

書籍販売サイトと連携した 研究室規模の蔵書管理システム - DB 設計と基本機能 -

川鯉 光起[†] 富永 浩之[†]

書籍販売サイトと連携した研究室単位の蔵書管理システムを提案する。登録作業の効率化のため、バーコードリーダーで ISBN 番号を読み取り、Amazon API で図書情報を取得する。学生証を IC カードリーダーで認識してユーザを管理する。貸出機能だけでなく、購入希望も管理する。情報系の解説書などに対して、利用者の状況に応じて、既読者からの助言やプログラム例を付加情報として閲覧できるようにする。本論では、DB 設計と基本機能について報告する。

Library Management System for Laboratory Cooperated with Book Seller Site - Database Design and Basic Functions -

Koki KAWAGOI[†] Hiroyuki TOMINAGA[†]

A laboratory in an engineering college possesses many books about technology and science. While members in the laboratory mainly use the books, outside students sometimes borrow them. Members must manage the collection by oneself. We propose a library management system for the laboratory scale. The system cooperates with some book selling sites. It reads an ISBN of a book by a bar code reader to acquire the information by Amazon Web service. It adopts IC chip reader for user check with a student card. The system offers purchase helping functions. It gathers book requests of members and tells a teacher to buy by a cart method of API. We also consider learning support functions, especially for books about information engineering. A user who have read a book submits his comment for understanding. He may upload his program as sample of practice. The following users can access restrictively for the educational purpose. In this paper, we report database design and basic functions of the system.

1. はじめに

本研究室では、教員の蔵書およびサークルで購入した図書を共同で管理している(図

1). 学生が卒業時に教科書などを寄贈することもある。分野は、情報系や数理系を中心として、就職関連や娯楽的な本も含まれる。ゼミ生やサークル生だけではなく、教員の授業を受講する学生も図書を借りに来ることがある。このような研究室規模の蔵書管理をオンラインで行うには、個人向けのツールでは、不十分である。また、大量の蔵書の登録作業がネックになる。本研究では、書籍販売サイトと連携した研究室規模の蔵書管理システムの開発を目指している。



図 1 研究室の本棚



図 2 教職員証と IC カードリーダー

2. 蔵書管理と書籍販売サイト

2.1. IC チップを搭載した学生証と IC カードリーダー

本学では、IC チップを搭載した教職員証と学生証が導入されている(図 2 左)。機械による本人確認が可能となり、入退室や証明書の自動発行、出席確認に用いられている(図 2 中)。Felica チップの機能で、生協でのプリペイドカードや琴電の IruCa カードとしても利用できる[1]。USB 接続の IC カードリーダーも普及し、ノート PC から簡単に利用できるようになった(図 2 右)。IC カードを用いて、研究室に出入りする学生の管理にも活用することが考えられる。

[†]香川大学工学部
Faculty of Engineering, Kagawa University

2.2. ISBN とバーコード

出版業界や図書館では、書籍の識別のため、国際標準の ISBN が定められている[2]。以前は 10 桁の ISBN-10 であったが、番号が枯渇に近付いたため、2007 年から 13 桁の ISBN-13 に移行した。前者から後者への変換が可能である。ただし、国内では、ISBN を基に付加的な情報を加えた、日本図書コードや書籍 JAN コードが使われている。ISSN(International Standard Serial Number)は、国際標準逐次刊行物番号である。日本では、国立国会図書館が管理を行っており、学会誌などに付与されている。国内の多くの雑誌は、これとは別の定期刊行物コードで識別されている。不定期の刊行物、付録、小冊子、資料などには、何も与えられていないことが多い。研究会やプロジェクトの報告などは、この場合もある。また、最近の書籍の裏表紙には、上記の番号がバーコードで付けられている(図 3 左)。バーコードリーダーも USB 接続の安価な機器が入手できるようになってきた(図 3 右)。簡易 DB に取り込めるツールも公開されている。これらを使えば、膨大な蔵書の登録作業が簡素化できる。



図 3 ISBN のバーコードとバーコードリーダー



図 4 Books.or.jp と Amazon の画面

2.3. 書籍の検索サイトと販売サイト

近年、書籍の検索サイトや販売サイトが幾つか運営されている。サイトによっては、書籍名や著者名などの単純なキーワードだけでなく、販売ランキングや他のユーザーによる書評などからも検索できる。また、取得できる書籍情報として、表紙の画像、目次、本文の一部など、実際に本屋で手に取るかのような情報も得られる。サイトの中には、Web サービスが提供されており、各自の Web ページからサイトのサービスを利用したアプリケーションを開発できる。これにより、書名、著者、出版社、発行年などの基本的な書籍情報を検索したり取得することができる。他にも、購入直前までの処理を代行できる。また、表紙の画像、目次、本文の一部など、ISBN を指定して Web 上から書籍情報を取得すれば、充実した DB の構築が容易になる。

Books.or.jp <http://www.books.or.jp/>

国内で発行され、現在入手可能な書籍を収録する(図 4 左)。各出版社から提供された書籍情報は、日本書籍出版協会の「データベース日本書籍総目録」に蓄積されている。収録データは、そのうちの入手可能な既刊分の約 85 万点である。書名、著者名、出版社から検索できる。

