文字画像のマークアップの試み

守岡 知彦

概 要

XML に基づいて文書の論理構造と視覚的構造の双方を表現可能な電子テキストを実現する試みを紹介する。文書の論理構造のマークアップは従来よりさまざまな形式でなされてきたが、視覚的構造のマークアップはこれまであまりなされてこなかったといえる。しかしながら、古典籍や石刻のデータベース化を鑑みれば、文書の論理構造と視覚的構造の双方を表現することが望ましいといえる。そこで、文書の論理構造の記述に TEI, 視覚的構造の記述に SVG を用いた多面的マークアップ手法を提案する。

1 はじめに

文字符号は通信用の符号をその起源とし、本来、少ない情報量で文字に関する必要最低限の情報を伝えるためのものであったといえる。文字符号は、本来、視覚的情報を包含していなかったが、パーソナル・コンピューターなどの普及により、視覚的情報を伝えるものとしても使われるようになった。このことから、漢字においては多数の異体字を符号として区別することが要請されることとなり、『漢字が足りない』という問題が起こったといえる。やがて、検索や再利用やインターネットでの情報交換の利便性が認識されると、文字符号は文字の視覚的属性を含むさまざまな属性を背負わされることとなったといえる。そして、文字符号に期待する役割は利用者や分野によって微妙に異なり、文字符号制定の場は相反する利害対立を調整するための政治的な場となり、その結果である文字符号は妥協の産物となることを余儀なくされている。こうした問題は文字符号で全てを表現しようという問題設定によって生じているといえる。つまり、少ない情報量で文字に関する必要最低限の情報を伝えるための仕組を機械的に拡張して、少ない情報量で文字に関する多数の情報を伝える仕組として使おうとしたことに無理があったということである。

このような問題を解決するためには、用途毎に適した表現を用いることが考えられる。例えば、中に書かれた内容に対する検索を行うことが目的ならば、異体字はなるべく正規化すべきだといえる。もちろん、異体字データベースを構築して検索時に対処することはできるが、検索のコストが増えることもあるし、どこまで異体字を分離するかの基準が入力する人によって異なったために異体字表現の品質を整えるためにコストがかかるということもある。逆に視覚的情報に意味がありそれを忠実に表現することが目的だとすれば、画像を利用する方が適切であるといえる。この場合、そのままでは検索ができないので目的に応じて適切なメタデータを付けるなどの工夫が必要である。

しかしながら、電子テキストの利便性のひとつが再利用性であり、本来とは違った目的のデータに転用することを考えれば、特定用途専用というのもまた問題である。このようなことを考えれば、それぞれの用途に適した表現が可能であることと、そうした各表現を統合する枠組の存在が重要であるといえる。このような枠組として有望なもののひとつが XML (eXtensible Markup Language) [7] である。XML では用途に応じたタグセットを定義することができ、実際に各種分野用にさまざまなタグセットが提案され利用されている。人文科学で扱うさまざまなテキストに対

しても TEI (Text Encoding Initiative) コンソーシアム [6] がタグセットに関するガイドライン [5] を作成している。XML は本来文字で書かれた文書のマークアップを想定していたが、マークアップする対象は（文脈自由文法で記述可能な）なんらかの構造を持ったデータならば何でも良く、画像に対しても SVG (Scalable Vector Graphics) [8] というタグセットが制定されている。

SVG は Postscript や PDF の XML 版といったものであるが、他のタグセットと共同で運用することが考慮されており、SVG 用プラグインを利用したり SVG 対応 WWW ブラウザーを利用することによって XHTML などに埋め込んで利用することが可能となっている。よって、テキストの論理構造を TEI などで表現し視覚的構造を SVG で表現することが可能で、内容の論理構造と視覚的構造の双方の側面をそれぞれに適した形で表現しつつ、両者を統合することが可能となる。そこで我々は SVG を用いた文字画像のマークアップ手法を試みている。

