

論文集の出版における OSS の活用

細川 康輝・池本 有里・鈴木 直美
池本 未希・山本 耕司

Applying Open Source Software to Publishing of Proceedings

Yasuteru HOSOKAWA, Yuri IKEMOTO, Naomi SUZUKI
Miki IKEMOTO and Kohji YAMAMOTO

ABSTRACT

Recently, many kinds of documents are digitized. These digitized documents are read on tablet computers, smartphones and so on. One of the advantages is the weight. In the case of taking many books, the weight is very heavy. However, digitized books can be taken as smartphones. This advantage is effective for many kinds of proceedings. In this paper, the method of publishing some kind of proceedings is shown. Open source software are used in this method.

キーワード：出版, OSS, PDF, L^AT_EX

1 はじめに

近年、インターネット環境の一般化と高速化が進み、さまざまなネットワーク上のサービスやそれを利用するための高性能なデバイスが一般的なものとして利用されつつある。例えば、facebook [1] などの SNS や amazon [2] などのオンライン通販サイトなどのサービスが、パソコンからだけでなく、タブレット端末、スマートフォンなどで利用されつつある。書籍に関しても、汎用あるいは専用のタブレット端末が安価に販売され、端末上で書籍の購入、閲覧ができるようになってきている。このメリットは、著作者、販売者、読者それぞれに考えられるが、読者においては、いつでもどこでも購入、閲覧が可能で、紛失、汚損がな

く、何百冊分でも常に持ち運ぶことも容易であるなどのメリットが考えられる。

各種学会などにおいても、論文誌や研究会予稿集の電子化は進められており、現在では数多くの論文を Web を通じて閲覧することが可能である。国際会議などの論文集の場合、多くの著者によって作成された論文を取りまとめ、さらに査読の処理やページ番号、ヘッダなどの加工が必要である。これらの作業は非常に煩雑で、時間のかかるものであり、業者に依頼した場合、高額となる。

徳島大学の 上田教授は、OSS な運用システムや独自のプログラムを組み合わせ、国際会議などで必要とされる論文に関する処理の多くを短時間かつ安価に行える方法を確立した [3]。そして、数多くの国際会議などで運用し、それらの成果は

2013年1月7日受付, 2013年3月8日最終受付
細川康輝 四国大学経営情報学部
Yasuteru HOSOKAWA, Member (Faculty of Management and Information Science, Shikoku Univ., Tokushima, 771-1192 Japan).
池本有里 四国大学経営情報学部
Yuri IKEMOTO, Member (Faculty of Management and Information Science, Shikoku Univ., Tokushima, 771-1192 Japan).
鈴木直美 四国大学経営情報学部
Naomi SUZUKI, Member (Faculty of Management and Information Science, Shikoku Univ., Tokushima, 771-1192 Japan).

池本未希 NPO 法人 AUX
Miki IKEMOTO, Nonmember

山本耕司 四国大学経営情報学部
Kohji YAMAMOTO, Member (Faculty of Management and Information Science, Shikoku Univ., Tokushima, 771-1192 Japan).

四国大学経営情報研究所年報 No. 18 pp.1-9 2013年2月

Webで公開されている。この公開された情報を利用すれば、効率的な学会運営だけでなく、多数のPDFファイルを取りまとめる必要のあるさまざまな場合においても、効率的かつ安価にそれらを実施できる。

本論文では、公開された情報を基に実際に学会運営などを行った時の体験で得られた知見、行った改良、問題点などを考察し、このような場合のOSSの意義や利点について述べる。

2 IAPR Commence Conference Management System

IAPR Commence Conference Management System [4] (以下CCMSとする)は、学会、研究会などを運営するためのPHPで書かれたWebアプリケーションである。2003年よりクイーンズランド大学 Brian C. Lovell 教授をプロジェクトリーダーとする複数の大学のメンバーを含む開発チームによって開発されている。このシステムはオンライン上で、投稿者のユーザ登録から、投稿、著者情報の登録ができ、査読者への査読依頼、査読結果の集計、採否通知などの査読処理もできる。投稿者、査読者などに一括してメールするなどの機能も実装されており、さらに、参加費の処理や名札、投稿されたPDFファイルの

