

# 中国古典文献のための電子テキストの概念モデル

白須裕之\*

京都大学人文科学研究所

## 概要

本稿は中国古典文献(古籍)、特に抄本、版本等のテキストを電子化することを前提に、その概念モデルを提出する。書物には論理的な文書構造と共に物理構造が存在し、更に古籍には現代の書物と異なる物理構造が存在する。このような複数の構造を捉えるために、本稿ではタグのOverlap問題を扱える多構造文書なる概念を提出する。これは Concurrent Markup のデータモデルの上に、複数の木構造を定義できるようにしたものである。更に校勘情報のために必要な概念についても議論する。

## Textual Encoding in Ancient Chinese Printings and Manuscripts

SHIRASU Hiroyuki\*

Institute for Research in Humanities, Kyoto University

### Abstract

This paper investigates multi-structuring and collation issue of textual encoding in ancient Chinese printings and manuscripts. We discuss requirements of digital text repositories for knowledge base on historical sources, and overlapping problems of concurrent markups. So this paper presents a conceptual model of digital texts based on *The Layered Markup and Annotation Language(LMNL)* with XML-based multi-structures. Furthermore we show applicability of this model to textual encoding of ancient Chinese printings and manuscripts.

## 1 はじめに

筆者は漢字文献の知識ベース化を試みるプロジェクトに属し、中国唐代(618-907)についての知識ベースを構築するために、官制、行政地理、人物に関するオブジェクト指向分析/設計を行ってきた(例えば、文献[2]等を参照)。本稿はそれらの基礎となる中国古典文献(古籍)のテキストを電子化するための概念モデルを議論する。

文献[4]は、歴史記述のための知識ベースを構築するという営為を理解するため、知識表現の枠組みの必要性を述べた。本稿ではその中で取り上げた歴史資料のためのテキスト表現である抽象テキストと、文献間の関連情報、書誌情報を表現する枠組みを詳述する。

XMLを使用したテキストエンコーディングでは、文書構造が木構造になっている必要がある。しかし、古籍には文書構造として複数の互いにoverlapした構造が存在し、従来のマークアップのoverlap問題と共通の問題点がある。この問題へのアプローチとしては Concurrent Markup によるもの、マークアップの

オントロジー表現[11]等の先行研究がある。

古籍のテキストを概念モデル化するために、本稿ではこれらの先行研究について議論し、その不十分な点を明かにする。

### 1.1 電子テキストに対する要求

まず古籍を電子テキストとして表現するにあたっての要求仕様を述べる。最初に電子テキストに対するユースケースを述べ、それを詳細化していく。電子テキストのための概念モデルには以下のようなユースケースが存在する。

- 知識ベースにテキストを提供する。
- 校訂テキストを作成する。また、版を指定してテキストを開覧する。
- マークアップテキストを作成する。マークアップテキストは電子テキストのシリアルライズの一種形態である。

\*京都大学 21 世紀 COE 東アジア世界の人文情報学研究教育拠点

\*Toward an Overall Inheritance and Development of Kanji Culture

これらは電子テキストを格納したりポジトリとして、電子テキストサーバーの構築を念頭に置いている。これらの要件を複数ユーザーに矛盾なく、独立に提供できるよう、分散環境で使用できるように設計する必要がある。これについては別稿に譲る。

**知識ベースからの要求** 知識ベースにテキストを提供するという要求について述べる。その詳細については文献 [4] を参照する。

以下のクラス図は文献 [4] で述べた抽象テキストである。「単位」はテキストを構成するための単位であって、文献 [1] で提案されている Chaon モデルの文字のようなものを考えている。

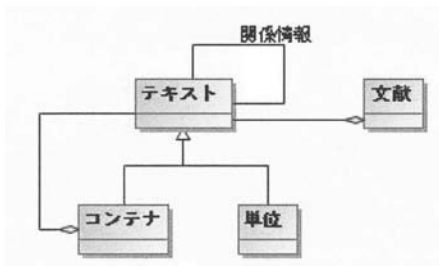


図 1: 抽象テキスト

「テキスト」は「単位」から Composite パターンを用いて構成される。各「テキスト」はテキストの間の引用、翻訳、校訂などの文献学的な関係情報を持つ。「文献」は複数の「テキスト」の構造を保持する。電子テキストは文献学的な関係情報を表現できることが要求される。

**文献学からの要求** 上で述べたユースケース、及び知識ベースからの要求は、テキスト間の文献学的な関係情報を表現することであった。以下がその例である<sup>1</sup>。

- 校訂、版、翻訳、引用、異文等、様々な要求によって、個々のテキスト断片が選択、指定できる
- 校訂の理由を参照できる
- 校訂によるブランチの管理ができる

## 1.2 古籍の基本構成

本節では古籍の文書構造について議論する。電子テキストにおいては以下に示すような様々な文章構

<sup>1</sup>ここで述べた要求仕様は文献 [5] に示唆されたものである。  
<sup>2</sup>文献 [7] 邦訳第三章「古籍の基本構成と校勘の根本原則」参照。  
<sup>3</sup>テキストの書き換え、または異文の割注での挿入で表現される。

造を扱わなければならない。次節以降でこれらの文書構造をどのような扱うかについて議論する。

- 電子化の対象となる文書の媒体上の構造
- 原テキストの標準の文書構造
- 文書の論理的な構造
- 文法的な構造

下図は『元和郡縣圖志』の版本(幾輔叢書)の例である。古籍はそのもともとの媒体によって文書の構造が異なる。また、その古籍を基としたその後の抄本(写本)や版本にも、その文書構造が受け継がれるという特徴を持っている。