2　文書の視覚的構造のマークアップ

PDF や SVG は埋め込みフォント技術や透明フォント技術を用いることにより、符号化文字列を用いてテキスト内容を表現しながら文字のレイアウトや字形を比較的自由に表現することができる。このことは符号化文字列の持つ検索可能性や再利用性と画像の持つ表示の自由さが両立可能であるということを意味し、実際、これまで SVG を使って中国古典テキストを版本に近い形で表示する CGI [11] や PDF に基づくフォント埋め込み技術を用いて異体字形を表示する手法 [10] といった手法が提案されており、我々も SVG を用いた検索可能な文字画像 [12] （図 1）という手法を提案している。

こうした手法はプレゼンテーションや文書公開のための手法としても確かに有効であるが、文書のデータベース化という観点からいえば文書の持つ視覚的構造を記述するということがより本質的な問題であると考える。

SGML や XML を用いてマークアップする場合、マークアップされたテキストをさまざまな形に加工して使うことが可能である。そこで多目的な用途で十分に利用可能な形式をマスターデータとして用いることが重要であるといえる。文書の論理構造に関しては TEI をはじめとしてさまざまな形式が提案され実際に実践されてきたが、レイアウトや字形などの視覚的情報に関してはそのような試みはあまりなされてこなかったといえる。しかし、古典籍や石刻のデータベース化を考えると、こうした視覚的情報も文書の内容やその論理構造と同様に重要な情報源であり、従来は電子テキストでは不十分な情報を画像表示によって補うことが多かったといえる。しかしながら、単なる画像では計算機がその構造を理解することはできず、画像の内容や構造を利用した検索も難しい。そして、文書の論理構造と画像の対応付けは利用者の頭の中で行う必要があるといえる。

我々が提案する文書の視覚的構造のマークアップ手法はこうした問題を解決するものである。これは、文書内容の論理構造と同様に文書の視覚的構造を想定し、その論理構造をマークアップする方法である。例えば、画像情報のプリミティブとしてピクセル・データやベクター・データを捉え、その上部構造として文字を捉え、文字の上部構造として行を捉え、行の上部構造として欄や柱を捉え、これらの上部構造として頁を捉え、頁の上部構造として本を捉えるといった視覚的構造に関する階層構造が考えられる。そして、文字や行や頁といった視覚的構造上の要素と文字や文や章といった論理構造上の要素との対応関係を記述すれば、多面的な文書記述が可能になると考えられる。

そこで我々は文書の視覚的構造の記述に SVG, 文書内容の論理構造の記述に TEI を用いることにより、「説文解字繫傳」を対象に実際にこうしたマークアップ・テキストを試みた。