三省堂 <http://www.books-sanseido.co.jp/reserve/bookSearch.do>

書名や著者やその他項目で検索が行える。表紙画像を確認することができる。書名、著者、出版社、価格、発行年月、ISBN の情報が得られる。そのまま、取り置きが行える。

イーブックオフ <http://www.ebookoff.co.jp/index.jsp>

表紙画像を確認することができる。書名、著者、出版社、価格、発行年月、ISBN の情報が得られる。他に、中古在庫のない商品に対しても、中古価格の情報が得られる。楽天ブックスから提供された商品紹介を閲覧できる。入荷メールを登録することができる。

楽天ブックス <http://books.rakuten.co.jp/>

他の書籍情報サイトで得られる情報に加えてレビュー機能がある。オークションやショップなど、楽天に関連のあるサービスから情報を取得できる。書籍によっては、数ページのチラ読みができる。

Amazon <http://amazon.co.jp/>

他の書籍情報サイトで得られる情報に加えてレビュー機能がある(図 4 右)。詳細な情報として得られる情報に次のようなものがある。本のタイプ、出版社、ISBN-10、ISBN-13、発売日、商品の寸法、内容や著者の略歴が得られる。また、ユーザが自由に作成できる書籍リストやほしいもののリストがある。

国立国会図書館サーチ <http://iss.ndl.go.jp/>

書籍に加え、雑誌、論文、新聞、児童書、DVD、CD、デジタルアーカイブも検索

できる[3]。検索条件は、書籍名、著者名、出版社、出版年、分類記号、ISBN、ISSN などである。翻訳検索も行える。現在、34 個のデータベースから収集した約 5,500 万件の文献情報を検索できる。また、9 個のデータベースを横断検索できる。全文テキスト化された資料に関しては、書誌情報だけでなく、本文の全文検索ができる。API による検索対象範囲は、プロトコルによって異なる[4]。SRU、SRW、OpenSearch、OpenURL、Z39.50 に対応している。ハーベスト用 API では、国立国会図書館サーチに格納されたメタデータのダウンロードができる。OAI-PMH に対応している。

2.4. 既存の蔵書管理システム

個人向けや小規模の図書館用の蔵書管理ツールには、以下のものがある。

私本管理 Plus <http://homepage1.nifty.com/EKAKIN/shihonp.htm>

蔵書管理用のデータベースソフトである(図 5 左)。ISBN で Amazon から検索し、書籍名、著者名、出版社、価格、発行日などの書籍情報を一括登録できる。書籍名、著者名、出版社名から、ISBN を取得する機能もある。未購入で、ISBN が不明の書籍情報も登録できる。また、複数の巻が出版されているシリーズものの小説や漫画の書籍情報を、一括で登録することもできる。その他、登録された書籍情報を全文検索する機能や、データをバックアップしたり、CSV 形式でエクスポートする機能を備える。同じ作者の CGI「私本管理 GOOUT」を利用して、ソフトのデータを PC や携帯電話の Web ブラウザ上から一覧表示したり、検索できる。

Bookshelf Application <http://www.woodensoldier.info/soft/BookshelfApplication.htm>

書籍、CD、DVD などを統合的に管理する(図 5 右)。Amazon の商品検索ツールとしても使える。1 つのキーワードで複数のジャンルにまたがる製品を検索でき、各種のオプションで、表紙などの画像やカスタレビューも閲覧できる。ASIN(ISBN)リスト形式でデータの取込みが可能である。未読/既読の区別や、読んだ本から得た知識など、商品ごとにメモを記録でき、個人の知識データベース作成を支援する。ツリービューを利用したインタフェースで、お気に入りをドラッグ&ドロップで整理できる。ローカルショッピングカートを搭載し、マーケットプレイスの価格を確認する機能もある。blog で紹介するためのアフィリエイト用の HTML 出力が行える。お気に入りの商品を知合いに勧めるため、PC メールや携帯メールへの送信もできる。

マイ・カタログ <http://www.vector.co.jp/soft/winnt/personal/se471694.html>

音楽 CD、ビデオ、書籍などの収集品をテキストと画像で記録する。登録後は、カテゴリ検索、印刷機能で閲覧できる。また、Web サイトからの情報取得機能により、登録作業の入力を軽減する。

文籍(Monjack) <http://hp.vector.co.jp/authors/VA036268/MonJack/>

学校図書館の貸出業務を電算化するソフトウェアである。主に、高等学校や中小学校の図書室での利用を想定している。自宅や大学の研究室での蔵書管理にも利用でき

る。Web データベースを作成可能である。

お気軽図書館システム <http://www.t-wings.jp/tosyo/>

学校の図書室やミニ文庫用の図書館システムである。書籍 JAN コードをバーコードリーダで読み取り、書籍名、著者名、出版社、出版年月日などを自動入力できる。貸出と返却についても、バーコードで対応する。病院、企業、団体などで、蔵書管理と貸出返却処理を行うのに向いたデータベースアプリケーションである。