チェック機能の拡張などが行われている。このように国際会議、研究会など運営において必要な処理が一通り含まれており、運営者の負担を大幅に軽減できるものである。しかしながら、2008年のバージョン1-3c以降更新されておらず、また、現状でいくつかのバグ、不具合などがあるため、実際の運用では、ソースの改変が必要である。本システムがオープンソースのライセンスである Mozilla Public License 1.1 (MPL1.1) [5] で開発されていたためソースコードの入手、バグの修正も可能であり、さらには運営する学会のニーズによって、適した改変さえも可能である。

3 国際会議での出版システム

本章では、実際にCCMSを用いて学会運営を行ったときのシステム運用について述べ、その問題点と行った対応についても紹介する。運用の概略を図1に示す。次節より作業区分毎に必要な処理について紹介する。

3.1 サーバ、ソフトウェアの準備

サーバは自前でもレンタルでもよいが、Webサーバソフトウェア Apache [6]、Webプログラム言語 PHP [7]、データベースソフトウェア

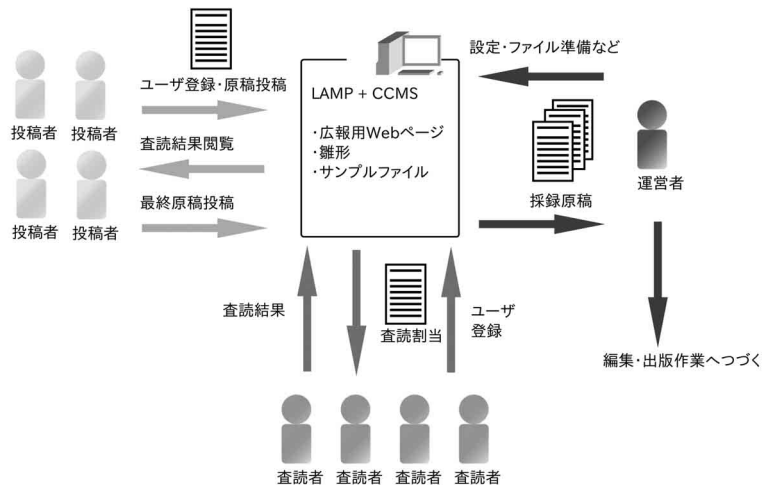


図1：CCMSによる学会運営。準備から最終原稿投稿締切まで。

MySQL [8], PHP 経由でのメール送信が利用できる環境が CCMS には必要で、当然 CCMS 自体も必要である。我々は、Linux 系 OS である Ubuntu [9] がインストールされた PC を運用しているが、この OS では、CCMS 以外のソフトウェアはすべてコマンド一つでインストールできる。他の多くの UNIX 系 OS でもコマンド一つで容易にこれらのソフトウェアをインストールできるようになっており、CCMS 用の環境構築は難しくない。CCMS については、公式サイトよりダウンロードし、Web サーバで公開されたディレクトリに適切なアクセス権で展開するだけで、インストール作業はブラウザから行う仕組みである。したがって、Web サーバ、データベースサーバの運用経験のあるものであれば、そのインストールは容易であろう。

しかしながら、詳しくは後述するが CCMS 自体にバグ・不具合があるため、国際会議での運用にはソースコードの変更が不可欠である。変更後は、各種動作テスト、公式ページとのリンク、システム利用方法の説明なども準備する。また、システム管理者が担当するとは限らないが、原稿の書式やサンプルファイルなどの準備も必要である。

3. 2 運用開始から査読開始まで

運用を開始し広報されれば、著者がユーザ登録、投稿作業を行うため、締切までの運用上の作業としてはユーザサポートぐらいである。締切後、査読に関する作業を行うことになる。まず、投稿者のミスなどによる 2 重投稿、原稿のないもの、原稿が閲覧できないものなどをチェックする。そして、査読候補者の選定・依頼を行うことになるが、CCMS には、それをサポートする機能も実装されている。その機能では、依頼を受ける査読者は自ら査読者としてユーザ登録を行うことになるが、我々の場合、利用しなかったため、管理者アカウントで一人ずつ登録した。この作業については、表計算などの list があれば、自動化できると考えられる。どの査読者がどの論文の査読を行うかに

