

電子フォームを用いた申請書処理の効率化

曾根直人*, 林 秀彦*, 菊地 章*, **

高度情報研究教育センターでは業務に関連する多くの申請書を扱っている。それら申請書の形式は紙に印刷した書類や電子化されたファイルなど多岐にわたっている。電子化されたファイルはワードプロセッサを用いて内容を記入し、電子メールの添付等により提出することでペーパーレス化を図ることができる。しかしこのようにアプリケーションのデータとして提出された場合、申請された情報をデータベースなどに登録し整理する場合は、いちいちファイルを開き必要な項目をカット&ペーストするなど複雑な作業が必要になり、事務作業の効率化には結びつかない。そこで電子フォームを活用し、利用者から提出された XML データを直接データベースへ読み込むことで事務作業の自動化を図るシステムを構築した。このシステムを利用することで従来よりも作業の自動化を進め、電子化のメリットを生かした効率的な事務処理が行える。

[キーワード: 電子フォーム, PDF, XML, XSLT, データベース]

1. はじめに

高度情報研究教育センターでは業務のために多くの申請書が存在しており、紙に印刷したものからワードプロセッサや PDF などのデジタルデータで提供されるものなどさまざまな形式の申請書がある。申請書に記入された項目はその後の事務処理のために電子化し、データベースなどに保存するが、紙で提出された書類はもちろんデジタルデータで提出されたものであっても必要な項目を自動的に抽出することが難しいなどの理由でカット&ペーストといった手入力を行う必要がある。手動での入力作業は複雑であり、間違いも発生しやすくなる。

手入力によるミスの発生を防ぎ、事務処理の効率を改善するための試みとしてセンターの扱う申請書の一つである「ウィルス対策ソフトウェア申請書」を電子フォーム化し、入力データを直接データベースへ入力可能なシステムを構築した。本稿では、ICT を活用した事務処理の効率化の一例として、電子フォームを利用した申請情報の自動データベース化の方法について提案する。

2. 電子フォーム

電子フォームは従来紙の帳票で行っていた申請などの業務を電子化し、アプリケーションを介して入力された項目をそのまま業務用のデータとして利用するための仕組みである。業務の効率化のため、企業や自治体などでは採用例が増えている。

2. 1 Web を利用した電子フォーム

電子フォームを実現する一つの方法が Web ページとしてフォームを組み込んだページを用意し、CGI プログラムとしてデータを受け取り処理する方法である。この方法は普段利用している Web ブラウザをそのまま

利用できるため、利用者への負担が少なく、採用例も多い。しかし、フォームの作成やデータを処理するためのプログラムを開発する必要があり、電子フォームを Web フォームで実現するには開発コストおよびメンテナンスコストが必要になる。

2. 2 アプリケーションを利用した電子フォーム

Web ではなくアプリケーションを利用し、電子フォームを実現する方法もある。いくつかのアプリケーションが電子フォームの作成に対応しているが、今回はジャストシステム社一太郎 XML テンプレートクリエータ 3 および Adobe 社 Acrobat 8 を用いてフォームの作成を行った。ここでは、これら 2 つのアプリケーションについて紹介する。

(1) 一太郎 XML テンプレートクリエータ 3

ジャストシステム社のワードプロセッサ一太郎を利用したフォーム作成ソフトに XML テンプレートクリエータ 3 がある [1]。このソフトは一太郎のアドインソフトであり、一太郎上で電子フォームを作成、電子フォームに入力されたデータを XML 形式で取り出すことができる。利用者は一太郎もしくは無料で配布されている一太郎ビューアを使うことで、電子フォームへの入力が可能になる。電子フォームに入力したデータはファイルとして保存できるほか、電子メールに添付する形で指定アドレスへ送信する設定も可能である。添付ファイルには XML テンプレートクリエータのファイル形式である jtdx 形式および XML 形式が選択できる。

