] Mixi, <http://mixi.jp/>, (参照 2008-06-02).
- [3] Amazon, <http://www.amazon.co.jp/>, (参照 2008-06-02).
- [4] Flickr, <http://www.flickr.com/>, (参照 2008-06-02).
- [5] 国立大学図書館協会学術委員会図書館システム検討ワーキンググループ, 今後の図書館システムの方向性について, 2007, 61p. http://wwwsoc.nii.ac.jp/anul/j/projects/si/systemwg_report.pdf, (参照 2008-06-02).
- [6] 国立情報学研究所学コンテンツ運営・連携本部図書館連携作業部会次世代目録ワーキンググループ, 次世代目録所在情報サービスの在り方について(中間報告), 2008, 25p. http://www.nii.ac.jp/CAT-ILL/archive/pdf/next_cat_interim_report.pdf, (参照 2008-06-02).
- [7] Project Next-L, <http://next-l.slis.keio.ac.jp/wiki/wiki.cgi>, (参照 2008-06-02).
- [8] 原田隆史, 図書館員自身の協同で作る図書館システム仕様: 日本発のオープンソース図書館システム作成を目指して, カレントアウェアネス, 2007, 292, p.4-6. <http://current.ndl.go.jp/ca1629>, (参照 2008-06-02).
- [9] 原田隆史, オープンソースと統合図書館システム, カレントアウェアネス, 2006, 289, 15-18. <http://current.ndl.go.jp/ca1605>, (参照 2008-06-02).
- [10] 国立大学図書館協会人材委員会, 大学図書館が求める人材像について -大学図書館職員のコンピテンシー-, 2007, 24p. <http://wwwsoc.nii.ac.jp/anul/j/projects/hr/jinzaizo1903.pdf>, (参照 2008-06-02).
- [11] 宇田則彦, 総論: システムライブラリアンをめぐる状況と課題, 情報の科学と技術, 2006, vol.56, no.4, p.150-154.
- [12] 中尾康朗, 永井善一, サービス指向環境下におけるシステムライブラリアンの役割とスキル, 情報の科学と技術, 2006, vol.56, no.4, p.155-160.
- [13] 丸山高弘, 山中湖情報創造館のデジタルライブラリアンがシステムライブラリアンについて語るいくつかの事柄, 情報の科学と技術, 2006, vol.56, no.4, p.161-165.
- [14] 深澤良彰, オープンソースにおける大学の役割: 早稲田大学を例として, OSSAJ オープンソースセビジネスミナー資料, 東京, 2005-05-30, オープンソースソフトウェア協会, http://www.ossaj.org/seminar/050530/ossaj_20050530_2.pdf, (参照 2008-06-02).
- [15] Ruby on Rails, <http://www.rubyonrails.org/>, (参照 2008-06-02).
- [16] Varry, <http://www.varry.net/>, (参照 2008-06-02).
- [17] 井上創造, 堀優子, SNS における新しい信頼モデルと図書館における応用, 九州大学附属図書館研究開発室年報, 2007, 2006/2007, p.22-29. <http://hdl.handle.net/2324/8087>, (参照 2008-06-02).
- [18] 統合開発環境 Netbeans, <http://ja.netbeans.org/>, (参照 2008-06-02).
- [19] Ruby Business Commons, <http://qwik.rubybizcommons.jp/>, (参照 2008-05-15).
- [20] プロジェクト管理システム Redmine, <http://www.redmine.org/>, (参照 2008-06-02).
- [21] 学術機関リポジトリ構築連携支援事業, <http://www.nii.ac.jp/irp/>, (参照 2008-06-02).