ついても、管理者アカウントで設定できるが、これについては、データベースの書き換え箇所も特定できており、自動化は可能である。これらの処理の後、査読者の査読依頼の文章を登録し、システムの設定を査読可能状態にすることで、査読者への依頼メールが送信される。

3. 3 査読開始から最終原稿締切まで

査読締切が近づく頃に、再度査読者へのリマインダなどもシステムから可能である。そして締切後、システムのデータから、実行委員会で採否が決定され、各論文の採否が決定される。その結果を、システムに登録し一斉送信機能で各著者へ連絡される。結果を受け取った著者は、システムにログインすることで査読コメントも閲覧することができ、採録された論文については、最終原稿締切まで修正した原稿を投稿することができる。

3. 4 CCMS での運用における問題点

ここまで、CCMS が学会運営に有効であることを示したが、実際の運用では、下記の点で問題となった。

1. メールアドレスが30文字までとして処理されており、長い物は送れない
2. メール送信機能の CC が機能しない
3. 「所属」の文字数も不足
4. 会員種別(学生・一般、学会会員、非会員)などがない
5. レジストレーションの機能があるものの、著者と参加者の区別、インターフェースなどが分かりにくい
6. 査読者の登録を一人ずつしかできない
7. 採否の結果も Web のインターフェースから一件ずつ入力する必要がある
8. PDF チェック機能を off にしているにもかかわらず、ユーザからは利用されている状態となっている

1 から 5 はソースコードを読み、改変することで対応した。6 については、入金が振込あるいは現地での徴収だったので、参加者の情報が収集できればよく、独自にシステムを構築した。7 から 9 は、今後対応が必要であるが、これまでは運営側の対応でカバーした。

出版までには、下記の作業が必要であるが、これらの作業に CCMS 自体は利用しない。

- ・原稿チェック
- ・テクニカルプログラムの確定(論文集の構成)
- ・挨拶などの文章
- ・CCMS のデータとテクニカルプログラム構成データから、原稿にページ番号、ヘッダなどを追加
- ・挨拶、目次、索引などを作成
- ・論文集としてまとめる
- ・CD, Web, 印刷など配布、公開のための処理

これらの作業を効率的に実施する手順、プログラム等についても、上田教授の Web ページで公開されており、上田教授が利用している CCMS のバージョン V1_2a を用いていれば、概ね問題なく利用できる。しかしながら、我々は異なるバージョン

1-3c を用いたこと、これら処理の一部に商用の PDFlib [10] が用いられていたことから、手順はほぼ踏襲したものの、その実装は独自に行う必要があった。

4 出版システムの完全 OSS 化

上田教授の方法では、投稿された論文の編集に PDFlib とその専用ライブラリ PDI が必要である。PDFlib は、多くのサーバ用 OS に対応しており、使用できる言語も C, C++, Cobol, Java, Perl, PHP など、多くの言語で利用できる。また、PDF ファイルの作成能力も高く、PDF に関するアプリケーション開発に最適だと考えられる。しかし、価格は数十万必要であり、簡単に購入できる額とは言い難い。

出版システムでの PDF ファイルの処理としては、ページ番号、ヘッダの付加、ファイルの結合ができればよく、それらの処理ができるライブラリであれば、高機能である必要はない。プログラム言語は限定されないが、我々は UNIX 系 OS での処理を行っているため、UNIX 系 OS で動作する言語が好ましい。

