研究者向け文献 PDF データベースシステムの開発

林　坂　弘 一 郎

概要

研究者が自身で収集した学術論文の文献 PDF ファイルを効率的に管理し，研究活動に有効活用することを目的として，文献 PDF データベースシステムを開発した．利用者は PDF ファイルを Web ブラウザからサーバにアップロードすることで，PDF ファイルを一元的に管理できるようになるとともに，全文検索，ジャーナル検索，著者検索，タグ（キーワード）検索が利用できるようになる．また，論文情報の登録などに BibTeX 情報を活用することも本システムの特徴のひとつである．本論文では文献 PDF データベースシステムの詳細について議論するとともに，性能評価実験の結果を考察する．

キーワード：データベース，学術論文，全文検索，BibTeX，3 層クライアント・サーバシステム，レスポンシブ Web デザイン，Ajax

1.　まえがき

研究者が行う作業フローのひとつに学術論文からの情報収集がある．従来では学会から送付される論文誌を自身の研究室の書棚に並べたり，図書館で文献のコピーを取り整理をして保管していたりした．近年では多くの学会論文誌が PDF 形式で電子的に提供され，学会や出版社の検索機能も利用できるようになっている．これによって研究者が短時間に多くの文献を収集できるようになったという利点がある．しかしながら，学会や出版社ごとに検索サイトが異なると横断的な検索は困難になる．

一方で CiNii[*1] や Google Scholar[*2] のような Web サイトを利用することで，学会や出版社を横断的に検索することが可能となる．しかしながら現状では次のような機能が不十分である

[*1] http://ci.nii.ac.jp/
[*2] https://scholar.google.com/

と言わざるをえない．すなわち，ダウンロードした PDF ファイルに利用者自身がコメントを書き込んだりマーカーでハイライトを追加できたとしても，その PDF ファイルを Web 上に保存して追って検索するなどの有効活用ができないことである．

多くの研究者が特に自身の論文に引用するような学術論文にはコメントを書き込んだり，引用部分をマーキングしたりしている．従来は印刷物に書き込んでいたが，PDF の普及とそれを取り扱うアプリケーションの進化によって，PC だけでなくスマートフォンやタブレット端末でも，PDF 内に電子的に手書きでコメントを書き込んだり蛍光マーカーでマーキングができる環境が広まりつつある．このとき研究者が編集した文献 PDF は研究者自身が何らかの方法で保存して活用する必要がある．これには主に 3 つの方法が考えられる．一つは研究者自身の PC 内（あるいは外付けストレージ）に保存する方法であり，もう一つはクラウドストレージを利用する方法，さらに文献管理ツールを利用する方法である．

研究者自身の PC 内に保存する場合は，何らかの法則に従ってディレクトリ（フォルダ）に保存・管理することになる．従来のディレクトリによるファイル管理では様々な問題点が現れる．まず，効率的なディレクトリの構成規則やファイル名の命名規則を設計することが困難である．例えば，ジャーナル名によってディレクトリを構成しジャーナルごとにファイルを保存する場合，ジャーナル名からファイルを検索することは比較的容易にできるであろう．しかしながら，ある著者の文献を探したい，あるいは，特定のキーワードを含む文献を検索したいといった用途には適していない．さらに，PDF ファイルの増加にともなって出版年で更に細分化して保存したい，といったようにディレクトリの再構成を行うことは非常に手間のかかる作業となる．また，Windows や Mac OS にはファイル内の文字列検索機能が備わっているが，現状では検索に要する時間やその精度において必ずしも十分な性能が得られているとは言いがたい．また，研究室に設置したデスクトップ PC にデータを保存している場合，外出先や自宅からアクセスできる環境がなかったり，あったとしてもその設定には労力を伴う．

一方で，文献 PDF ファイルの管理にクラウドストレージなどのクラウドサービスを利用する方法も考えられる．Dropbox[*3] や Google Drive[*4], Evernote[*5] などを利用することで，インターネットに接続できる環境さえあれば，どこからでも自身で収集した PDF にアクセスでき，全文検索も可能となる．しかしながら，これらのサービスは広く一般的な利用目的に対して設計されたサービスであるため，例えば著者検索やジャーナル検索，出版年で絞り込む，といったように研究者が研究目的で利用するであろう機能は提供されていない．また，自身の PC 環

[*3] https://www.dropbox.com/
[*4] https://www.google.com/intl/ja/drive/
[*5] https://evernote.com/

境に保存する場合と同様に，ディレクトリ構成の問題も存在する．

さらに，文献管理ツールも様々なものが提案されている．商用ソフトウェアである EndNote [1] を利用すれば PDF ファイルを取り込み，文献情報と紐付けて管理することが可能になる．しかしながら，PDF からデータを取り込むためには PDF にデジタルオブジェクト識別子 (Digital Object Identifier: DOI) の記載が必要であり，DOI を持たない論文や未公表論文などの情報管理には適していない．MS-Word 形式で参考文献リストを出力することは可能であるが，BibTeX 情報との連携機能が不足している．RefWorks [2] も有償のソフトウェアであり，MS-Word との連携すること前提に開発されている．

また，文献管理のクラウドサービスとして CiteULike [3], Zotero [4], Mendeley [5] などが挙げられる．CiteULike では Web 上に個人の文献データベースを作成することができるが，個人の文献データベースが他者に公開されるオープンな仕様のため利用には注意が必要である．また，Zotero では BibTeX 情報が利用できるが CiteKey をキーにした取り扱いには問題が生じる．Mendeley は非常に高機能な文献管理ツールではあるが，PDF ファイル内の全文検索や著者，ジャーナル検索などができない．

