

ワープロ帳票の回収による構造化データベースの構築法と 集計冊子の自動生成

鈴木 靖征[†] 稜川 友宏[†] 加賀 周[†] 富樫 敦[†]

静岡大学情報学部[†]

1. はじめに

自動集計や 2 次利用性などから、構造化データの蓄積は極めて重要である。事務文書の多くは様式が定められており構造化に適しているが、様式とデータが渾然一体となったワープロ帳票として流通しているため、せっかく電子的に入力されたデータがほとんど再利用されていない。

本稿では、回収したワープロ帳票を XML により構造化してデータベースに格納し、集計冊子の生成などに利用可能とする手法を提案する。この方法は、回収担当者に XML の知識を要求しないばかりでなく、すでに蓄積されているワープロ帳票も簡単に構造化可能であるという特徴を持つ。

2. 従来の事務処理

ワープロ帳票からデータを抜き出し構造化する場合、帳票のどの欄に何のデータが格納されているかを知る必要がある。このため、既存帳票のデータを構造化するには帳票ごとに専用のソフトウェアを用意する必要があり、現状では、Web フォームと CGI によるシステムに再入力するか、そもそも Web 上で原稿を作成する方法が一般的である。しかしながら、規格化ドキュメントのうちでも申請書や報告書などはドキュメント作成に対する推敲の割合が多く、オンラインの入力にはなじまない。実際、本学部のシラバス入力・修正システムはごく一部の教官しか利用しておらず、ワープロ文書を担当者にメールで送付する手続きが主流になっている。

3. 事務処理自動化システムの提案

ワープロ文書が支持されている理由の 1 つとして、手馴れたワープロを用いて自宅や移動中にオフラインで編集でき、オンライン入力を強いる Web インタフェースに比べ敷居が低いことがあげられる。実際、電子文書を完成させてからメールで提出するという方法は従来の紙文書提出のプロセスと感覚的に近く、Web インタフェースで電子文書を入力する方法は事務官の目前で書類を仕上げる感覚に近い。

このため、ワープロ文書から構造化データを抽出し再利用可能とする方法を提案する。データの蓄積・加工や合成文書作成のために中間データ形式を XML に

することで Web など別メディアへの変換も容易になる。本システムの特徴をまとめると以下ようになる。

ワープロ帳票からの構造化データ作成

- A) 入力者がレイアウトを意識しながら編集可能
- B) 入力者がローカルで編集できる
- C) 構造化されたデータの蓄積・加工ができる
- D) 蓄積されたデータから容易に冊子化、Web 化ができる

帳票処理の自動化

- E) 文書ファイルから構造化ドキュメントを作成してデータベース化
- F) 構造化ドキュメント同士を併合して合成ドキュメントを作成
- G) 目次やページ番号、索引を付した冊子体としてふたたび文書ファイルに戻す

3.1 ワープロ文書の構造化方式の検討

本章では、ワープロ文書からデータを抽出し、適切にタグ付けする方法について検討する。

3.1.1 タグ埋込済テンプレート配布方式の検討

はじめに、多くのワープロで用いられている共通フォーマットの RTF (Rich Text Format^[1]) を利用し、RTF のコメントコマンドを利用して XML タグを埋め込む方式を検討した。RTF は仕様が公開されており、加工が容易なテキストベースのフォーマットであったことが採用の主な理由である。

RTF のコメントコマンドを利用して XML タグを埋め込んだ文書ファイルはワープロで開くと普通の文書ファイルとして見える。このため、通常の帳票と同じ感覚でデータ入力すればコメント化された XML タグの間に文字が入力される。

データ入力済み RTF 文書から XML への変換には、Majix^[2] というフリーツールを利用する。Majix は見出し、段落、表のセルなどの文書体裁で文字列をマークアップしてくれるツールであるので、生成される XML は文書体裁ベースの XML 文書であり、意味でマークアップを行う文書構造の XML データは得られない。しかし、文書体裁のマークアップを消去し、埋め込んでおいたコメントを復活させれば文書構造ベースの XML 文書を得ることができる。

しかし、実際にこの方法を実装してみたところ文字削除とともにコメント化された XML タグも消えてしまう場合があり実用には至らなかった。また、既存の文書に対して構造化が難しいという欠点があった。

A Proposal for Construction of Structured Database and Compositional Booklet from Word-Processor Forms

[†]Yasuyuki Suzuki, [‡]Tomohiro Haraikawa,

[‡]Syu Kaga, [‡]Atsushi Togashi

3.1.2 帳票・テンプレート差分方式の検討

次に、回収した帳票と、タグ付けのために用意したテンプレートの差分を利用する方法を考案した。管理者は必要な部分が未記入なワープロの帳票を作成し、配布する。ワープロを用い、記入者はこの帳票の空欄にデータを入力して帳票を完成させ、回収者は同じ空欄に対してデータの意味 (<科目名> など) を入力してテンプレートを完成させる。