これの条件で、いくつかのライブラリを調査し

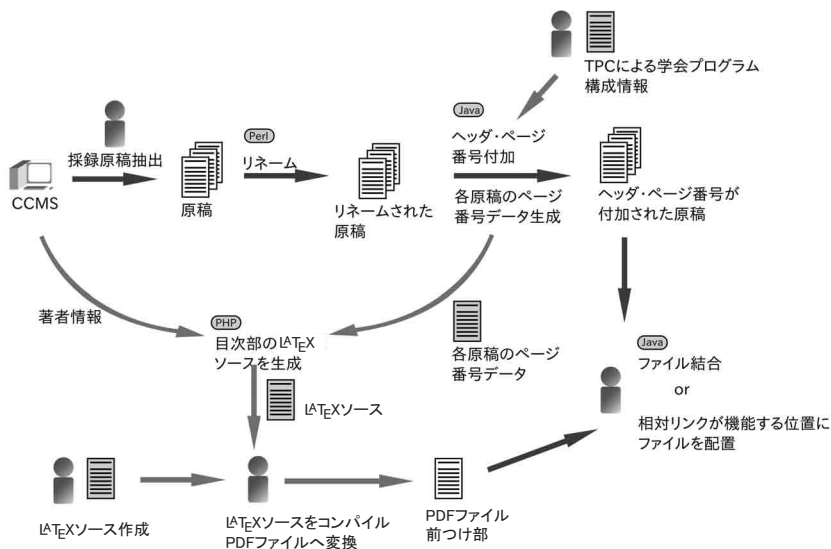


図 2：出版までの作業.

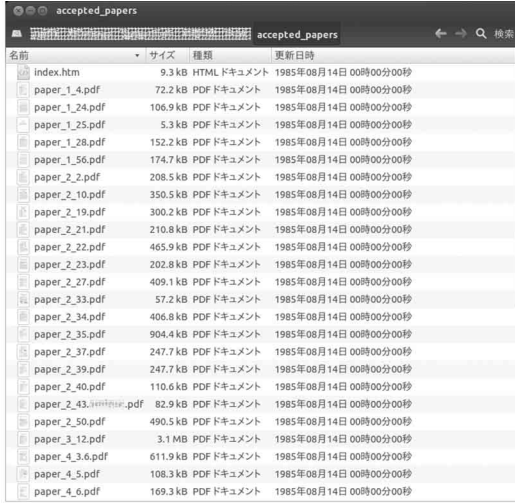


図3：CCMS を用いて一括ダウンロードした採録論文の一覧。

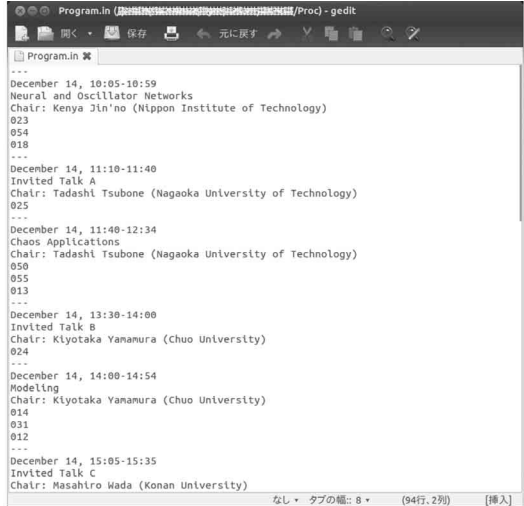


図4：学会プログラム（発表順など）の入力データ。

だが、既存 PDF ファイルの読み込みと改変が可能であり、OSS であるものは iText [11] 以外にほとんどなかった。iText のライセンスは AGPLv3 [12] で、Java 用の PDF ライブラリである。iText は公式ページでのリファレンスなどの情報も充実しており、経験のある Java で開発できることから、PDF ファイルの編集には Java と iText を用いることとした。図2は、実際に行った出版までの作業を表している。以下に、これら一連の作業について紹介する。

採録された論文は、図3のように CCMS の機能を用いて一括ダウンロードすることができる。論文のファイル名は、基本的には paper_(Topic No.)_(Paper ID).pdf の形で保存されているものの、一部のファイルは paper_(Topic No.)_(Paper ID)_(ユーザ名, 何らかの番号など).pdf と規則が分からないものもあった。しかしながら、Paper ID までは統一されているため、ファイル名を (Paper ID).pdf に変換するプログラムを Perl で作成し処理した。

次に、Java+iText でページ番号の付加とヘッダ部への会議名、ロゴの付加を行うプログラムを作成し処理した。この処理のアルゴリズムは上田教授の手法を踏襲している。テクニカルプログラ



図5：ヘッダー、ページ番号を付加された原稿。

ムチェアが構成したプログラム（発表順）に沿って、図4のように、セッションタイトル、ペーパーIDなどを記したテキストファイルを作成する。その内容を読み取り、順番にページ番号、ヘッダを図5のように各原稿に付加する。CCMSには、