図 5 私本管理 Plus と Bookshelf Application

2.5. Amazon Web サービス

Web サービスとは、XML ベースのプロトコルでアクセスできるインターネット上のサービスのことである。Amazon Web サービス(AWS)は、Amazon が提供する Web サービスである[5][6]。AWS を利用し、個人の Web ページに Amazon の商品データを掲載することができる。商品の検索機能や、ランキング表示、ショッピングカートへの追加など、Amazon サイトが提供する多くの機能を利用できる。アソシエイト・プログラムを利用し、リンク先の商品が購入された場合に、Amazon から紹介料を受け取るサービスもある。AWS の利用は無料だが、ID 登録が必要である。アソシエイトを利用する場合は、アソシエイト ID も必要である。

AWS では、個人サイトから Amazon の商品情報にアクセスして、XML データを自動的に取得する。データの取得方法には、REST(Representational State Transfer)と SOAP(Simple Object Access Protocol)の 2 種類の方法がある。両者のうち、REST の方が実装が簡単であり、サービスの利用者の 85% が REST を利用している。REST 方式では、取得したい商品の ID を、URL (<http://webservices.amazon.co.jp/onca/xml?>)への GET パラメタとして付加すれば、XML データが取得できる。また、検索キーや条件を表すパラメタも付加すれば、様々なデータを取得できる。指定できるパラメタには、以下のようなものがある。複数のパラメタを設定するときは、それぞれのパラメタを&で繋ぐ。

サービス名	Service=サービス名
基本的には Service=AWSECommerceService と指定	
アクセス ID	AWSAccessKeyID=登録 ID
アソシエイト ID	AssociateTag=アソシエイト ID
操作の種類	Operation=操作
ItemSearch	データベースから商品の情報を検索
ItemLookup	ASIN や ISBN で商品を検索
SimilarityLookup	関連商品を検索
SellerListingSearch	マーケットプレイスの出品情報を検索
SellerListingLookup	マーケットプレイスの出品者 ID から出品情報を検索
商品ジャンルの指定	SearchIndex=ジャンル
Blended: 全商品	Books: 和書 ForeignBooks: 洋書
Music: ミュージック	Video: ビデオ
検索パラメタの設定	Keywords=キーワード
Operation に ItemSearch を指定した場合にキーワードを指定	
日本語をキーワードは UTF-8 の文字列を URL エンコードして指定	

取得したデータの表示については、XSLT エンジンが提供されている。XSLT スタイルシートを用意すれば、目的に応じて、XML データを整形して表示できる。XSLT スタイルシートを適用するには、情報を取得する際に、次のようなパラメタを付加する。

スタイルシートへのパス	Style=パス
表示のフォーマット	ContentType=フォーマット
HTML での表示は ContentType=text/html	
使用するバージョン情報	Version=バージョン

2.6. Amazon API

AWS を効率的に利用するため、第三者によって各種の API が開発されている。Java[7][8], PHP[9][10], Ruby[11][12][13][14] 言語などに対応したものがある。本研究の開発言語である Ruby 用では、amazon-ecs, ruby-amazon, ruby-aws, ruby-aaws などがある。ただし、2009 年 8 月 15 日から、AWS は、署名認証を含まないリクエストを処理しないように変更された。そのため、これ以降に更新されていない API では対応できない。本研究では、AWS の最新版に対応していて多機能な ruby-aaws を採用する。この API では、購入のためのカート機能が用意されている。

● Java	
○ AWS SDK for Java	クラウド EC2 やストレージ S3 を Eclipse から利用
○ Amazon EC2 API Tools	クラウド EC2 をコマンドラインで利用
● PHP	
○ PEAR::Services_Amazon	データ取得のラージモードに対応せず
○ PHP Amazon Search	検索のみを行う API
● Ruby	
○ amazon-ecs	2007.06 更新 カート機能無し
○ ruby-amazon	2008.03 更新 カート機能有り
○ ruby-aws	2007.10 更新 カート機能無し
○ ruby-aaws	2010.05 更新 カート機能有り

2.7. ruby-aaws

Ruby-aaws は、Ian Macdonald が開発した。現時点での最終バージョンは、2010 年 5 月 20 日にリリースの 0.8.1 である。購入処理に関するクラスとメソッドが充実している。書籍を購入するには、カート(Cart)クラスを利用する。カートには、Active 領域と Later 領域(今は買わない)がある。リストには、wishlists, や registry lists などがある。

商品の検索	Amazon::AWS::ItemLookup
商品の検索	Amazon::AWS::ItemSearch
リストの検索	Amazon::AWS::ListLookup
リストの曖昧な検索	Amazon::AWS::ListSearch
特定の販売業者が販売する商品を検索	Amazon::AWS::SellerListingLookup
指定された販売業者の商品	Amazon::AWS::SellerListingSearch
カート関係	Amazon::AWS::ShoppingCart::Cart
カートの準備	cart_create
リモートのカートを取得	cart_get
商品の追加	cart_add
商品の全削除	cart_clear
商品の個数変更	cart_modify
指定した商品がカートの中にあるか確認	include?
Active 領域のアイテム確認	active?
Later 領域にあるか確認	saved_for_later?