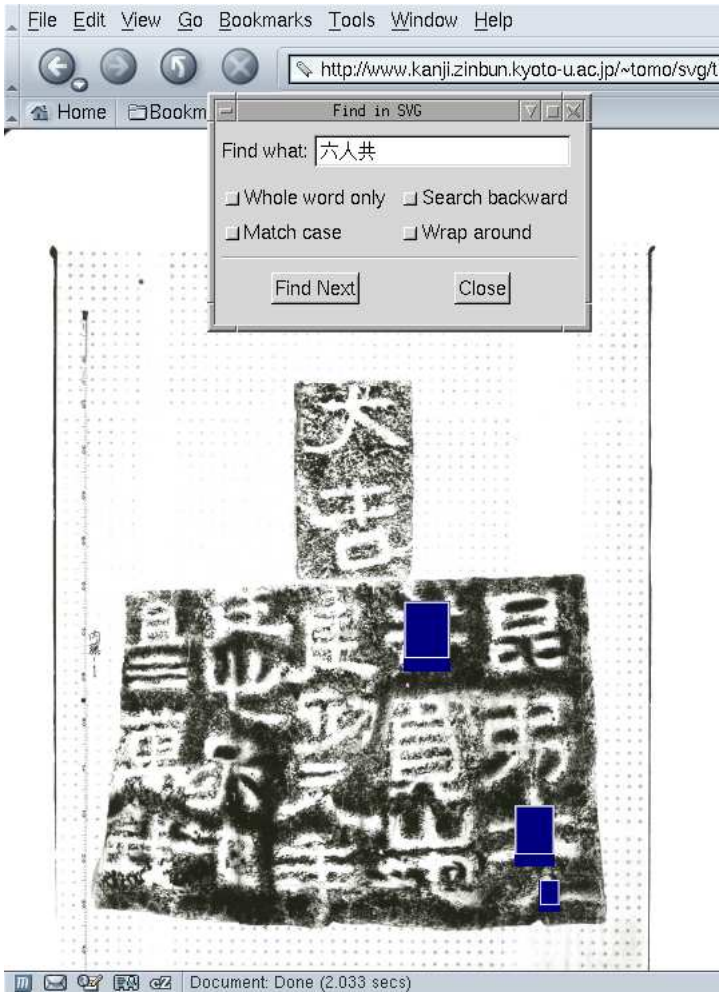

図 1: SVG を用いた文字画像の例

2.1 解決すべき課題

大量のテキストデータベースを実現するためには OCR の利用をはじめとする機械化・自動化と適切なワークフローの確立が欠かせないといえる。この際、OCR から文字情報だけでなくレイアウト情報も同時に取り出せるかということと、SVG テキストの効率的な編集環境をどのように用意するかということが問題となる。しかしながら、現状ではその双方とも十分とは言い難く、何らかの工夫を行う必要がある。

2.1.1 OCR

現在、多くの OCR ソフトウェアが市販されているが、こうしたものの目的は文書内容を符号化文字列として取り出すことであり、文字のレイアウト情報を取り出すことではない。しかしながら、ここでは文書内容の情報と同時に各文字の配置情報（座標や大きさ・範囲など）が必要である。

この問題は松下電器製「読取革命 Ver.7」を用いることにより一応の解決をみた。このソフトウェアは認識結果を「PDF（透明文字）」と呼ばれる形式で保存する機能を持っているが、この機能で作成された PDF ファイルには各文字の座標情報が記録されている（図 2）。

```
%PDF-1.2

1 0 obj
        <<
                /Length 2 0 R
        >>
stream
q 107.5 0 0 -70.0 0 70 cm /Im1 Do Q
BT
        /F11 2 Tf
        0.0 Tc
        3 Tr
        1 0 0 1 53 51 Tm
        0 0 0 rg
        /F11 2 Tf
        <92CA> Tj
ET
BT
        /F11 2 Tf
        0.0 Tc
        3 Tr
        1 0 0 1 53 48 Tm
        0 0 0 rg
        /F11 2 Tf
        <904A> Tj
ET
```

図 2: 「透明文字 PDF」の例（部分）

この PDF ファイルでも大きさや範囲などの情報は取れないが、隣接する文字の縦方向の座標の差を取り出すことにより、こうした情報を補完することは可能である。

2.1.2 SVG テキストの編集環境

以前は SVG を読み書き可能なグラフィック・エディタがそもそも存在しなかったが、最近は、Illustrator 10 などの商用ソフトウェアでも SVG がサポートされており、自由ソフトウェアでも SVG を編集可能なエディタが登場してきている。しかしながら、こうしたものは基本的に画像を

編集するものであって、文書をマークアップするためのソフトウェアではない。

視覚的構造のマークアップという観点では、各構成要素を図形として操作する通常のグラフィック・エディタ的機能に加えて、各構成要素の階層構造を簡潔に表示した状態で編集する機能や、中に含まれる文字内容を編集する機能も必要である。また、文書内容の論理構造との対応を取るためには適切に ID を振ったり、SVG 以外の属性やタグを保存する必要がある。また、Unicode で表現された文字を正しく処理・表示できることも重要である。