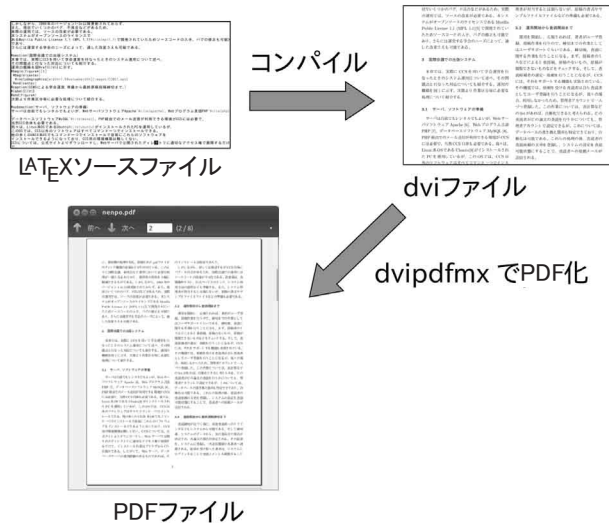


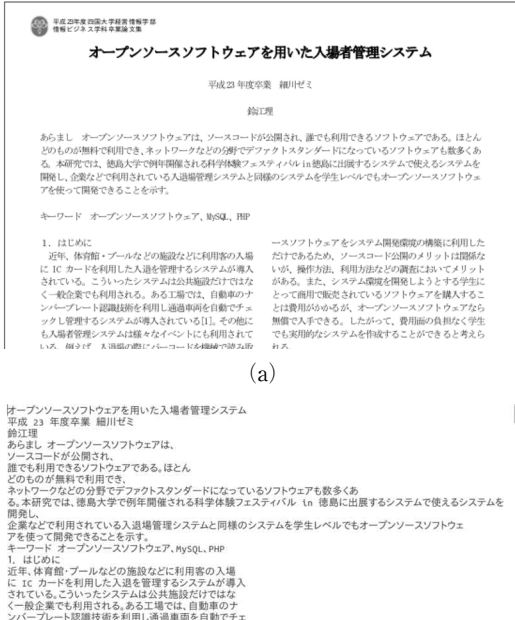
図6：LaTeXによるPDFファイルの作成。

著者情報入力欄に原稿のページ番号も登録するようになっているが、原稿との対応はチェックされていないためプログラムで改めてカウントしページ番号を付加している。その情報は目次に利用できるため、別途テキストファイルとして出力される。

表紙、委員長の挨拶、会場案内、目次、索引などを別途用意し、各論文と結合する必要がある。これらの作成にはLaTeXを用いる。LaTeXは、図6のように、ソースファイルと呼ばれるテキストファイル上に、文章とその書式をコマンドとして書き込み、コンパイルという作業を行うことで、書式が適用されたdviというファイルが生成される。このdviファイルを開覧、印刷するソフトウェアも整備されており、さらにはdviファイルを変換するソフトウェアもあるため、必要な文章をPDFファイルとして生成することが可能である。しかしながら、さまざまなコマンドを使いこなさなければ、思い通りの体裁を整えることはできず、ワープロソフトのように一般的であるとは言えない文書作成システムである。ここでこのLaTeXを用いる理由は、PDFのしおり、リンク機能を実装でき、プログラマブルに文章が作成できる点である。ページ番号を付加する

時に生成したデータ、学会プログラムの情報、そして、データベース上の論文タイトルおよび著者の情報を用いて、目次、索引となるLaTeXのソースファイルをプログラミングにより生成する。これによって、ヒューマンエラーの多くを防ぐことができ、学会プログラムの変更などにも即座に対応できる。この処理を行うPHPプログラムは上田教授のWebページで公開されており、PHPとLaTeXを使えるものであれば、生成されるソースコードを調整することも難しくはない。各論文のPDFファイルと目次などの結合は、基本的にリンクとして目次などから閲覧できるため不要であるが、印刷などでは不便であるため、結合したのも生成する。この生成にもJava+iTextを用いた。

本章では、行われた処理とその実装について述べたが、上田教授がそれらの手順、プログラムを明確にWebに公開されている点が、まさしくOSSといえ、その恩恵を我々は大いに得られた。具体的には、公開されているプログラムだけでなく、その手法の解説もあったため、投稿システムをその情報を基に運用することができ、今回の完全OSS化の部分に関しても、コーディングだけの問題であったため、アルゴリズム等で悩むことなく実装できた。