3. Amazon API と連携した登録部と検索部

3.1. 提案するシステム

本研究では、書籍販売サイトと連携した研究室規模の蔵書管理システムを提案する[15]。システムの全体構成は、図 6 の通りである。現在、主要な DB と基本機能を実装中である。総称的な著作物としての書籍と、個々の具体物としての蔵書の情報を整理し、検索、登録、貸借、購入の作業を効率化する。例えば、実際の本棚の収納位置に合わせ、表紙や背表紙の画像を並べ、書籍名が似た蔵書や関連する蔵書を探しやすくする。また、蔵書の有無を忘れて重複購入してしまうことを防ぐため、書店からのモバイル端末での利用にも対応する。電子書籍も扱えるようにし、輪講の補助を行う。ユーザにグループを作成し、グループ毎に、権限を変更できるようにすることで、複雑な貸出のルールに対応する。既存のシステムとの差異としての学習支援では、本の要約やソースコードの共有を行えるようにする。本の要約やソースコードは、研究室で既に行なっている、小レポートのページから取得する。既存のサイトを参考にしながら、ユーザ評価を行って、使いやすいインタフェースを提供する。

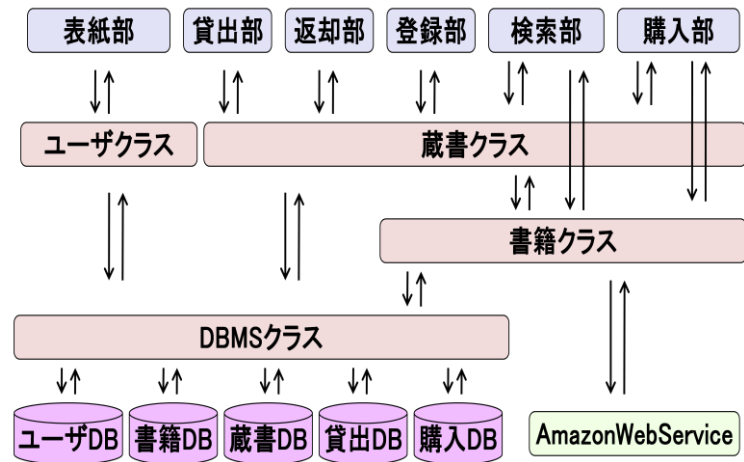


図 6 蔵書管理システムのモジュール構成

3.2. 書籍 DB

書籍 DB では、Amazon から取得した書籍情報などを保存する(表 1)。書籍には、ISBN が付与されているものと、そうでないものがあるため、書籍 ID で識別する。また、

貸出に利用したり、検索に利用するため、ISBN も保存する。ISBN には、10 桁と 13 桁があるので情報を減らさないために、両方を保存する。著者が複数の場合があるので、全て保存する。また、著者は、英名(アルファベット)と和名を分けて保存する。監修や翻訳など、著者の役割についても、保存する。Amazon において、翻訳本の場合は、同一人物が著者と原著として分けられていることがある。そのとき、一方がカタカナで一方がアルファベットのことがある。本システムでは、同一人物が英名と和名で記述されていた場合、同じ人物の英名と和名で保存する。そのために、修正が必要な場合があるので修正フラグを用意する。表紙画像は、Amazon サイトの URL を保存する。また、書籍の内容を大まかに知るため目次データを保存する。関連情報を同時に得るため、関連リンクも保存する。検索に利用するため、キーワードの保存も行う。

表 1 書籍 DB のスキーマ定義

部	項目	属性名	型	例
管理	書籍 ID	book_id	serial	1
	ISBN(13 桁)	book_isbn13	text	9784396314415
	ISBN(10 桁)	book_isbn10	text	1234567890
	修正フラグ	book_recheckflag	bool	1234567890
題名	書名	title_name	text	初めての Ruby
	巻名	title_volumename	text	初めての Ruby
	巻数	title_volumeno	integer	1
著者	日本語名	author_jp[]	text	Yugui
	英名	author_en[]	text	Yugui
	役割	author_type[]	text	著者
情報	発行年	bookinfo_year	integer	2009
	価格	bookinfo_price	integer	1480
	通過	bookinfo_currency	text	JYP
	出版社	bookinfo_publisher	text	白鷗社
	分野	bookinfo_genre	text	Ruby
	キーワード	bookinfo_keyword	text	Ruby
	目次データ	bookinfo_Index	text	
	表紙リンク	bookinfo_coverlink	text	http://
	関連リンク	bookinfo_relatedlink	text	http://

表 2 蔵書 DB スキーマ定義

部	項目	属性名	型	例
管理	蔵書 ID	biblio_id	serial	1
	書籍 ID	biblio_bookid	text	1234567890
	重複番号	biblio_number	integer	3
	存在フラグ	biblio_existflag	bool	TRUE
	有効フラグ	biblio_activeflag	bool	TRUE
	区別フラグ	biblio_relationflag	bool	TRUE
状態	状態 ID	state_id	integer	1
情報	登録者	biblioinfo_register	integer	3
	登録日時	biblioinfo_regdate	date	2010/10/5
	所有者	biblioinfo_owner	integer	3
	本棚位置 ID	biblioinfo_shelfid	integer	2