Mac OS X 上では BibDesk [6] を利用することで，BibTeX 情報を一元管理でき，PDF ファイルへのリンク情報も保存できる．さらに，榎原ら [7, 8] によって BibTeX 文献管理システム「bole」が開発された．これによって BibTeX 文献情報を Web ベースで管理し，文献に対するコメントや評価なども共有できる．しかし，BibDesk や bole の主たる機能は BibTeX 情報の管理であり，全文検索などの機能は実装されていない．

このような現状に鑑み，研究者が自身で収集した文献 PDF を効率的に管理し，研究活動に有効活用できるようにすることを目的とし，文献 PDF データベースシステムを開発した．この文献データベースシステムは，データベースサーバ，Web サーバと Web ブラウザから構成される 3 層クライアント・サーバ・システムである．利用者は Web から収集した PDF ファイルや自身で作成した PDF ファイル，印刷物をスキャナによって取り込んだ PDF ファイルを Web ブラウザからインターネット経由でアップロードでき，その情報はデータベースサーバに蓄積される．蓄積された情報は全文検索だけでなく様々な方法で検索可能になる．またクライアントには特定のオペレーティングシステムを要求しないので Windows だけでなく Mac OS X や Linux などの PC 端末が利用でき，レスポンシブ Web デザインの採用によって, Android や iOS を搭載したスマートフォンやタブレット端末からもアクセスができるようになっている．さらに，論文情報の登録と出力に関しては BibTeX を利用している点も本システムの特徴である．

本論文では今回開発した文献 PDF データベースシステムの詳細について議論する．本論文

の構成は以下のとおりである．**2.**では文献 PDF データベースシステムの概要とシステム構成について述べる．**3.**では研究者自身が収集した文献 PDF ファイルを Web ブラウザからデータベースに登録し，その情報を更新する機能について述べる．**4.**では，データベースに蓄積された文献情報の参照や検索機能について述べる．**5.**では文献 PDF データベースシステムの検索性能評価に関する実験結果について考察し，**6.**で本論文をまとめる．

2.　システムの概要と構成

2.1　システムの概要

ここでは今回開発した文献 PDF データベースシステムの概要を述べる．開発したシステムの主要な特徴は次の通りである．

1. 利用者は Web ブラウザからネットワーク経由で操作できる
2. PDF ファイルのアップロードにより，データベースに登録される
3. 文献情報（論文タイトル，ジャーナル名，BibTeX 情報など）の更新ができる
4. PDF ファイルの更新ができる
5. 登録された文献情報の一覧表示ができる
6. PDF ファイル内のテキストを全文検索できる
7. ジャーナル，著者，タグなどで検索できる
8. BibTeX 情報の一覧を出力できる

図 1には本システムのトップ画面を示す．本システムを利用するためには Web ブラウザと PDF のビューアが必要となるが，これは特定のオペレーティングシステムに依存しないので，Windows，Mac OS X, Linux を搭載した PC だけでなく，Android や iOS を搭載したスマートフォンやタブレット端末からでも利用できる．さらに，システムから出力される Web ページには Bootstrap [9] を使ったレスポンシブ Web デザインを採用し，スマートフォンのような小さな画面であってもストレスなく利用できるようにした．

利用者が PDF ファイルをアップロードする際には，ファイルを一つひとつ順番にアップロードするのではなく，複数のファイルを同時にアップロードできるようにした．これには HTML　5 [10] で新たに採用された機能を利用している．さらに，アップロードされた PDF ファイル一つひとつタイトルや著者名を登録していく作業には多大な労力を必要とするが，インターネット上でも得られる BibTeX 情報を利用することで，短時間で効率的に論文情報を登

図 1　PDF データベースシステムのトップ画面

録できるようにしている．また，一旦システムにアップロードした PDF ファイルを Web サーバからダウンロードして，PC やタブレット上においてコメントやマーキングを電子的に追加した PDF ファイルを Web サーバに再度アップロードすることで，データベースの更新も可能である．

登録された文献 PDF 情報は一覧表示や検索ができる．一覧表示では登録順や更新順，閲覧順，閲覧回数順に文献 PDF 情報が表示され，その表示結果から著者名，ジャーナル名，キーワード（タグ）によって横断的に検索ができる．また全文検索機能では PDF ファイル内の全文を検索出来るだけでなく，著者，タイトル，アブストラクト，BibTeX の CiteKey などでも検索が可能である．さらに BibTeX 情報は .bib ファイルとして出力できるので，LaTeX で論文を執筆する際の参考文献リストとして利用できる．

2.2　システムの構成

次に，文献 PDF データベースシステムのシステム構成について説明する．この文献データベースシステムの構成は，図 2に示すようなデータベースサーバ，Web サーバとクライアントから構成される 3 層クライアント・サーバ・システムである．神戸学院大学ポートアイランドキャンパスのサーバ室に設置された 1 台の物理マシンには VMware ESXi 5.1 ハイパーバイザがインストールされており，ハイパーバイザ上では CentOS 7.2 がインストールされた複数の仮想マシンが動作し，それぞれデータベースサーバ，Web サーバとして機能する．なお，デー

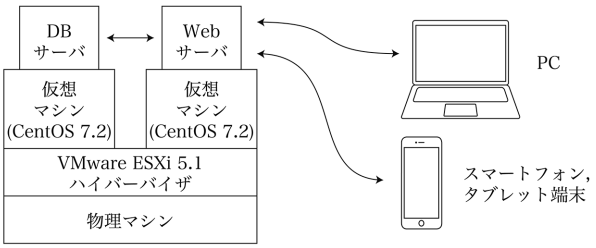

図 2　システム構成

表 1　必要なパッケージとバージョン

パッケージ	バージョン
DB サーバ	
MySQL	5.7.12
Mroonga	6.0.2
Web サーバ	
Apache	2.4.6
perl	5.16
perl-DBI	1.627
perl-DBD-MySQL	4.023
ImageMagick	6.7.8.9