残念ながら、現状ではこうした条件を全て満たすシステムは存在していないようである。こうした中で Sodipodi [4] （図 3）は比較的条件を満たしているので、我々は SVG 編集作業において主にこのソフトウェアを用いることにした。

Sodipodi は X Window System を利用する UNIX 系 OS 環境で動作可能な SVG エディタである。システムは C で記述されており Gtk+ やその他いくつかの GNOME 関連ライブラリを利用している。こうしたことから GNOME 環境に対する親和性が高いと考えられるが、GNOME 環境以外でも動作可能である。Sodipodi は GNU General Public License, Version 2 の元で配布される自由ソフトウェアであり、我々のプロジェクトにも適している。

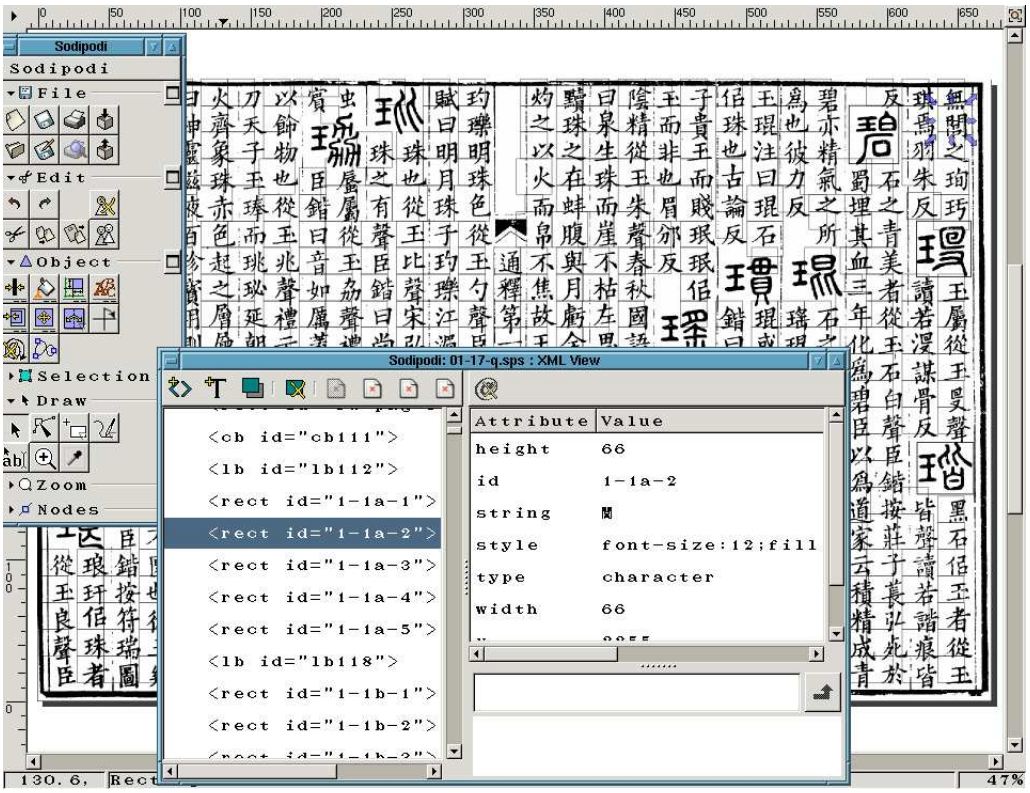

図 3: Sodipodi

Sodipodi は通常の線画編集機能に加え、簡単な XML エディタを備えており、現在の最新版では少なくとも ISO/IEC 10646 BMP [2] の範囲の文字を正しく表示することができるようである。また、それ以外の範囲の文字は表示できないもののデータとしては保持できるようである。また、Sodipodi は SVG 以外の属性や入れ子状でないタグを保持することができ、我々のプロジェクトでの必要条件を満たしている。そこで、我々は SVG 編集作業に Sodipodi を用いることにし、標準の SVG タグセットに TEI と連係するための付加情報を加えた形式を Sodipodi で編集可能な形で定義した。これを『編集用 SVG 形式』（あるいは『SPS 形式』）と呼ぶ（図 4）。