表 3 本棚 DB スキーマ定義

部	項目	属性名	型	例
管理	本棚 ID	shelf_id	integer	3
分類	分類名	genre_name	text	プログラミング
位置	部屋	position_room	integer	3
	列	position_column	integer	2
	行	position_row	integer	3
	左右中	position_align	integer	1

3.3. 蔵書 DB と本棚 DB

蔵書 DB では、蔵書を管理するための情報を保存する(表 2)。蔵書を識別するために、蔵書 ID を保存する。どの書籍なのか識別するために、書籍 DB の外部キーとなる書籍 ID を保存する。同じ書籍の何冊目か分かるように重複 ID を保存する。蔵書が存在しなくなった場合や、利用できなくなった場合に、蔵書情報を削除せずにすむように、存在フラグと有効フラグを用意する。同じ書籍が複数あるとき、区別する必要がある場合とない場合が考えられる。前者は、所有者が異なる、一方が傷んだり書込みがある、

などの場合である。そのため、区別フラグを用意する。また、貸出中や持ち出し禁止など、蔵書の状態を保存する状態 ID を保存する。また、データの登録者や登録日時の情報も保存する。本の格納位置を特定するため、本棚位置を保存する。本棚には、位置ごとに分類も決めているため、別の DB で扱う。

本棚 DB では、本棚の位置と書籍の分類を保存する(表 3)。本棚を識別するために、本棚 ID を用意する。蔵書は情報系分野がほとんどであるため、書籍の分類は、独自に分野を設定する。例えば、UNIX, DB, アルゴリズム, ゲームプログラミング, LEGO プログラミング, 知識工学, 各種プログラム言語(C, C++, Java, Ruby, PHP, Perl, BASIC, JavaScript)などである。就職対策, 資格試験, 一般常識などの分野も設ける。プログラム言語は, Ruby 言語だけでも数十冊あるため, 入門, Rails などと細分化する。また, 学年の変わり目など, 卒業生が大量に寄贈する場合など, 蔵書数に応じて分類を再編する。本棚の 1 つの棚には, 50 冊程度を収納できる。細分化の必要に応じて, これをブックエンドで仕切り, 左右中の区画を設ける。本棚の位置情報として, 本棚のある部屋番号, 本棚の列と行, 棚の中の左右中を保存する。

3.4. 登録部

登録部では、書籍 DB へ書籍情報と蔵書 DB へ蔵書情報の登録を行なう(図 7)。書籍情報は、書籍それぞれにある書籍固有の情報である。ISBN コードをキーとして Amazon API を利用して Amazon から取得する。ISBN コードは、バーコードリーダーで読取ることができる。ISBN コードのない書籍や Amazon から情報が取得できない場合は、手入力で書籍情報を入力する。今後は、国立国会図書館サーチなど、他の書籍情報サイトから情報を取得できるようにする。蔵書情報は、本システムが扱う書籍 1 冊 1 冊に別々に保存された情報である。同じ書籍が複数ある場合は、別の蔵書として管理する。登録時に手入力する必要がある。その際には、キーボードだけではなくタッチパネルも利用して入力することができる。書籍の登録方法は、ISBN コードの有無で分かれる。また、ISBN コードがある書籍も、Amazon から情報を取得できるものと、そうでないものに分かれる。高校の教科書などの書籍情報は得にくい。ISBN コードがあり、Amazon から情報を取得できる書籍は、API を利用して Amazon から書籍情報を取得する。ISBN コードがない書籍や Amazon から情報を取得できない書籍は、手入力で書籍情報を登録する。今後、他の方法で Web から取得することも考えている。また、蔵書情報を CSV ファイルから入出力できるようにすることで、同じ蔵書情報の書籍の登録を容易にしている。蔵書情報では、同じ書籍の蔵書が複数ある場合にそれらを区別するかどうかを決めることができる。そして、蔵書の所有者が異なる場合などに対応する。また、貸出禁止にすることも可能である。