(a)

図 7：pdftotextによるテキストデータ抽出例。(a)PDF形式の原稿。(b)抽出されたテキストデータ。

5 卒業論文集への応用

PDF ファイルを取りまとめる必要のあるさまざまな状況において、前章までの手法、およびプログラムをカスタマイズすることで利用できる。本章では、平成24年度四国大学経営情報学部情報ビジネス学科の卒業論文を例に応用の可能性を示す。本学科の卒業論文提出は、各指導教員へPDFファイルで提出することとなっている。したがって、図1のようにCCMSなどから取り出せる情報がなく、提出されたPDFファイルと図1でTPCの作成している発表順のデータのみから論文集を作成する必要がある。この点の対応が学会の論文集の場合と大きく異なる。CCMSの持つ情報は、各論文に含まれているため、PDFファイルの文章をテキストとして抽出できるコマンドpdftotextを利用した。このコマンドは、OSSであるXpdf [13]に含まれるものである。このコマンドを利用し、論文をテキストとして取り出し、プログラムで、タイトル、著者名、アブスト、キー

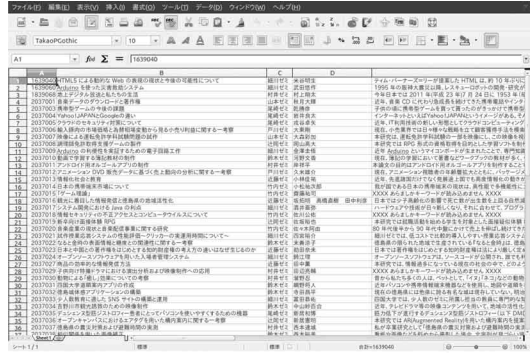


図 8：抽出されたテキストデータから著者情報等を抽出したもの。

ワードなどを抽出すればよい。

まず、pdftotextを使って、すべての原稿からテキストを抽出し、テキストファイルとして保存するPerlプログラムを作成した。図7は、プログラム実行結果である。原稿作成時のPDFへの変換の方法によっては、うまく抽出できないものもあるようだが、本学科の学生は、概ね同一バージョンのWordとそのPDF出力機能を利用しており、抽出できないものはなかった。

次に、抽出されたテキストファイルから目次作成用データを作成し、cvsファイルとして保存するPerlプログラムを作成した。図8は、実際にcvsファイルとして保存されたデータである。一部、作成側の書式ミスなどで抽出できていないものがあるものの、書式通りあるいはそれに準ずるものは抽出できている。

これらの処理によって、論文集に必要なデータを得ることができる。あとは、国際会議の場合と同様に、発表順から原稿にページ番号ヘッダなどを追加し、その情報を基にL^AT_EXで目次を作成。そして、それらをPDFファイルとしてとりまとめるだけである。

過去の卒業論文集では印刷したものが作成されていたため、その業務を委託する業者へ全員の原稿を集める必要があった。しかしその一方で、書式ミスなどは担当教員だけでなく、提出先の事務でもチェックしていたため、2重チェックとなるメリットもあった。一方、ここ数年の卒業論文集



図9：卒業論文のあらまし一覧。

は、ゼミ毎にUSBメモリで提出する方法をとっているため、原稿のチェックは担当教員のみであり、一部体裁に問題のあるものもあった。本論文で述べた方法では、この部分の処理でチェックとなった部分もあったが、締切後の指摘となり教育上好ましいとは言えない。また、担当教員が修正を望まない場合に、プログラムでイレギュラーな処理が必要となったなど、バグにつながり兼ねない細かな処理が必要となった。これらの処理では、提出前に学生自らしっかりチェックするよう指導することが、教育上最も重要であるが、その支援としてここで紹介した処理のプログラムを基に提出前のチェック用webシステムなどを開発できると考えられる。また、本方法では、図9のようなあらましの一覧なども、処理で用いるデータを