一太郎の画面で直接電子フォームを作ることができるため、紙ベースの帳票を一太郎で作成していた場合、簡単に電子フォームを作成できる。

図 1 に XML テンプレートクリエータ 3 を用いて一太郎上で電子フォームを作成している例を示す。

* 鳴門教育大学 高度情報研究教育センター

** 鳴門教育大学 生活・健康系(技術)教育講座

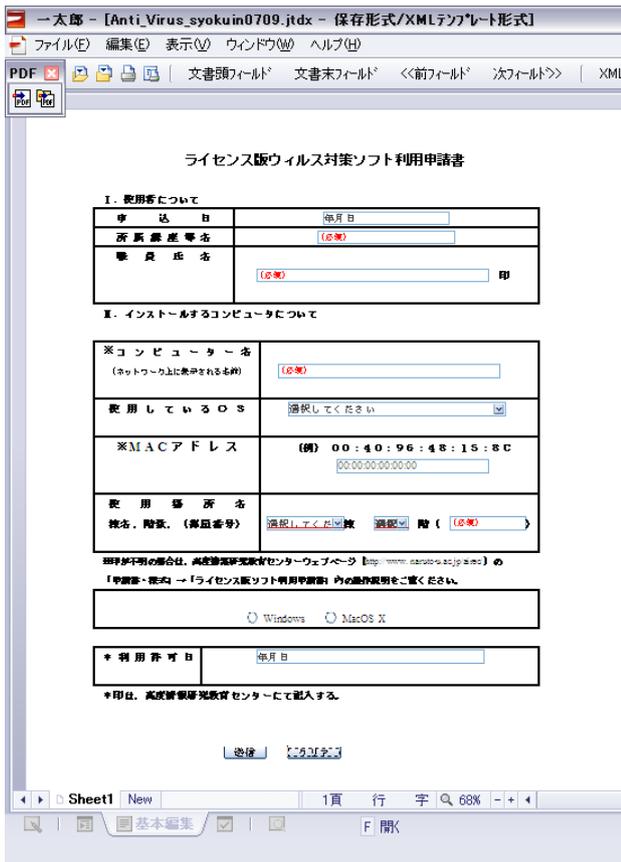


図1 一太郎を用いた電子フォーム作成

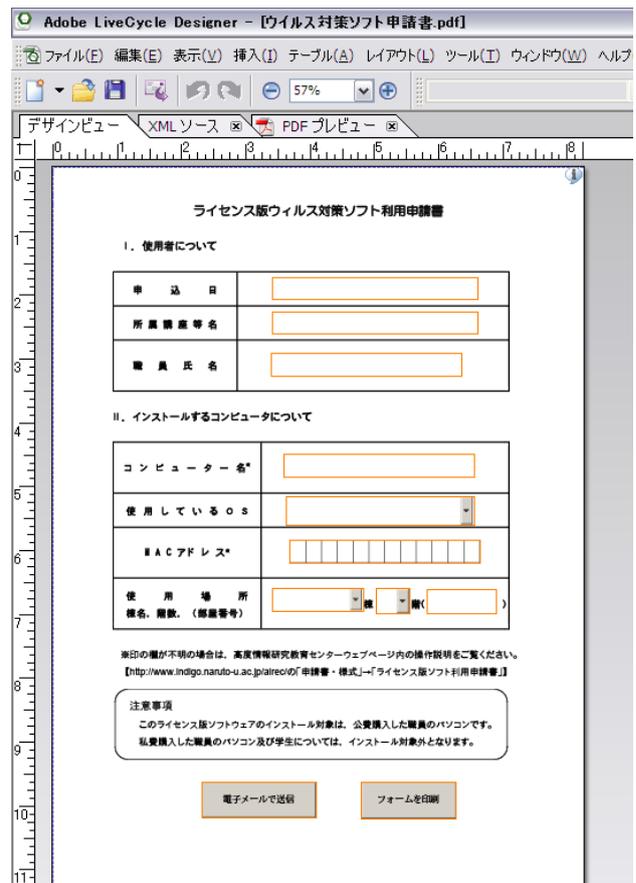


図2 Acrobat 8を用いた電子フォーム作成

(2) Acrobat 8

Adobe社の開発したPDF (Portable Document Format) はその名が示すように非常に多くの機種で表示、印刷がサポートされており広く普及している。申請書などの帳票がワードプロセッサによって作成されたものであればPDFに変換後、Acrobat 8 Professionalに付属の電子フォーム作成ソフトウェアLiveCycle Designerを利用して電子フォームを追加することができる。図2にLiveCycle Designerを用いて電子フォームを作成している例を示す。

利用者はPDF閲覧用ソフトウェアAdobe ReaderやAcrobat Readerを使って電子フォームに値を入力することができる。またAcrobat 8からは電子フォーム作成時にオプションを設定することで、電子フォームに値を入力したPDFファイルを開覧用ソフトでも保存可能になった。この機能により、利用者は提出したフォームの内容を手元に保存しておきたい場合、紙に印刷するほかファイルの形で保存できるようになった。