タベースサーバと Web サーバを別の物理マシン上でそれぞれ稼働させたり，単一 OS 上で両サーバの機能を同時に稼働させるような運用方法も可能である．今回は計算機資源の有効利用と保守性向上の観点から図 2に示すような仮想化技術を利用したシステム構成を採用した．

データベースサーバおよび Web サーバにインストールしたアプリケーション，ミドルウェア等のパッケージとそのバージョンを表 1に示す．MySQL [11] は Oracle 社が開発・配布を行っているオープンソースのデータベース管理システムであり，Yahoo!, Facebook, Twitter などのウェブサイトでも利用されている実績がある．Mroonga [12] は全文検索エンジンである Groonga [13] をベースとした MySQL のストレージエンジンである．Mroonga ストレージエンジンを利用することで，MySQL 上で高速な日本語全文検索が利用できるようになる．

Web サーバのサーバソフトウェアには Apache [14] を利用し，アプリケーション開発言語には主に perl を，補助的に JavaScript も使用している．また Web サーバからデータベースサーバへ接続するために perl-DBI と perl-DBD-MySQL を利用している．ImageMagick [15]

はアップロードされた PDF から表紙の JPEG サムネイル画像を生成するために必要である．

一方で，クライアントは Web ブラウザを必要とする．インターフェースとなる Web ページは HTML 5 [10] や CSS3 [16] で新たに追加された機能を積極的に利用していることから，PC 用の Web ブラウザの場合，Internet Explorer 11, Edge 13, Firefox 21, Google Chrome 26, Safari 6.1, Opera 15 以降が必要である．それ以前のバージョンのブラウザでは一部の機能が正常に動作しない．また，スマートフォン，タブレット端末用の Web ブラウザの場合，文献の一覧表示，検索機能の利用に関しては iOS Safari 7.1, Android Browser 4.4, Chrome for Android 50 以降が必要であり，Opera Mini では最新の 8 でも対応しない．複数 PDF ファイルのアップロード機能に関しては最新の Android Browser 50, Chrome for Android 50 でも対応していない．

2.3　データベースの設計

図 3には PDF データベースの設計図である ER 図を示す．図 3の中央に配置した paper テーブルの 1 レコードがひとつの文献 PDF に割り当てられる．この paper テーブルには doc_id を主キーとして，タイトル (title)，サブタイトル (subtitle)，サブタイトル 2 (subtitle2)，巻 (volume)，号 (num)，ページ番号 (startpage, endpage)，出版年 (year) と外部キーとしてジャーナル ID (journal_id) が記録される．ここでサブタイトル 2 は後述する詳細検索で利用されるが，日本語論文の英文タイトルなどを記録することができる．また， paper テーブルは journal テーブルと多対 1 の対応関係になっている．

図 3上の author テーブルや keyword テーブルは paper テーブルと多対 1 の対応関係である．著者情報に関する author テーブルでは，author_order 属性に著者のその論文における執筆者順位を記録する．著者の氏名は姓 (lname)，名 (fname) の他に，属性 (lnamesort) に姓の読みを入力することができる．この属性には日本人著者の読みをアルファベットで入力することで，検索や並び替え時に有効に機能する．各論文のキーワード（タグ）は keyword テーブルに記録することができる．なお，author テーブルについては，データベースの冗長性を排除するという観点から paper テーブルと多対多の対応関係にすべきである．しかしながら，あらゆる同姓同名の著者を所属機関等の情報によって整理・分類して冗長性のない author テーブルを構築するには相当の労力が必要となることから，本システムではデータ入力の効率性に主眼を置き，author テーブル内のデータの冗長性を許容した設計を採用した．

図 3左の abstract, doctext, note, pdf テーブルはぞれぞれ，アブストラクト，PDF ファイルから取り出されたテキスト全文，ユーザによるメモ，PDF のファイル名を記録するテーブ

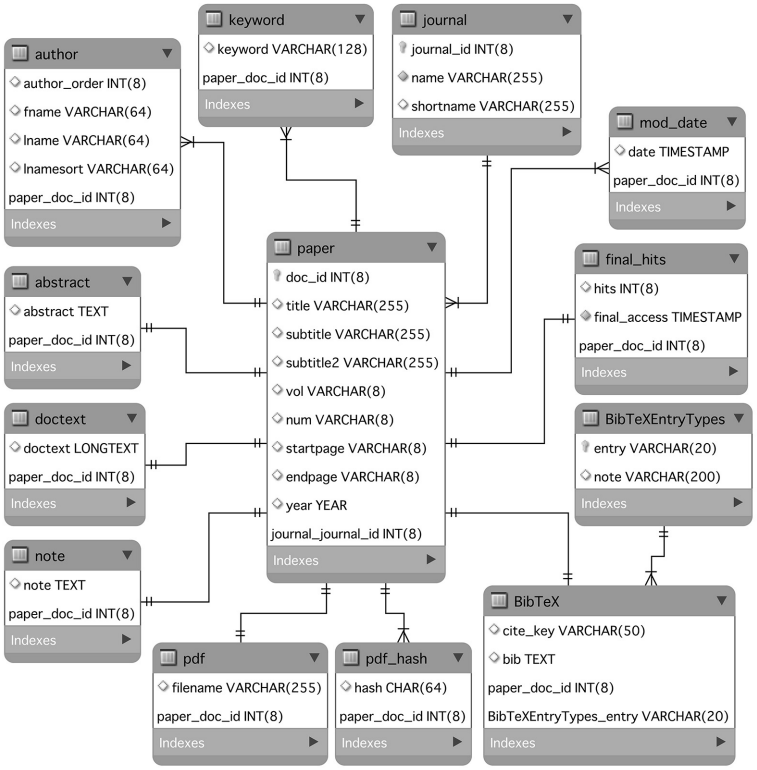

図 3　データベースのテーブル構造

ルである．これらはすべて paper テーブルと 1 対 1 の対応関係である．このため，これら 4 つのテーブルの属性は paper テーブル内に取り入れることも可能である．しかしながら，特に doctext はテキスト全文を保存することからそのサイズが将来的に大きくなる可能性を考慮して，paper テーブルとは分離している．PDF ファイルのフィンガープリントとして SHA-256 によるハッシュを pdf_hash テーブルに保存する．この pdf_hash テーブルは paper テーブルと多対 1 の対応関係である．つまり，PDF ファイルが更新されるたびに pdf_hash テーブルにハッシュが追加されることにより，過去の PDF ファイルのハッシュが履歴としてすべて蓄積されるので，同じ文献 PDF ファイルが誤って複数の paper レコードとして重複登録されることを防ぐことができる．