活用すれば容易に作成できる。提出原稿の質がしっかりしていれば、これらの処理自体は数秒でできるものであり、発表順の変更、原稿の再提出などの対応は、実質的には、CDへの焼き付け、印刷などの作業日程のみに依存する。しかしながら、実際の処理では提出された原稿の書式などが間違っているものが含まれており、プログラムで概ね対応できるものの、その処理をすり抜けるものがないとは断言できない。したがって、提出時のチェック、作成された論文集のチェックなど、最終的には人によるチェックはかかせない。

6 他の応用例とオープンソース活用

他にも、PDFファイルをまとめるという点では、多人数による報告書をまとめる場合にも有効である。今回紹介していないが実際に行った事例としては、地域学習コンテンツの教材 [14] の報告書作成にあたり、差し替えのある可能性のある135個のPDFファイルを実質1日でまとめることができた。これは、筆者らがすでに今回紹介した手法のノウハウやプログラムを持っていたことだけでなく、まとめる必要のあるPDFファイルに関する情報共有がしっかりなされていたことが大きかった。しかしながら、この事例は一般的に有効な手段とは言い難い。それは、この事例ではプログラムやまとめるページは独自に改変、新規作成しており、それらの内容について、単独あるいはごく少数の人間で決定することは、通常ではあまりないと考えられるためである。学会や卒業論文のように繰り返し行われ書式なども概ね確定しているのであれば、多少の労力に見合う結果が得られるが、そうでない場合、PDF編集ソフトウェアを利用し作成する方が一般的には有効であろう。

この点においても、本論文の内容は有効に活用できる状況すなわち、多数のPDFファイルを編集するという状況は非常に限られており、そして、その状況に置かれ、かつ本論文の内容を活用できる程度のプログラム能力を有しているという場合

はごく稀であると考えられる。一方で、そのような状況に置かれた筆者らの場合には、非常に有効であったことも事実であり、今後、電子化がさらに進んだ場合に、より多くの者にとって有益となる可能性がある。

このような多くの者にとって有益ではないものの少数の者にとって非常に有益なものの場合に、OSS は非常に有効であるといえる。具体的には、PC の処理能力の向上と一般化、PC-UNIX の発展、さらに、PC-UNIX にインストールされ、活用されてきた品質の高い数多くのソフトウェア、インターネット環境の高速化、これらに対応するプログラム言語やライブラリなど、低価格が進んだハードウェアと通信費のみで情報の入手とその実践の場が手に入る状況であり、それらを学び活用できるものであれば、その能力と必要性あるいは興味に応じて、自ら改変、開発できるのである。実際、このような状況から、多くの OSS の開発が行われ、コード自体の閲覧、改変ができるなどから、さらに、多くの OSS の開発へとつながっている。学習環境としてもコード自体やその改変ができることは、あらたな開発者を育てることとなっている。実際に、CCMS のバグ修正を通じて、PHP のコーディングを学べ、運用上もバグをとったものを利用できたことなど、OSS の恩恵は計りしれない。

7 まとめと今後の課題

本論文では、論文集の出版における OSS の活用を実際に実施した事例とともに紹介し、その有効性と OSS の意義について述べた。

今後の課題としては、CCMS のパッチの公開、作成したプログラム群の公開、一連の処理の一般化、さらなる利用方法の検討などが挙げられる。

参考文献

- [1] facebook, <https://ja-jp.facebook.com/>
- [2] amazon, <http://www.amazon.co.jp/>
- [3] 講演論文集の作り方,
<http://risa.is.tokushima-u.ac.jp/tetsushi/makeproc/>
- [4] IAPR Commence Conference System,
<http://iaprcommence.sourceforge.net/>
- [5] Mozilla Public License Version 1.1,
<http://www.mozilla.org/MPL/1.1/>
- [6] Apache, <http://httpd.apache.org/>
- [7] PHP, <http://php.net/>
- [8] MySQL, <http://www.mysql.com/>
- [9] Ubuntu, <http://www.ubuntu.com/>
- [10] PDFlib, <http://www.pdflib.com/>
- [11] iText, <http://itextpdf.com/>
- [12] GNU Affero General Public License,
<http://www.gnu.org/licenses/agpl-3.0.html>
- [13] Xpdf, <http://www.foolabs.com/xpdf/home.html>
- [14] 池本有里 他, “地域学習コンテンツの教材バンク構築”, 電気関係学会四国支部連合大会講演論文集 p.324, 2012.