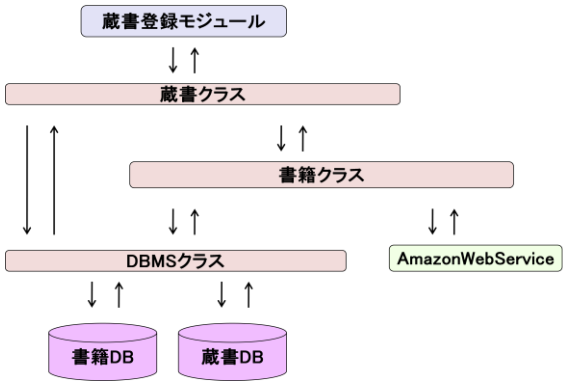


図 7 登録部のモジュール構成

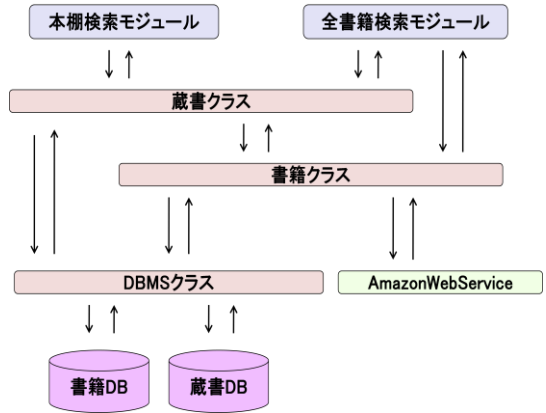


図 8 検索部のモジュール構成

3.5. 検索部

検索部では、蔵書の検索と蔵書の有無に関わらない全ての書籍検索を行う(図 8). 蔵書の検索では、本システムの蔵書 DB の中からのみ検索を行う. また、システム上で、実際の本棚を基にした、仮想的なバーチャル本棚を利用することもできる. バーチャル本棚を利用することによって、本棚の位置など曖昧に記憶していた場合などでも検

索が可能である. 蔵書の有無に関わらない検索では、蔵書 DB だけでなく Amazon API も利用する. また、一覧表示の場面でも、ユーザに最も関連のある、貸出可能かどうかや、蔵書としてあるかの情報を優先的に表示する. また、誰が寄贈したか、今まで何人が借りたのかの情報も表示する. UI は、多くのユーザが使い慣れている Amazon に近いものにする(図 9). Amazon API から取得した表紙の画像に加え、研究室で撮影した背表紙の画像も表示することで書籍を識別しやすくする. また、絶版かどうかなど蔵書を購入する際に、重要な情報も表示する.



図 9 検索部の UI 設計

表 4 ユーザの権限と根拠

		期間	購入 実施	購入 希望	書籍 登録	一括 登録	根拠
Staff	研究室教員	長期	○	○	○	○	個人蔵書を提供
Semi	ゼミ生	長期	×	○	○	○	自分の机に常駐
Club	サークル生	中期	×	○	○	○	部室を利用
Assist	授業補助学生	中期	×	×	×	○	業務に必要で貸出
Teacher	研究室外教員	短期	×	×	×	○	連絡が容易
Student	一般学生	短期	×	×	×	×	授業の指示で貸出、事前登録なし
Guest	学外関係者	長期	×	×	×	×	普段は大学に来ない、IC カードなし

4. IC カードリーダーによる認証とバーコードリーダーによる貸借

4.1. ユーザの区分と権限

ユーザを区分し、区分ごとの権限を定める。先に権限を分けて、それに応じたグループを作る(表 4)。ユーザの区分は、教員、ゼミ生、サークル生、TA、一般教員、一般学生、学外利用者に分ける。学外利用者とは、共同研究者や一部勉強会の参加者などを指す。教員は、研究室に常駐しており、購入した個人蔵書を提供したり、サークルの書籍を購入したりしている。そのため、全ての権限が与えられる。ゼミ生は、研究室に常駐していて、書籍を研究に利用するため、購入実施以外の全ての権限が与えられる。サークル生は、部費で購入した書籍も蔵書に含まれている。また、部室を利用しているため、連絡を取ることが容易である。そのため、中期の貸出期間とゼミ生同様の購入実施以外の権限を与える。TA は、事前に研究室のサーバにユーザ登録を行うため、一括登録が行える。また、TA に必要な書籍もあるため、中期の貸出期間を認める。一般教員は、連絡先が公開されているため、一括登録が可能である。一般学生は、授業で指定された参考書を借りるので、短期貸出機能のみ利用可能である。学外利用者は、頻繁に大学に来ることができないため、長期の貸出期間を認める。また、ログイン時に ID を選択可能にする場合に全てのユーザを選択可能にすると、選択肢が増えすぎて選択肢の意味が無くなってしまうため、一括登録可能なユーザのみ選択肢を用意する。IC カードによるログインは、教員も学生証も持たない学外利用者は、行えない。

表 5 ユーザ DB のスキーマ定義

部	項目	属性名	値	例
管理	ユーザ ID	user_id	integer	231
	存在フラグ	user_existflag	bool	TRUE
	有効フラグ	user_activeflag	bool	FALSE
アカウント	アカウント ID	account_id	text	mofumofu
	パスワード	account_passwd	text	8c0c3027e3
	氏名	account_urname	text	山田 太郎
	氏名読み	account_ureading	text	やまだ たろう
所属	ロール ID	role_id	integer	3
連絡先	電話番号	contact_phoneno	text	098-1233-1233

表 6 ユーザ登録時の必要事項

	一括	個別
ユーザ ID	システムが自動	システムが自動
存在フラグ	システムが自動	システムが自動
有効フラグ	システムが自動	システムが自動
アカウント ID	サーバのユーザデータ	ユーザが手入力
パスワード	ユーザが手入力	ユーザが手入力
氏名	連絡票	ユーザが手入力
氏名読み	サーバのユーザデータ	ユーザが手入力
ロール ID	サーバのユーザデータ	システムが自動
電話番号	連絡票	ユーザが手入力