PDFを用いた電子フォームも一太郎XMLテンプレートクリエイータ同様、入力されたデータをXMLの形式で取り出すことができるほか、電子メールに添付する形で指定アドレスへ提出が可能である。

3. ウィルス対策ソフトウェア利用申請書処理システム

今回我々はセンターが一括購入し利用者へ提供しているウィルス対策ソフトウェアの利用申請書を電子フォーム化し、データベースと組み合わせることで処理の効率化を試みた。本システムによる利用申請書の処理の流れを図3に示す。

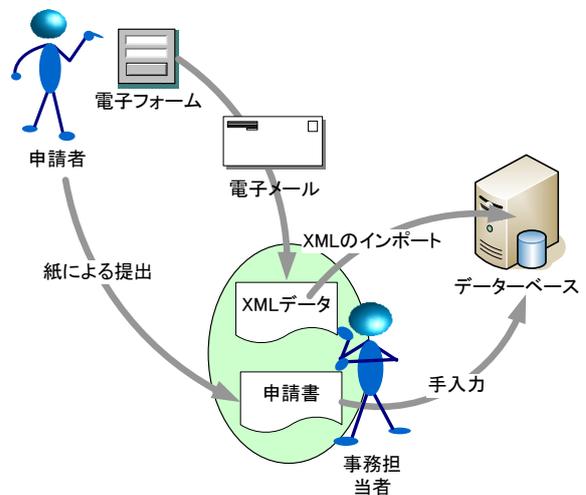


図3 利用申請書処理の流れ

3. 1 電子フォーム

ウィルス対策ソフトウェア申請書において、利用者が記入すべき項目を表 1 に表す。

表 1 利用申請書記入項目

項目名	データ形式
申込日	日付
所属講座等名	文字列
職員氏名	文字列
コンピュータ名	文字列
使用している OS	選択 (Win or Mac)
MAC アドレス	文字列
使用場所 (建物, 階, 部屋番号)	文字列

電子フォームのアプリケーションとして今回は Acrobat を選択した。これは Acrobat が幅広い OS で電子フォームを利用できる点を評価したためである。一太郎は Macintosh 上で動作しないため電子フォームの利用者を制限するが、ワードプロセッサを用いて電子フォームを作成できるため、電子フォームの作成作業は Acrobat より容易であった。また Web を利用した電子フォームもプラットフォームに依存せず、幅広い OS で利用可能であるが、開発やメンテナンスコストの負担が問題となるため、採用しなかった。

図 4 に示すように、Acrobat により作成した電子フォームでは表 1 の値を入力し、かつ利用者の入力を補助するためにドロップダウンリストや入力値の制限、マウスをフォーム上に移動したときに入力項目のヒント情報表示するツールヒント、より詳しい情報を記述した Web ページへのリンクを設定した。これらの入力補助機能は従来の紙の申請書では実現できないため、電子フォームは利用者にとっても利便性を向上させると考えられる。

全ての利用者が電子フォームを利用し、データを提出する形式が理想的ではあるが、従来の紙ベースでの提出を好む利用者もいることが予想される。電子フォームと紙の申請書二種類を作成、維持するのは煩雑であり、統一することが望ましい。そこで、電子フォームを印刷した場合、“電子メールで送信”、“フォームの印刷”ボタンは印字されないよう設定を行った。

この電子フォームにより、添付される XML データの例を図 5 に示す。

3. 2 データベース

申請されたデータを保存するためにデータベースを構築した。このデータベースは FileMaker Pro 8 Advanced 上に構築しており、

- 新規レコード追加(データ手入力)