```
<cb
   n="1"
   id="cb111" />
<lb
   n="1a"
   id="lb112" />
<rect
   style="font-size:12;fill:none;fill-rule:evenodd;stroke:#000000;"
   x="2255"
   y="0"
   width="67"
   height="67"
   id="1-1a-1"
   type="character"
   string="無" />
<rect
   style="font-size:12;fill:none;fill-rule:evenodd;stroke:#000000;"
   x="2255"
   y="67"
   width="66"
   height="66"
   id="1-1a-2"
   type="character"
   string="闇" />
```

図 4: 編集用 SVG 形式の例（部分）

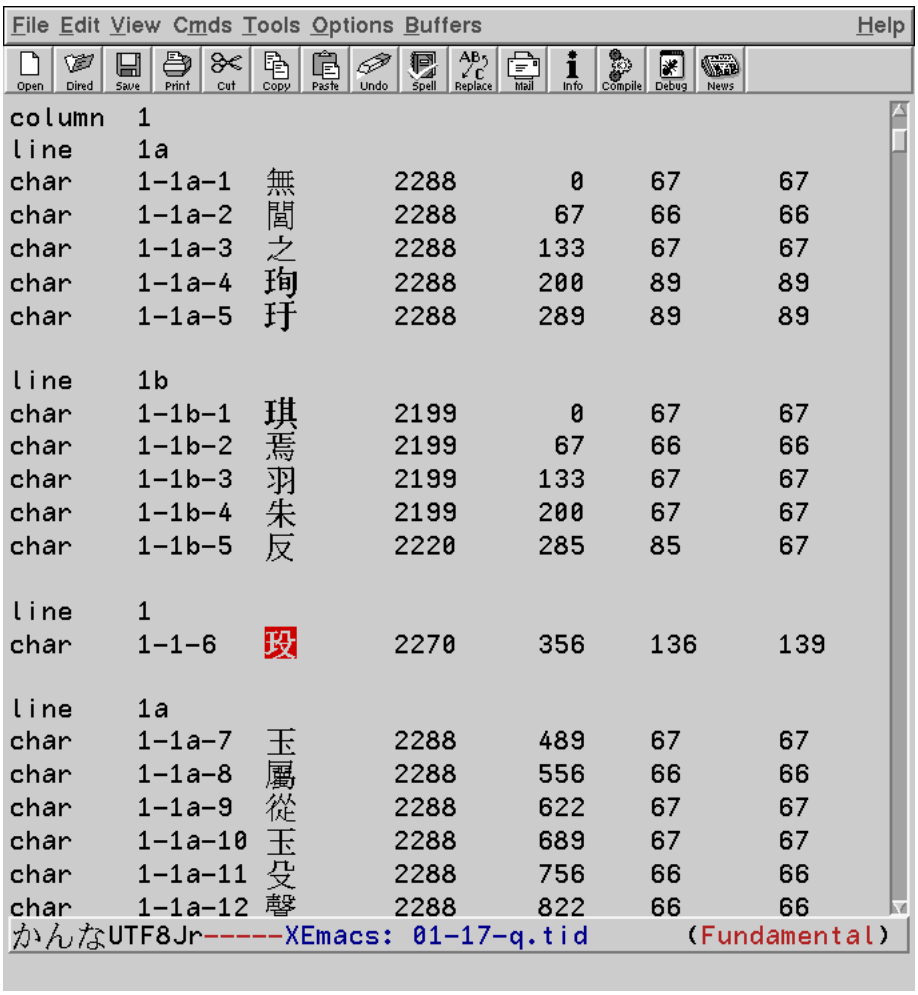

図 5: TID 形式の例（部分）

このように、現状でも Sodipodi は視覚的構造のマークアップという観点で必要とする条件を満たしている。しかしながら、現状の Sodipodi の XML エディタは各構成要素の階層構造や文字列を大規模に編集するには不向きであり、そうした目的にはテキストエディタを用いる方が効率的であるといえる。しかしながら、通常のテキストエディタで編集するには SVG 形式は冗長であるので、この目的のために『TID (Text Image Description) 形式』（図 5）と呼ぶ行指向の形式を定義し、これをテキストエディタでの編集作業に用いるようにした。