図 3右下の BibTeXEntryTypes テーブルには，article, book, inproceedings など BibTeX の仕様として定義されている 14 種類のエントリ種別が登録されている．BibTeX テーブルには

図 4　BibTeX 情報の例

文献の BibTeX 情報を格納する．ここで cite_key は文献を一意に識別できる引用キーであるため，データベースにおいて Unique 制約が設定されている．例えば図 4に示した BibTeX 情報の場合，1 行目の article が entry 属性に，Rinsaka_2014KGU が cite_key 属性に登録される．2 行目以降の情報は bib 属性に登録される．

図 3右上の mod_date テーブルには，文献 PDF 情報の登録日時と更新日時が記録され，final_hits にはその情報への累積アクセス回数と最終アクセス日時が記録される．これらのテーブルを利用することで，文献一覧表示の際に，登録順だけでなく，最終更新順，最終閲覧順，閲覧回数順に並べ替えを行うことが可能になる．

更に，paper テーブルの title 属性や abstract テーブルの abstract 属性，doctext テーブルの doctext 属性など，文字列を登録する属性の多くには Mroonga のインデックスを設定している．ここでインデックスには形態素解析のひとつである MeCab [17] トークナイザを使用している．これによって高速な日本語全文検索が実現できる．

3.　文献 PDF の登録と編集

ここでは PDF ファイルを文献 PDF データベースに登録する機能について詳述する．

図 5　ジャーナルデータの登録

3.1　ジャーナル情報の登録

PDF ファイルはどのジャーナルの論文であるかを指定してアップロードする必要があるため，予めジャーナル情報を登録しておくことが推奨される．ジャーナル情報の登録は図 5に示す画面から行うことができる．ジャーナルタイトルと必要に応じて省略表記を入力して登録ボタンをクリックすれば，自動的に journal_id が与えられ，データベースの journal テーブルにレコードが登録される．

3.2　文献 PDF の登録

文献 PDF ファイルをアップロードして PDF データベースに登録する機能について説明する．図 6は PDF ファイルをアップロードするための画面である．本システムでは一つひとつ PDF ファイルを選択してアップロードすることも可能であるが，最大 100 個までの PDF ファイルを同時にアップロードして登録することが可能である．これは HTML 5 で新たに採用された <input type=”file”> タグの multiple 属性を使用することで実現している．また PDF ファイルのドラッグ&ドロップエリアを CSS によって拡大し，操作性の向上を図っている．セキュリティを考慮して，ファイルの拡張子が PDF ファイル以外のファイルは Web ブラウザがアップロードを受け付けず，拡張子を偽装したファイルは Web サーバでフィルタリングする仕様になっている．ただし，複数ファイルの同時アップロードは一部のスマートフォン，タブレット端末用最新 Web ブラウザ (Android Browser 50, Chrome Android 50) でも対応していない．

図 6のアップロード画面において，PDF ファイル，ジャーナル，出版年の必須項目，および

図 6　文献 PDF のアップロード画面

必要に応じて複数の PDF ファイルで共通するタイトル，キーワードや巻，号を入力し，「アップロードして登録」ボタンをクリックすれば，PDF ファイルが Web サーバの一時作業ディレクトリにアップロードされ，データベースへの登録処理が行われる．なお，キーワードは，半角のカンマ，セミコロン，コロンまたは，全角のカンマ，読点，セミコロン，コロンで区切れば複数指定することができる．さらに，ここで入力された情報は JavaScript によって Web ブラウザの WebStorage に記録される．これによって次にアップロード画面を開いた時には前回の登録内容と同じデータが初期状態として入力されるため，多数あるジャーナルの選択コントロールから毎回目的のジャーナルを探す手間を省略することができる．

データベースへの登録処理手順は次のとおりである．まず，Web サーバにアップロードされた PDF ファイルは受付日時に応じた 12 桁の数字，および同時にアップロードされたファイルを識別するための 2 桁の数字からなる 14 桁のファイル名に変更される．その後，PDF ファイルのハッシュ値を SHA-256 アルゴリズムによって取得する．取得されたハッシュ値はデータベースの pdf_hash テーブルに問い合わせが行われ，重複確認が実施される．重複が認められれば，同じファイルがすでにデータベースに登録されていると判断し，アップロードされた PDF ファイルは Web サーバから削除される．一方で重複がなかった場合には一つひとつの PDF ファイルに対して次の処理が実行される．