4.2. ユーザ DB

ユーザ DB では、認証などに利用するユーザ情報を保存する(表 5)。システムでユーザを識別するユーザ ID を付与する。ログインするため、アカウント ID とパスワードを保存する。香川大学の教職員であるユーザのアカウント ID は、香川大学のメールアドレスを利用する。ゼミ生やサークル生など、研究室サーバに別のアカウントを持つユーザは、入力時にそちらを使うこともできる。ただし、内部的には、上記のメールアドレスで管理される。学外のユーザも、申請したメールアドレスでユーザ登録する。利用者が誰なのか分るように、氏名と読みを保存しておく。ユーザが居なくなったり、システムを利用できなくなったときには、アカウントを DB から抹消せず、存在フラグと有効フラグを用意して、無効にすることで対応する。グループ毎に権限が異なるため、所属しているグループを識別するロール ID を用意する。本を貸出したユーザに連絡を取れるようにしておくため、電話番号を保存する。

4.3. 表紙部

表紙部では、ユーザ認証とユーザ登録を行なう(図 10)。ユーザ認証の方法は、IC カードによる認証と ID とパスワードを手入力する方法である。IC カードによる認証は、セキュリティや必要な機材の問題から学内からのみ行える。ID の入力時には、あらかじめ登録されたアカウントは、一覧の中から選択することができる。ユーザ登録の方法は、2 つある(表 6)。全てのユーザ情報を手入力する方法と、サーバからデータを取得しての一括登録する方法がある。サーバからの一括登録では、アカウント ID、氏名、氏名読み、ユーザ区分、電話番号が登録される。そのため、一括登録されたユーザは、パスワード登録を行うだけでよい。一括登録は、サーバにある/etc/passwd ファイルを

利用するため、ユーザ区分は、そこから得られる。一括登録ではない場合、一般教員か、一般学生か、外部利用者かは、アカウント ID であるメールアドレスで判別できる。

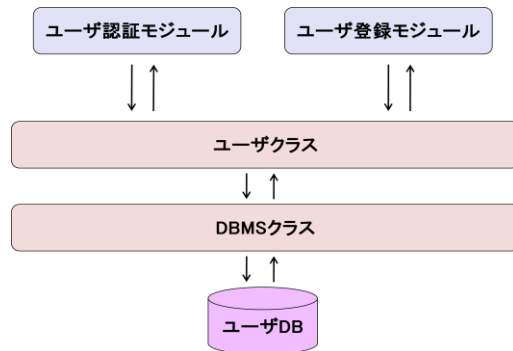


図 10 表紙部のモジュール構成

4.4. 貸出 DB と貸出部

貸出 DB では、貸出返却に必要な情報を保存する(表 7)。個々の貸出ごとに、貸出 ID を付与する。貸し出したユーザと蔵書の ID、貸出の開始と終了の日付を保存する。行為コードは、通常貸出、移動貸出、予約など、ユーザ権限に基づく貸出方法を表すコードである。貸出ステータスは、貸出中や返却済のような貸出状態を表す。

貸出部では、ユーザの権限に応じた利用を管理する(図 11)。貸出するときは、蔵書の ISBN コードをバーコードリーダーで読取り蔵書の貸出を行なう。ISBN コードのない蔵書の数は、少ないため ISBN コードのない書籍一覧を表示し選択することで、書籍を特定する。借りる蔵書と同じ書籍の蔵書が複数あり、区別が必要なときのみ、選択する。貸出の種類としては、通常の貸出の他に、指定日の予約貸出や移動の長期貸出を行なう。また、持出の禁止にも対応する。通常は、権限による貸出期間を短期と中期と長期の 3 つに分ける。一般に権限が大きいほど貸出期間を長くする。ゲストユーザに関しては、学外の利用者のため頻繁に貸出返却に來られない事を想定し 1 ヶ月としている。また、ゼミ生は、研究室の個々の席に本棚が与えられている。そのため、自分の席の本棚に借受けする、移動貸出を行えるようにする。研究室室内の本棚に借受けしている場合すぐに返却も可能である。そのため、移動貸出は、通常貸出より長期間が認められる。

表 7 貸出 DB のスキーマ定義

部	項目	属性名	値	例
管理	貸出 ID	lent_id	integer	3
	貸出ステータス	lent_status	integer	2
日付	貸出開始日	date_begin	date	2010/9/22
	貸出終了日	date_end	date	2010/9/29
ユーザ	ユーザ ID	user_id	integer	3
蔵書	蔵書 ID	biblio_id	text	1234567890
行為	行為コード	action_code	integer	3

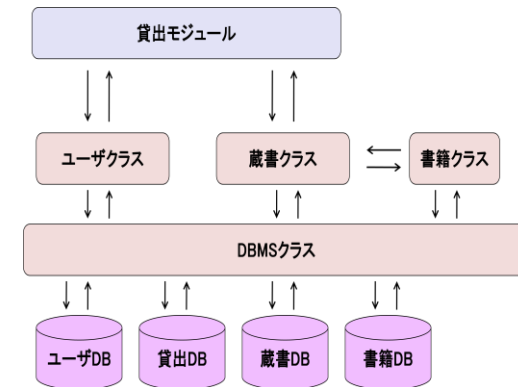


図 11 貸出部のモジュール構成

4.5. 購入部

購入部では、蔵書の購入希望申請と購入実施が行える(図 12)。購入希望申請では、蔵書の購入を新規に申請と、すでにある購入希望に投票することができる。購入希望を新規に申請するときは、希望理由の選択肢を選ぶ。追加として、個別の要望を備考に書く事もできる。既にリクエストがある場合は、賛同を投票することができる。書籍販売サイトと連携し、在庫状況なども反映される。教員やサークルの会計係がリクエストの状況を考慮して、注文する。これらの過程は、蔵書 DB にも反映される。購入実施では、Amazon API を利用して、カートに入れる購入直前の動作まで代行できる。