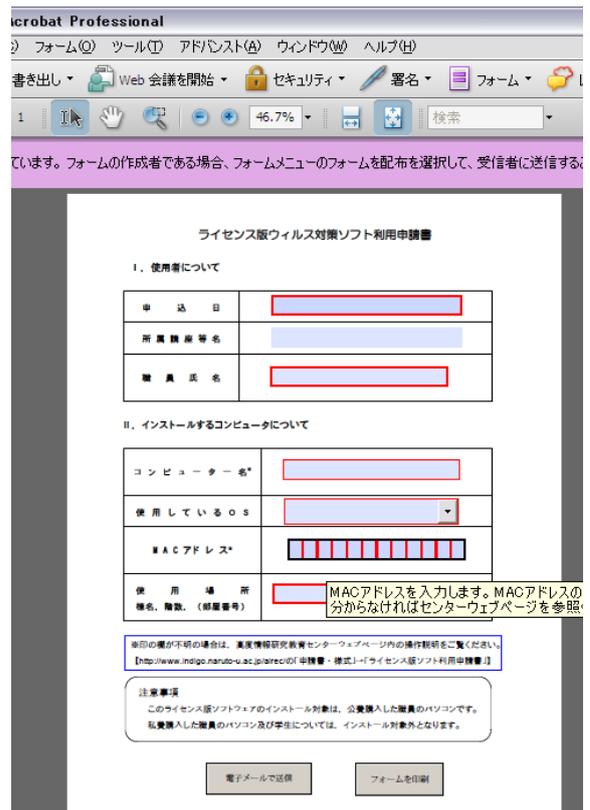


図 4 PDF による電子フォーム

```
<?xml version="1.0" encoding="UTF-8"
?>
<topmostSubform>
  <所属>高度情報研究教育センター</所属>
  <氏名>曽根直人</氏名>
  <コンピュータ名>mint</コンピュータ名>
  <OS>WindowsXP, Vista</OS>
  <MAC>000B97B8D628</MAC>
  <棟>高度情報研究教育センター</棟>
  <階>1</階>
  <部屋番号>情 103</部屋番号>
  <申込日>2007-12-10</申込日>
</topmostSubform>
```

図 5 XML データの例

- XML インポート
- 承認メール作成

処理をスクリプトとして登録している。データベースの画面を図 6 に示す。

新規レコードは紙による申請書提出に対応する処理であり、新規レコード登録画面が開き、各項目を手入力する。

XML インポートは電子フォームにより提出された XML データをデータベースへと取り込む処理を行う。提出され

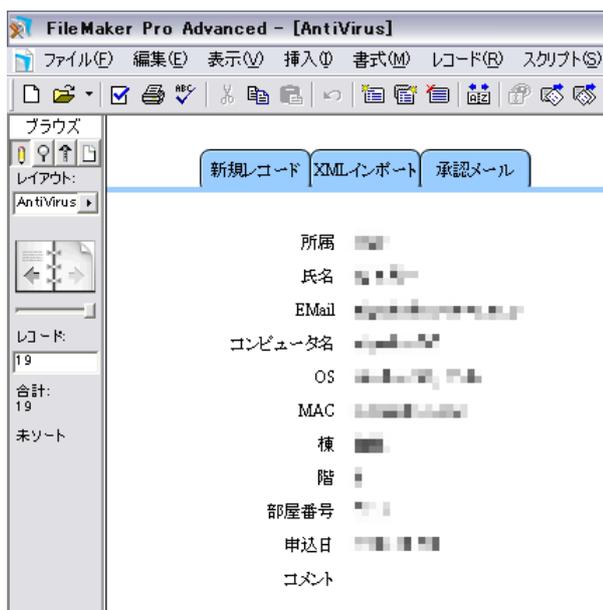


図6 利用申請書データベース

た XML データをファイルとして保存し、そのファイルを読み込む。PDF の電子フォームに入力されたデータは図 5 のようになっている。XML データを FileMaker へ読み込むためにはデータは FileMaker FMPXMLRESULT DTD(文書型定義)に準拠している必要がある [3]。電子フォームから送信されたデータはそのままで読み込めないため、XML データ読み込み時に指定した XSLT を利用して FMPXMLRESULT 形式へ変換する。変換に利用した XSLT を付録に示す。このファイルは FileMaker に付属のサンプルを元に作成している。読み込まれたデータは MAC アドレスの ASCII 文字への変換などスクリプトによるデータの正規化処理が行われる。

承認メール作成はデータベースに登録された情報を元に申請者へ登録確認用のメールを作成する。これは必要な情報を申請者へ告知するとともに申請者の本人認証を兼ねている。電子フォームの利用では申請はデータだけのやり取りになるため窓口で申請者の確認を行うことができないが、承認メールを送ることで本人以外の請求であれば不正が発覚する。

承認メールの文面はスクリプトにより必要な項目を抽出し、自動生成しており、事務作業の負担を増やさずに確認処理を行うことができる。

4. まとめ

事務作業の効率化を図るため、電子フォームと XML を利用したウィルス対策ソフトウェア申請書処理システムを開発した。このシステムは、

- PDF を利用した電子フォーム
- 申請データ処理用データベース

で構成されており、これまで手作業で行っていたデータベースへの登録作業を XML データの読み込みにより自動

化した。また、電子フォームは利用者に対してインタラクティブにメッセージを表示することができるなど紙と比べて優れたユーザビリティを有する。例えば、ウィルス対策ソフトウェア申請書では MAC アドレスを記入する項目があるが、MAC アドレスの確認方法を解説したページへのリンクを埋め込むなど紙媒体では実現できないことが可能になる。