まず，paper テーブルにジャーナル，出版年などともに，サブタイトルにはアップロードさ

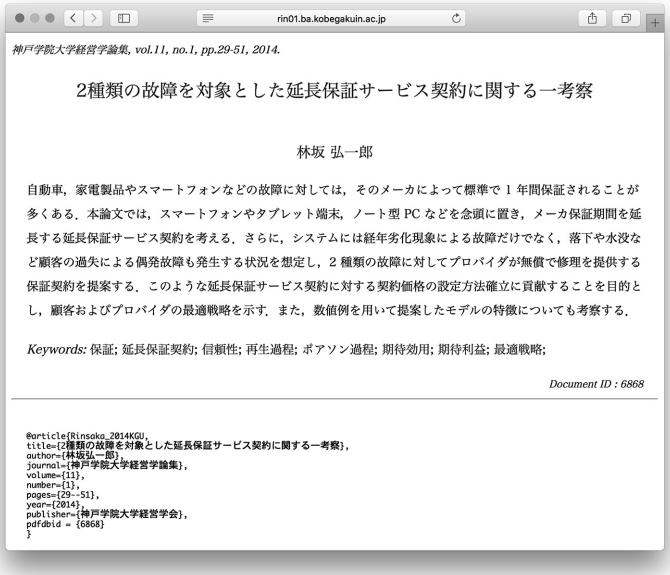

図 7　生成された HTML ページ

れたファイル名が登録され，これによって PDF の doc_id が設定される．区切り文字によって分割されたキーワードは keyword テーブルに登録され，web サーバによって変更された PDF ファイル名は pdf テーブルに登録される．PDF ファイルからは透明テキスト情報が取り出され，doctext テーブルに登録される．したがって，論文タイトルなど詳細情報を登録しなくとも，PDF のテキストデータから全文検索が可能になる．次に mod_date と final_hits テーブルには登録日時が，pdf_hash にはハッシュ値が登録される．

さらに, PDF から取り出した透明テキスト情報から, PDF ファイルと同じ基底名 (basename) を持つ静的な HTML ファイルを生成する．これは，PDF ファイルの閲覧が困難なスマートフォンなどでも全文容易に確認するために利用でき，後述のとおり著者名や論文タイトル，アブストラクトなど追加して論文情報を更新すれば図 7のような HTML ページになる．また，ImageMagick によって PDF ファイルの 1 ページ目から 2 種類の解像度の JPEG ファイルを生成する．一時作業ディレクトリの PDF ファイルは PDF 保存用のディレクトリに移動される．

アップロードされたすべての PDF に対して上記の登録作業が完了すれば，図 8に示す画面が表示される．

図 8　文献 PDF のアップロード完了画面

3.3　文献情報の編集と更新

アップロードされた PDF ファイルはすでにそのテキスト情報が doctext テーブルに登録されているので，全文検索が可能になっている．しかしながら，論文のタイトルや著者情報，アブストラクトなどを登録することでより詳細な検索が可能になる．文献情報の編集は図 8のアップロード完了画面や後述する論文の一覧，検索結果画面の「文献情報の編集・更新」リンクから辿ることのできる図 9の画面から可能である．図 9の「著者を増やす」「著者を減らす」ボタンによって，著者数を増減できる．著者名やタイトルなど詳細情報を入力し，更新ボタンを押下すれば paper テーブルなど各テーブルが更新され，mod_date テーブルには更新日時が追加される．しかしながら，論文情報を一つひとつコントロールに入力するには非常に多くの手間を要することになる．

簡単に論文情報を入力するには，「BibTeX の追加」ボタンを押下して表示される図 4の画面に BibTeX 情報を入力すれば良い．この BibTeX 情報は CiNii や Google Scholar などの Web サービスから取得することができる．図 4のように BibTeX 情報を入力して「BibTeX から論文情報を生成」ボタンをクリックすれば，図 10 のように著者名，タイトルなどが BibTeX 情報から抽出され，各コントロールに設定される．必要に応じてアブストラクト，キーワードなども入力して論文情報を更新することが可能になる．なお，図 10の CiteKey コントロールの値

図 9　文献情報の編集画面

が変更されるたびにその重複を確認するための問い合わせが Ajax の非同期通信によって Web サーバを経由してデータベースサーバに送信される．CiteKey の重複が確認された場合は，エラーメッセージが表示されるとともに更新ボタンが無効化される．

図 10のように BibTeX 情報を入力した場合には BibTeX テーブルにその情報が登録される．このとき，PDF ファイルや HTML ファイルのファイル名は 14 桁の数字から CiteKey と同じファイル名に変更される．これにより，ローカル環境に PDF ファイルをダウンロードして作業する際に目的の PDF ファイルを容易に検索することができる．

また，図 10の編集画面では，ローカル環境で PDF ファイルにメモを追加したりマーキングを行った PDF ファイルを Web サーバにアップロードすることで PDF ファイルを更新することもできる．PDF ファイルがアップロードされた時には，ハッシュ値が計算され pdf_hash テーブルに追加される．これにより PDF ファイルのハッシュ値の履歴がすべて残るので，PDF ファイルに書き込んだ後であっても，オリジナルの PDF ファイルが後にアップロードされて同じ文献が重複登録されることを防止できる．

備考欄には論文に対するコメントや評価などを自由に入力することができる．ここに入力された文字列に関しても後述する全文検索機能を使って検索が可能になる．

論文情報の更新と同時に PDF ファイルのテキスト情報から生成された静的 HTML ページも更新される．タイトル，著者名，アブストラクト，BibTeX 情報などの登録後に生成された HTML ページを図 7に示す．図 7の HTML ページには PDF ファイルから抽出された本文も含まれており，ページのリンクから PDF ファイルを開いたり，著者，ジャーナル，キーワードの検索結果を表示することも可能である．