帳票とテンプレートは文書体裁が同じであるため、どのデータに対してどの意味でタグ付けをすればよいかは文書体裁タグを解析することで判明する。このため、データを対応するタグで囲むことにより、鑄型で抜き取るように必要な項目を構造化データとして抽出することができる。これをもとに文書構造ベースの XML 文書を作成することができる。

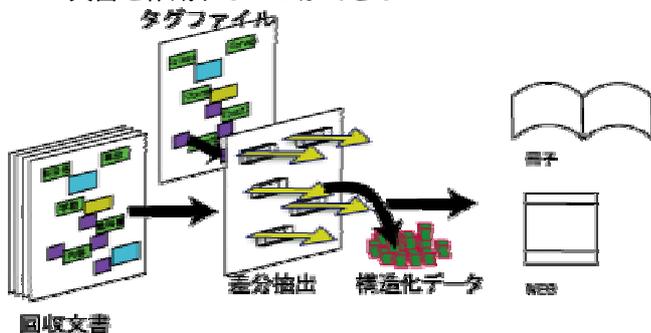


図 1 作業の流れ

この発想は非常に単純であるが、極めて有効である。タグ埋込法と異なり、回収者が RTF に直接手を加える必要がなく、蓄積されている既存帳票に対してもタグ付け用テンプレートをワープロで作成しさえすれば、回収者に XML の知識を要求することなく構造化が可能である。本方式で得られる構造化データは平坦な階層構造をもつが、多少の知識があれば、通常の階層構造へも XSLT で容易に変換可能である。

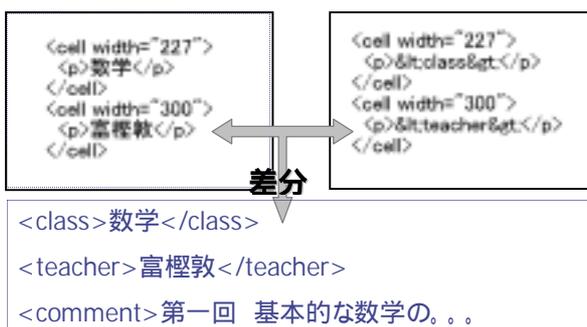


図 2 体裁ベース XML から構造化ベース XML への変換

3.2 構造化データの自動回収と自動集計方式の検討

ここまでで、システムの核となる“回収されたワープロ帳票から構造化データを抽出する作業の自動化手法”を提案した。本章では、構造化されたデータベースからの効率的な冊子化、Web ページ化を含め、入力・集計・出力の一連の工程を効率化する方式を検討する。

3.2.1 メールサーバによる文書回収受付

帳票の種類ごとにメールアドレスを設定し、受信された添付形式の帳票をメールアドレスごとにわりふり各帳票類毎に作成されている差分元タグファイルとの差分処理を次々に行っていくことにより構造化データベースを構築していく。しかし、入力項目の中には選択肢の中からひとつを選ぶというものもある。Web フォームであるならば入力者に用意された選択肢を選ばせることもできるが、ワープロ編集ではこれができない。そこで、差分元タグファイルのタグ情報の中に選択的な入力が必要であるというキーワードを設定しておくこととし、メールサーバは差分処理を行ったときに同時に選択肢に違反した入力があるかチェックを行う機構を設けた。もしも違反された入力を含む帳票の場合、違反箇所を明記したメールを返信し再編集を促すようにした。

3.2.2 XSLT による HTML への変換 (Web 化)

構造化データ抽出時には不要であった文書体裁ベース XML であるが、このレイアウト情報を有効に活用できることがわかった。これにより、データベースから取り出したデータを帳票とほぼ同等の Web ページに表示するための XSLT 変換ルールを自動で生成することができる。

3.2.3 ページ参照の問題 (冊子化)

Web ページにはページ番号の概念がないが、紙媒体ではページ番号は必須項目であり、なおかつ引用が行われている場合、引用部や参照部と、その対象との整合性がとらなければならない。RTF に書き戻す方式では、ページ参照を完全にはサポートすることが難しい。そこで XSL Formatting Object (XSL-FO) とそのプロセッサである FOP (Formatting Object Processor)^[3] を利用することによりページ参照をサポートすることができた。

4. まとめ

当初考案した埋め込み法では見込まれていなかった付加価値が差分法から導かれた。埋め込み法では、前もって必要な処理を施した帳票へ入力者が追加編集する方式なので、新規に作成されるワープロ帳票からしか構造化データを抽出することができなかった。しかし、差分法では同じレイアウトの文書を用意すれば既存の文書からも構造化データの生成は容易である。例えば、過去何年分も蓄えられた報告書などのワープロ帳票であっても、毎年レイアウトが変更されていなければ、同様の手順を追って構造化データを抽出しデータベース化することも可能である。

5. 参考文献

- [1] Microsoft, Specification of RTF 1.6, <http://msdn.microsoft.com/library/?url=/library/en-us/dnrftspec/html/rtf1spec.asp?frame=true>
- [2] TetraSix, Majix <http://tetrasys.dhs.org/>
- [3] The Apache XML Project, FOP (Formatting Object Processor), <http://xml.apache.org/fop>