学内には多くの帳票が存在しているが、利用者から見れば帳票への記入は面倒であり、帳票を処理する事務側からも項目の抽出や入力は大変な労力を必要としている。今回我々が試みたように帳票を電子フォーム化することにより、利用者、事務双方にメリットがもたらされる。今後、事務処理の効率化を目指すためにも電子フォームの利用が広がることを期待し、本稿が参考になれば幸いである。

参考文献

1. ジャストシステム. XML テンプレートクリエイター3. (オンライン)
<http://www.justsystems.com/jp/software/dt/xml3/>.
2. Adobe. Adobe Acrobat Professional. (オンライン)
<http://www.adobe.com/jp/products/acrobatpro/>.
3. FileMaker. FileMaker Pro 8 ヘルプ.

付録

電子フォームから送られた XML ファイルをファイルメーカーにインポートするための XSLT ソースコードを示す。このソースは FileMaker Pro8 Advanced に付属のサンプル”msdso_attrib.xslt”を元に作成した。

<!—

ファイル: msdso_attrib.xslt

属性ベースの MSDSO 文法に基づいたデータを FMPXMLRESULT 文法に変換し、ファイルメーカーでインポートできるようにします。

Copyright c 2002 FileMaker, Inc.

All rights reserved.

ソースおよびバイナリ形式での再頒布および使用は、変更のあるなしに拘わらず、下記の条件が満たされる場合にのみ許可されます。

- * ソースコードの再頒布には、上記の著作権表示、本条件書および下記の否認文を明記し保持しなければなりません。
- * バイナリ形式での再頒布には、上記の著作権表示、本条件書および下記の否認文を、頒布と共に提供されるドキュメンテーションまたはその他の資料において複製しなければなりません。
- * FileMaker, Inc. の名称や著作者の名称は、書面による特定の事前許可なしに、本ソフトウェアから派生する製品を推奨あるいは促進するために使用してはなりません。

本ソフトウェアは著作権所有者および著作者によって「現状のまま」の状態を提供され、明示または黙示の保証は商品性および特定目的への適合性なども含め一切否認されます。著作権所有者あるいは著作者は、本ソフトウェアの任意の使用により発生する如何なる直接的損害、間接的損害、付随的損害、特別損害、懲罰的損害あるいは派生的損害(代用品またはサービスの調達、使用の損失、データの損失、または利益の損失、あるいは事業の中断による損失なども含む損害)に対しても、因果関係を問わず、契約の有無、無過失責任の有無、不法行為(不注意による事故その他を含む)の有無を問わず、一切責任を負いません。損害発生の可能性についてたとえ知らされていたとしても同様とします。

—>

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/*">
    <FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
      <ERRORCODE>0</ERRORCODE>
      <PRODUCT BUILD="" NAME="" VERSION=""/>
      <DATABASE DATEFORMAT="M/d/yyyy" LAYOUT="" NAME="" RECORDS="{count (/*/*)}"
TIMEFORMAT="h:mm:ss a"/>
      <METADATA>
        <xsl:for-each select="/*/*">
          <FIELD>
            <xsl:attribute name="EMPTYOK">YES</xsl:attribute>
            <xsl:attribute name="MAXREPEAT">1</xsl:attribute>
            <xsl:attribute name="NAME"><xsl:value-of select="name()" /></xs
l:attribute>
            <xsl:attribute name="TYPE">TEXT</xsl:attribute>
          </FIELD>
        </xsl:for-each>
      </METADATA>
      <RESULTSET>
        <xsl:attribute name="FOUND"><xsl:value-of select="count (/*)" /></xsl:a
ttribute>
        <xsl:for-each select="/*">
          <ROW>
            <xsl:attribute name="MODID">0</xsl:attribute>
            <xsl:attribute name="RECORDID">0</xsl:attribute>
            <xsl:for-each select="*">
              <COL>
                <DATA>
                  <xsl:value-of select="." />
                </DATA>
              </COL>
            </xsl:for-each>
          </ROW>
        </xsl:for-each>
      </RESULTSET>
    </FMPXMLRESULT>
  </xsl:template>
</xsl:stylesheet>

```