さらに，登録された BibTeX 情報の一覧は .bib ファイルとしてダウンロードすることができ

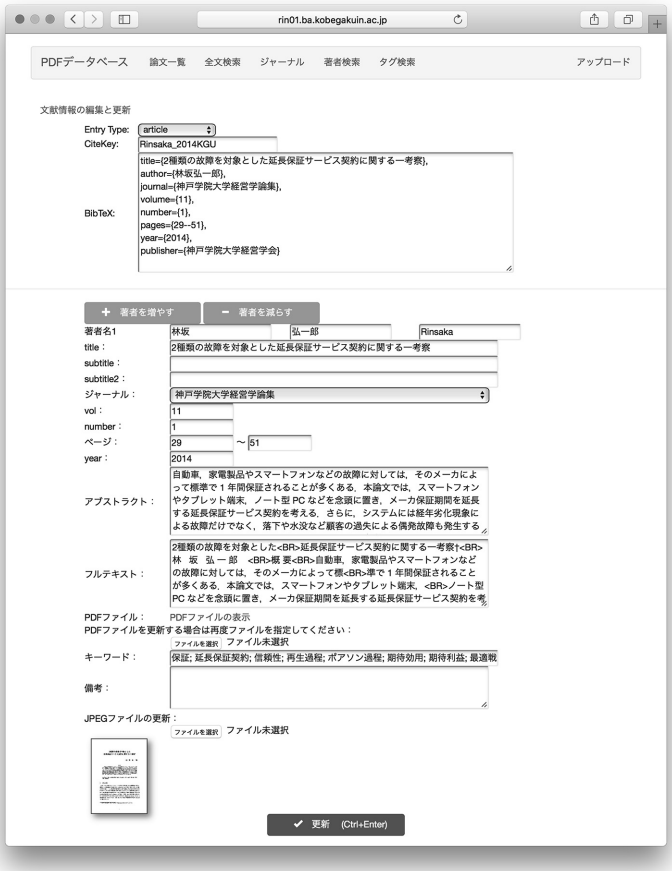

図 10　BibTeX 情報が追加された文献情報の編集画面

る．このファイルは LaTeX 文書作成時の参考文献リストとしてそのまま利用できる．

4.　文献 PDF 情報の参照と検索機能

ここでは登録された文献 PDF 情報の一覧表示と検索機能について説明する．なお本システムには 2 種類に大別される検索機能が実装されている．すなわち全文検索機能とジャーナル名，著者名，タグの Ajax による非同期通信検索機能である．

図 11　文献情報一覧表示

4.1　文献 PDF 情報の参照

図 11には PC で登録論文を一覧表示した画面を示す．本システムは Bootstrap によるレスポンシブ Web デザインを採用しているため，スマートフォンで同じ一覧を表示した際には図 12のようにレイアウトが変更される．レスポンシブ Web デザインでは Web サーバで Web ブラウザの種類を判別するのではなく，Web サーバはどのような種類の Web ブラウザにも同じ HTML 文書を生成して送信する．HTML 文書内にはブラウザのウインドウサイズによって要素の配置を変更するためのコードが記述されているので，Web ブラウザはその記述に従って要素を配置する．例えば，メーニューのリンクは PC ではページ上部にそれぞれ配置されているが，スマートフォンで表示した際には，右上のハンバーガーボタンに変化する．また，論文一覧のレイアウトもスマートフォンに適した配置に変更される．

図 11や図 12の論文一覧では，その結果表示の順序は「最終登録順」「最終更新順」「最終閲覧順」「閲覧回数順」から選択することができる．PDF ファイル 1 ページ目のサムネイル画像をクリックすれば，高解像度の JPEG ファイルで 1 ページ目のイメージを確認できる．論文のタイトルをクリックすれば，図 7に示した HTML 文書を表示することができる．また，著者のリ

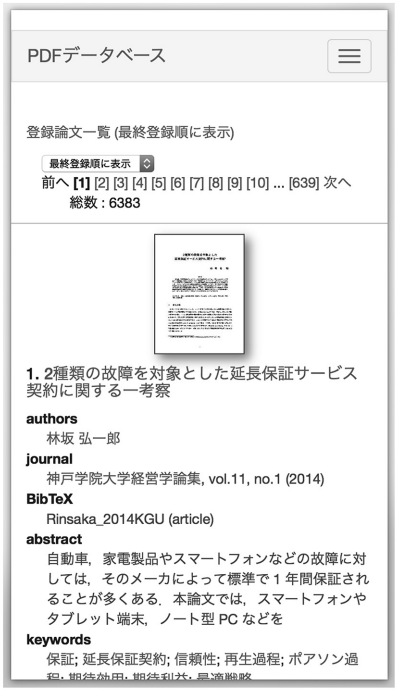

図 12　スマートフォンでの文献情報一覧表示

ンクをクリックすれば，その著者の文献を一覧で表示できる．ジャーナル名をクリックすればそのジャーナルに登録されているすべての文献を一覧で取得できる．キーワードも同様に同じキーワードが登録されている文献を一覧で表示できる．さらに，PDF ファイルの表示や図 9，図 10の文献情報の編集・更新画面へのリンクも表示される．なお，HTML ページ，PDF ファイル，文献情報の編集・更新リンクをクリックすれば，指定ページが表示されるとともに，データベースで final_hits テーブルの累積アクセス回数と最終アクセス日時情報が更新される．

4.2　全文検索機能

全文検索機能はさらに標準検索機能と詳細検索機能に分けられる．標準検索機能では，利用者が入力したキーワードを PDF ファイル内の透明テキスト情報が保存された doctext テーブルから検索し，Mroonga の検索スコア順で 1 ページにつき 10 件ずつ結果が表示される．複数のキーワードをスペースで区切って検索することで絞り込み検索も可能である．

図 13は詳細検索の結果表示画面である．図 13において「検索ワード」は PDF ファイル内の

図 13　全文検索（詳細検索）結果表示

透明テキストを検索するための項目である．著者，タイトル，アブストラクトなどの各項目にキーワードを入力することで，検索結果を絞り込むことができる．検索結果には検索キーワード周辺のテキスト（スニペット）も出力される．また検索結果の表示順も「検索スコア順」だけでなく「出版年順」「最終登録順」から選択することができる．さらに，標準検索では結果は 1 ページに 10 件ずつ表示されるが，詳細検索では 1 ページあたり 1 件から 100 件の範囲で自由に変更することも可能である．また，特に BibTeX の CiteKey や備考欄に入力したコメントからも文献情報を検索できる機能は本システムの特徴である．これらの機能を活用することで論文執筆時の文献引用作業を効率的に行うことが可能になる．