2.2 作業の流れ

文書のマークアップ作業は図 6 に示すような手順で行う。

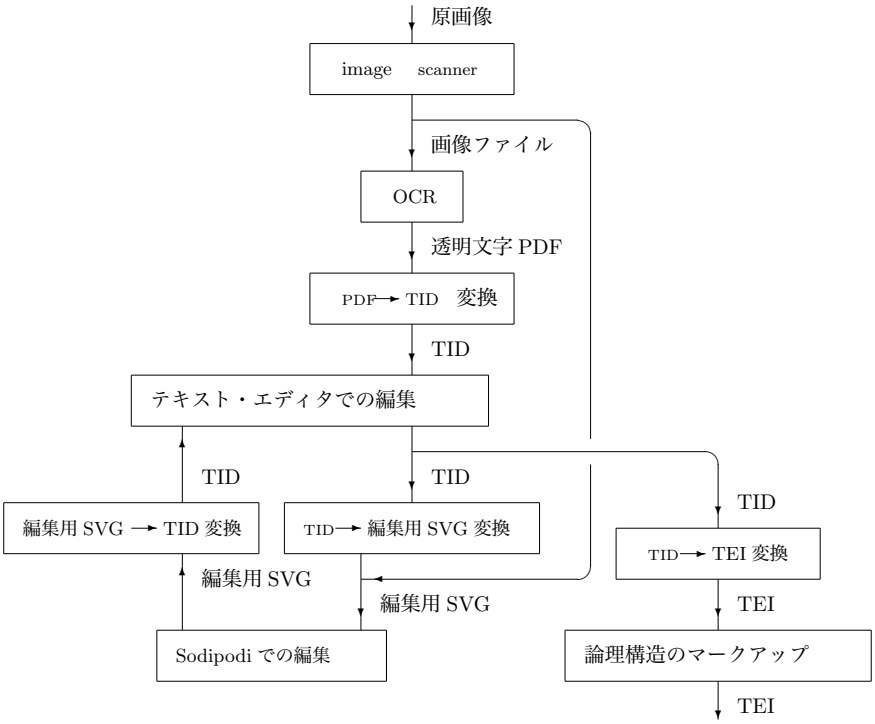

図 6: マークアップ作業の流れ

このワークフローのうち、テキスト・エディタでの編集作業には主に XEmacs CHISE [9]（旧 XEmacs UTF-2000 [1]）を利用することにした。これにより多様な種類の文字が利用可能となる。

一方、論理構造のマークアップ作業は Emacs Lisp で記述された高性能な SGML/XML 編集システムである PSGML [3] を利用することにした。この PSGML は XEmacs CHISE 上で利用することが可能である。そして、XEmacs CHISE では Unicode 外の文字を SGML の実体参照形式で読み書きすることができるので、XEmacs CHISE と PSGML を組み合わせることにより Unicode に含まれる文字も含まれない文字も画面上では区別することなく通常の文字として扱いつつファイル上では SGML/XML 的に適切な形で扱うことが可能である。

また、このワークフローを実現するために必要な

1. PDF → TID 変換

2. 編集用 SVG → TID 変換

3. TID →編集用 SVG 変換

4. TID → TEI 変換

を実現するプログラムを XEmacs CHISE で動作可能な Emacs Lisp プログラムとして実現した。

これにより、画像の読み込み、文字認識、Sodipodi による SVG 画像の編集作業を除く各工程を XEmacs CHISE 上で一貫して行うことが可能となった。