購入希望時に必要な理由の選択肢は、研究、授業、基礎勉強、情報技術の基礎知識、就職活動、課外活動を考えている。購入希望の書籍には、状態の新しいフィールド値として、意思と在庫がある。意思は、購入者の意思である。具体的には、承認、保留、却下がある。在庫は、オンライン購入する場合の在庫状況である。納品、注文、欠品がある。また、楽天ブックス API やヤフーオークション Web API を利用して、Amazon 以外の在庫状況の取得も検討している。

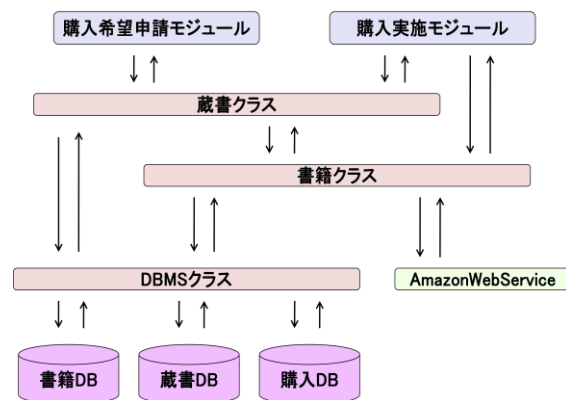


図 12 購入部のモジュール構成

5. おわりに

書籍販売サイトと連携した研究室単位の蔵書管理システムを提案した。登録作業を効率化するため、バーコードリーダーで ISBN 番号を読み取り、AWS で書籍情報を取得する。学生証を IC カードリーダーで認識して、ユーザを管理する。人気順位や利用履歴などからの検索も行える。貸出機能だけでなく、購入希望を管理する。

今後に実装する応用的な機能について述べる。記録部では、書籍へのユーザ評価や、要約、演習問題のソースコードを扱う。それらの情報を公開することで、効率的な学習を支援する。また、サークル生以上のユーザのみ閲覧可能にする。本研究室では、学習に利用した書籍の小レポートのページを作成することがある。そこで、書籍の情報を得る際に、小レポートのページの内容を自動的に取得できるようにする。それにより、本システムのために新規に要約を記入する必要がない。また、プログラム関係の書籍では、演習問題が含まれていることがある。そこで、演習問題のソースコードも、システムを利用することで共有できるようにする。また、公開時の条件も設定可

能にし、最初から全ての解答を閲覧可能にしない。他に、ユーザや書籍のキーワードに、先輩からの学習アドバイスを付加することもできる。例えば、書籍のキーワードに対して、推薦する利用順序などを示す。

分析部では、システムの利用の履歴 DB を用意し、ユーザの利用を分析する。それを基にして、貸出回数や貸出期間から人気の書籍を探すことができる。また、個人の履歴を確認することで、先行研究に携わった卒業生が利用した書籍を参考にすることができる。将来的には、論文や各種資料の管理も行い、研究室内の他のグループウェアとの連携も目指している。

文 献

- [1] Sony Felica, <http://www.sony.co.jp/Products/felica/>
- [2] 日本図書コード管理センター, <http://www.isbn-center.jp/>.
- [3] 国立国会図書館サーチの公開について,
<http://iss.ndl.go.jp/information/2010/08/releasenote/#12>.
- [4] 国立国会図書館サーチ API, <http://iss.ndl.go.jp/information/api/>.
- [5] Amazon Web サービス,
<http://www.amazon.co.jp/gp/feature.html?ie=UTF8&docId=451209>.
- [6] 関正秀, まえだひさこ, 加藤貴之, "Web API マッシュアップブック", 毎日コミュニケーションズ, (2006).
- [7] AWS SDK for Java, <http://aws.amazon.com/sdkforjava/>
- [8] Amazon EC2 API Tools,
http://aws.amazon.com/developertools/351?_encoding=UTF8&jiveRedirect=1
- [9] PEAR::Services_Amazon, http://pear.php.net/package/Services_Amazon
- [10] PHP Amazon Search <http://scripts.incutio.com/amazon/>
- [11] RDoc amazon-ecs, <http://amazon-ecs.rubyforge.org/doc/>.
- [12] RAA ruby-amazon, <http://raa.ruby-lang.org/project/ruby-amazon/>.
- [13] RubyForge ruby-aws, <http://rubyforge.org/projects/ruby-aws>.
- [14] RDoc ruby-aaws, <http://www.caliban.org/ruby/ruby-aws/>.
- [15] 川鯉光起, 富永浩之, "書籍販売サイトと連携した研究室規模の蔵書管理システム", SJCIEE 2010, p.258, (2010).