4.3　非同期通信検索機能

全文検索機能では利用者が検索ボタンを押下してはじめて Web サーバに検索のリクエストが送信され, 画面遷移によって検索結果が表示される．本システムのジャーナル検索，著者検索，およびタグ検索では Ajax による非同期通信検索を採用することで，ユーザがキーを入力するたびにバックグラウンドで検索リクエストが送信され，画面の遷移を伴うことなくその結

果が同じ画面に表示されることになる．

図 14〜16にはそれぞれジャーナル検索，著者検索，タグ検索の結果を示す．ジャーナル検索の初期状態ではデータベースに登録されているすべてのジャーナル一覧と各ジャーナルに登録されている文献数が表示される．例えば 450 件のジャーナルが登録されている時，検索キーワードとしてコントロールに「reliability」と入力すれば，ユーザがキーボードから 1 文字入力する毎に合計 11 回の検索リクエストが Web サーバに送信される．Web ブラウザは画面の遷移を伴うことなく Web サーバからの返答を受け取った時点でページ内に表示する．「reliability」と入力した時点で検索結果は 17 件まで絞りこまれ，「reliability ieee」と入力した時点では 5 件まで絞り込まれる．さらに「reliability ieee tr」まで入力した時点で図 14のとおり検索結果は 1 件に絞り込まれる．この機能によって利用者は簡単に目的のジャーナルを検索することが可能になる．

図 15に示す著者検索も同様に Ajax による非同期通信を利用している．例えば著者名に「r」と指定した時点で結果は 408 件，「ri」では 56 件，「rin」まで入力した時点で 3 件まで絞り込まれる．さらに「rins」まで指定した時点で図 15のとおり 2 件まで絞り込まれる．ここで，著者検索では author テーブルで著者の姓を登録する lname 属性だけでなく，読みをアルファベットで登録する lnamesort 属性についても検索の対象としている．したがって，アルファベットで検索をすればその読み情報から漢字の著者名も検索することが可能になっている．

図 16に示すタグ検索では keyword テーブルに登録された各文献のタグ（キーワード）情報を検索することができる．例えば「warranty cost」と入力することで，関連する文献を検索することが可能になる．

5.　性能評価実験

ここではデータベースサーバの性能評価と文献 PDF データベースシステム全体の性能評価実験について説明し，その結果を考察する．表 2に示す通り，データベースサーバ，Web サーバは神戸学院大学ポートアイランドキャンパスのサーバ室に設置された一台の物理マシン上の仮想マシンとしてそれぞれ動作している．

まず，データベースサーバの性能評価実験について述べる．文献 PDF データベースシステムでは，クライアントからの検索要求は Web サーバに送信され，Web サーバ内の perl プログラムが SQL 文を生成し，データベースサーバに問い合わせを発行する．ここではデータベースサーバに直接ログオンし，Web サーバが生成する SQL 文と同じ SQL 文 100 種類を連続実行することでその応答性能を検証する．

図 14　ジャーナル検索

図 15　著者検索

図 16　タグ（キーワード）検索

表 2　実験環境

物理マシン		
CPU	Intel Xeon E5-1410 2.8GHz 4 コア / 8 スレッド	
メモリ	20GByte	
ストレージ	15000rpm SAS 6Gbps 300GByte × 4 台 RAID5 構成	
ハイパーバイザ	VMware ESXi 5.1	
仮想マシン	DB サーバ	Web サーバ
CPU	4 コア	4 コア
メモリ	4GByte	4GByte
ストレージ	100GByte	100GByte
OS	CentOS 7.2	CentOS 7.2

表 3　SQL 連続実行処理性能

	単位	最良値	最悪値	平均値	中央値	標準偏差
標準検索（英）	秒	0.775	1.327	0.811	0.796	0.061
標準検索（日）	秒	0.508	0.585	0.519	0.516	0.014
詳細検索（英）	秒	107.381	119.287	110.445	110.212	1.875
詳細検索（日）	秒	14.499	22.008	20.119	20.581	1.834

具体的には，標準検索と詳細検索をそれぞれ英語と日本語のキーワードで以下のとおり行った．標準検索では「data」「estimation」「software」など 100 種類の英語検索ワードと「データ」「推定」「ソフトウェア」などの 100 種類の日本語検索ワードを用意し，それぞれ 100 種類の SQL 文を生成した．英語と日本語の詳細検索ではそれぞれ 10 種類の検索ワードと 10 種類の著者名を用意し，全文検索とアブストラクト検索，および著者検索で組み合わせた 100 種類の SQL 文をそれぞれ生成した．これらの 100 種類の SQL 文をデータベースサーバ上で連続実行しその処理時間を計測する実験を行った．さらにこの実験を 100 回ずつ繰り返した結果を表 3に示す．なお paper テーブルの PDF 登録件数は 6383 件，author テーブルの登録件数は 8773 件である．

表 3より，キーワードが 1 個の標準検索の処理時間は 1 回の検索につき平均 0.01 秒未満であり，十分な検索性能を有していることがわかる．また，詳細検索は英語で 1.1 秒程度，日本語で 0.2 秒程度である．詳細検索では検索時にテーブルの内部結合を実行していることから，標