TID 形式から TEI 形式に変換するプログラムでは、TID 形式で記述されている頁・欄・行の情報を TEI 形式に変換すると共に、「説文解字繫傳」に関するヒューリスティックを用いて文字の記述や親字、説解、徐注、反切といった情報をマークアップすることができる。

3　おわりに

XML に基づいて文書の論理構造と視覚的構造の双方を表現可能な電子テキストを実現する試みを紹介した。

SVG を用いることにより、XML の枠組に基づいて画像をマークアップすることが可能であり、特別な文字コードやフォントを使うことなく原テキストにおける視覚的情報を再現可能である。また、テキストの論理構造を記述する他のタグセットと共同運用することにより、論理構造と視覚的構造を有機的に統合することも可能である。

我々は文書の論理構造のマークアップ形式に TEI を採用し、SVG と TEI を併用した多面的マークアップ手法を試みた。こうしたマークアップ・データの量産化を計るために、実際に「説文解字繫傳」を対象にマークアップ作業に関するワークフローを設計すると共に、必要なツールを開発した。

ここでは、TEI と連係可能にするために付加情報を付けた『編集用 SVG 形式』を用いると共に、これと同等の情報を通常のテキストエディタで効率的に編集可能にした『TID (Text Image Description) 形式』を定義し SVG と併用することにした。そして、画像の読み込み、文字認識、Sodipodi による SVG 画像の編集作業を除く各工程を XEmacs CHISE 上で一貫して行うことが可能となった。

謝辞

本プロジェクトの共同研究者である坂内千里氏と Christian Wittern 氏の両氏に感謝する。本プロジェクトは両氏の尽力と助言に負う所が大である。しかしながら、本論文における誤りは全て著者の責であることを明言しておく。

参考文献

[1] bit 別冊「インターネット時代の文字コード」, 第 9 章「文書編集系における文字コード」. 共立出版, 2001.