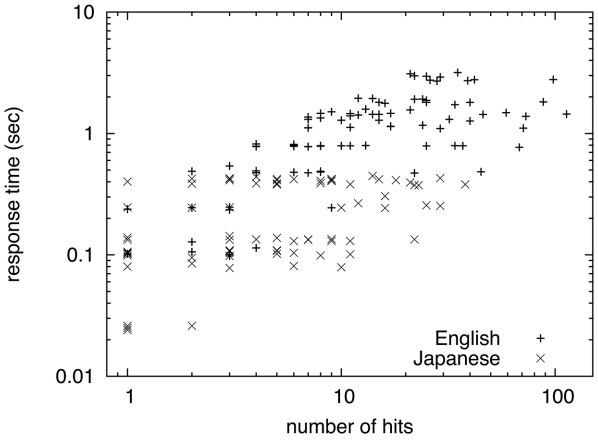

図 17　データベースサーバの詳細検索レスポンスタイム

準検索と比較して処理に時間を要していると考えられる．また，標準検索，詳細検索ともに日本語検索が英語検索と比較してより高速に検索処理を実行できることがわかる．詳細検索における検索ヒット数と SQL 1 個あたりの検索レスポンスタイムの関係を図 17に示す．図 17から検索ヒット数と検索レスポンスタイムには正の相関を見て取れる．なお，英語検索の相関係数は 0.483，日本語検索の相関係数は 0.462 である．さらに図 17より，日本語検索では検索ヒット件数にかかわらずすべての点が 0.45 秒以下に分布しているのに対して，英語検索では 4 以上の検索ヒット件数の場合にほぼすべての点が 0.45 秒以上に分布していることが読み取れる．つまり検索ヒット数が同じ場合であっても，日本語検索は英語検索と比較してレスポンスタイムが短くなる傾向がある．

次にクライアントの Web ブラウザから PDF データベースに検索を行う処理についての性能評価を行った．実験環境は次のとおりである．クライアント PC はサーバ物理マシンと同じサーバ室に設置され，1Gbps の L2 スイッチを経由して接続されている．クライアントのオペレーティングシステムは CentOS 7.2 であり，Web ブラウザは Google Chrome 50.0.2661.102 (64bit) である．各検索実験について Web ブラウザにて検索処理を開始してから検索結果ページの全要素の描画が終了するまでのターンアラウンドタイムを Google Chrome のデベロッパーツールを利用して計測し，これを 100 回ずつ繰り返すことで表 4の結果を得た．

表 4より，全文検索では標準検索，詳細検索ともに概ね 1〜2 秒程度で結果が表示されることがわかる．また，一覧表示やジャーナル，著者検索，タグ検索後の一覧表示ページは 500 ミリ秒程度で表示が終了する．したがって，Web ブラウザを含めたシステム全体についても十分な

表 4　Web ブラウザからの検索処理性能

	単位	最良値	最悪値	平均値	中央値	標準偏差
標準検索（英）	秒	1.31	3.35	1.91	1.83	0.42
標準検索（日）	秒	1.12	1.53	1.22	1.20	0.07
詳細検索（英）	秒	0.19	4.12	2.09	2.24	1.09
詳細検索（日）	秒	0.39	1.74	1.30	1.31	0.29
一覧表示	ミリ秒	434	665	471	453	40.51
ジャーナル検索	ミリ秒	251	830	562	571	85.63
著者検索	ミリ秒	358	623	472	466	39.47
タグ検索	ミリ秒	327	776	477	465	62.61

処理性能を持つことが確認された．

6.　むすび

本論文では研究者が行う研究フローの一部を支援する目的で開発した文献 PDF データベースシステムについて述べた．本システムを利用することで Web からダウンロードした文献の PDF や，印刷物をスキャナによって取り込んだ PDF を一元管理でき，さらには全文検索機能やジャーナル検索，著者検索，タグ検索などを利用して短時間で効率的に文献を検索できるようになった．

なお，Web からダウンロードした PDF ファイルは二次配布が著作権によって制限されていることが少なくないため，本システムは研究者が個人で収集した PDF ファイルを個人的に利用することを想定して開発した．今後の課題のひとつとして，本システムにマルチユーザ利用機能を追加し SNS のようなユーザ間での情報共有を可能にすることが考えられるが，この際には著作権の保護を考慮した設計を行うことが肝要になる．さらに，BibTeX だけでなく，MS-Word 形式で参考文献リストを作成したり，論文情報の取得・管理にデジタルオブジェクト識別子などを活用することも今後の課題として考えられる．

参考文献

[1] EndNote, http://www.usaco.co.jp/products/isi_rs/endnote.html (2016 年 6 月 8 日確認).

[2] RefWorks, https://www.refworks.com/ (2016 年 6 月 8 日確認).

[3] CiteULike, http://www.citeulike.org/ (2016 年 6 月 8 日確認).

[4] Zotero, https://www.zotero.org/ (2016 年 6 月 8 日確認).

[5] Mendeley, https://www.mendeley.com/ (2016 年 6 月 8 日確認).

[6] BibDesk, http://bibdesk.sourceforge.net/ (2016 年 6 月 8 日確認).

[7] 榎原博之, 大塚隆弘, 山上悠喜：研究室向け BibTeX 文献管理システム, 情報処理学会誌, Vol. 53, No. 8, pp. 2049–2060 (2012).

[8] 榎原博之, 大塚隆弘, 宮川朋也：BibTeX 文献管理システムに対する有用性の評価, 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2013, No. 5, pp. 1–6 (2013).

[9] Bootstrap, http://getbootstrap.com/ (2016 年 6 月 8 日確認).

[10] HTML5, https://www.w3.org/TR/html5/ (2016 年 6 月 8 日確認).

[11] MySQL, http://www.mysql.com/ (2016 年 6 月 8 日確認).

[12] Mroonga, http://mroonga.org/ (2016 年 6 月 8 日確認).

[13] Groonga, http://groonga.org/ (2016 年 6 月 8 日確認).

[14] Apache, https://httpd.apache.org/ (2016 年 6 月 8 日確認).

[15] ImageMagick, http://imagemagick.org/ (2016 年 6 月 8 日確認).

[16] CSS, https://www.w3.org/TR/CSS/ (2016 年 6 月 8 日確認).

[17] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis, in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vol. 4, pp. 230–237 (2004).