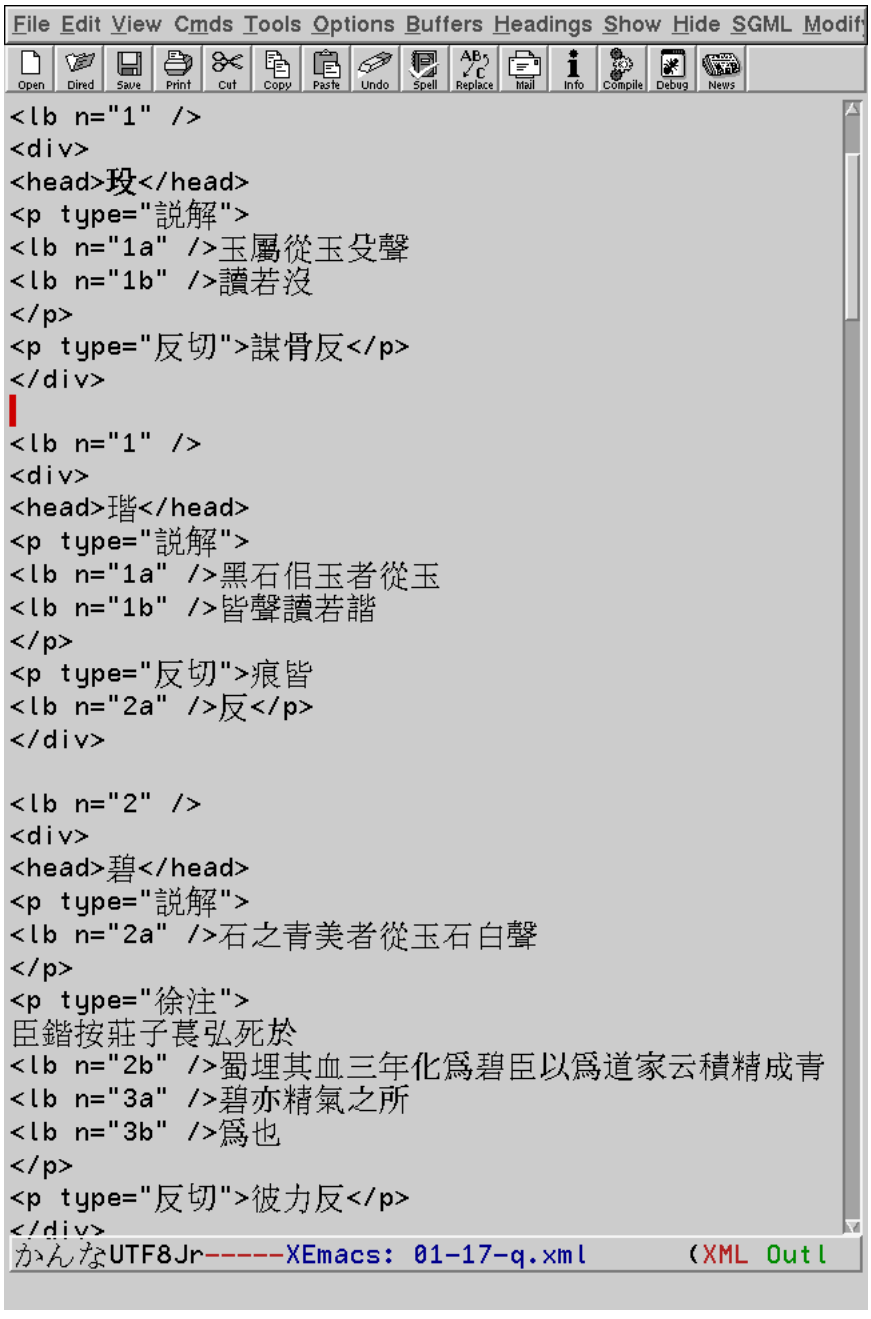

図 7: TID 形式から変換した TEI ファイルの例（部分）

[2] International Organization for Standardization (ISO). *Information technology — Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane (BMP)*, March 2000. ISO/IEC 10646-1:2000.

[3] the GNU Emacs lisp library, PSGML. http://sourceforge.net/projects/psgml/.

[4] Sodipodi. http://sodipodi.sourceforge.net/.

[5] C. M. Sperberg-McQueen and Lou Burnard, editors. *TEI P4: Guidelines for Electronic Text Encoding and Interchange — XML-compatible edition.* University of Oxford, 2002.

[6] The TEI consortium. http://www.tei-c.org/, May.

[7] The World Wide Web Consortium (W3C). *Extensible Markup Language (XML) 1.0 (Second Edition)*, October 2000. http://www.w3c.org/TR/2000/REC-xml-20001006.

[8] The World Wide Web Consortium (W3C). *Scalable Vector Graphics (SVG) 1.0 Specification*, September 2001. http://www.w3.org/TR/SVG/.

[9] XEmacs CHISE. http://kanji.zinbun.kyoto-u.ac.jp/projects/chise/xemacs/.

[10] 安岡孝一. 外字と異体字. アジア情報学のフロンティア — 全国文献・情報センター人文社会学学術セミナーシリーズ No.10, 全国文献・情報センター人文社会学学術セミナーシリーズ, 第 10 巻, pp. 25–34, 2000.

[11] 上地宏一. 版本 cgi. http://web.sfc.keio.ac.jp/ kamichi/kpl/viewer/hanpon.html.

[12] 守岡知彦. ポスト文字コード時代の文書処理技術に関する展望. データベースの活用と人文社会学 — 全国文献・情報センター人文社会学学術セミナーシリーズ No.12, 全国文献・情報センター人文社会学学術セミナーシリーズ, 第 12 巻, pp. 59–70, 2002.