図 2: 『元和郡縣圖志』の版本(幾輔叢書)の例

そのような特徴は古籍の基本構成に基づいている。文献 [7] では古籍の基本構成を大まかに二種、複雑な多層構成と単純な多層構成に分類している<sup>2</sup>。

原稿あるいは原版が失われ、年代の隔たりによって、現存の古籍文献には不可避免的に伝承過程で生じた訛誤が含まれ、しかもそれは重層化しています。

これらの重層化は異文で表現され<sup>3</sup>、一般に単純な多層構成を形成すると述べている。

一方、経典的古籍や重要な著作については、歴史的な政治思想の発展変化にともない、様々な解釈や演繹が行われており、異なる流派の学説が形成されています。

こちらは異文としてではなく、解釈の相違として表現され、一般的に複雑な多層構成をなすと述べている。これらの基本構成の分類は校勘情報を電子化する際に考慮しなければならない、重要な点であると考えられる。これについては節5で再考する。

## 2 Concurrent Markup による事例

本節では、版本文書構造を Concurrent Markup によって表現しよう。また、その特徴と短所について議論する。Concurrent Markup と言われるものは数多く存在する。通常は XML を使用して、Concurrent Markup を実現しようとするが、このアプローチの利点は XML のために構築された環境(文書の妥当性のチェック、妥当性を保ちながらの編集作業)が使用できる点があげられる。

### 2.1 LMNL

本稿では Concurrent Markup の特徴と問題点を見るために、記述が簡単である The Layered Markup and Annotation Language(LMNL)[10] を使用する。LMNL は XML 構文ではなく、独自の構文を持つ。

LMNL の構文を簡単に纏めておく。LMNL で基本となる概念は以下である。

- 「文字」(character) — 予約された文字は「{&}」であり、これらはタグを構成する目的で使用するため、通常の文字として使用する場合にはエスケープが必要である。
- 「範囲」(range) — 名前 `name` を持つ「範囲」は、開始タグ `[name]` から、終了タグ `{name}` までの範囲を示す<sup>4</sup>。空タグは `[name]` と書かれる。
- 「注釈」(annotation) — 「注釈」は「範囲」に補足的な情報を付加する。対象となる「範囲」の開始タグ、あるいは終了タグ内に記述される。構文は「範囲」と同様。

<sup>4</sup>タグは overlap する場合もあるので、要素を表現すると言うよりも「範囲」を表現している。

「注釈」は XML の属性と似た働きを持っているが、異なる点は以下である。一つの「範囲」に複数の同じ名前の「注釈」が付加でき、また順序が保たれる。「注釈」に「注釈」を付加することができ、構造化することが可能である。但し、overlap は許されない。LMNL には更に層 (layer) という概念があるが、これについては後ほど説明する。

### 2.2 単純な構成の場合

以下は版本文書構造が単純な場合の典型的な例を簡略表記したものである。この文章はこの書物の巻4、葉16の最初の部分であるとする。小文字の部分が割注である。割注がある場合は二行を古籍の一行であるとして見てほしい。

```
AAAx yyBB
  zz tt
CCCCCCCC
DDDDDDuuu
      uuu
wwwEEEEEE
www
```

これを LMNL でエンコードした例が以下である。二つ目の割注の部分「uuuuuuwwwww」が3行目、4行目に渡っていて、割注のタグと行のタグが overlap していることが分かる。このような単純な例でも、割注が複数行に渡る場合にはタグの overlap が発生する。

```
[!lmln version="0.2"]
[文献]
  [書名]{ }
  [版]{ }
}
[巻=v04]
[葉=p16]
  [表]
  [行]AAA[割注]xx yyzz tt[割注]BB[行]
  [行]CCCCCCCC{行}
  [行]DDDDDD[割注]uuuuuu{行}
  [行]wwwww[割注]EEEEEE{行}
  ...
  [表]
  [裏]...{裏}
{葉=p16}
...
[巻=v04]
{文献}
```

しかし、LMNL でエンコードすることが難しい例もある。割注は一般的には一つの行を二行に分割して、半分の大きさの文字を使用する。通常の割注の読み方は、この二行を一行目の終わりまで読み、次に二行目を読むようになっている。しかし、そのような読み方をしない場合がある。例えば、上の例の一つ目の割注を「xxzz yytt」というように、二つのブロックとして読む場合である<sup>5</sup>。

一行目の割注を物理的な構造 (レイアウト) でエンコードしたものが以下である。

```
[行]AAA[割注]xx yy{割注}
      [割注]zz tt{割注}BB{行}
```

また、一行目の割注を論理的な構造 (読み) でエンコードしたものが以下である。

```
[行]AAA[割注]xx zz{割注}
      [割注]yy tt{割注}BB{行}
```

この二つの割注の構造をどのようにエンコードするのが自然であろうか? また、XML スキーマのような妥当性を検証する仕組みはいかにすべきであろうか?

## 2.3 注疏を含む場合

注本は経典等の基になる古籍を注釈した書物であり、疏本は注をさらに注釈した書物である。宋代以降、本文に注疏の文を割注として合刻した版本が多く見られる。

例えば、以下は注疏を含むテキストの一部を簡略表記したものである。本文は二行で、一行目に「注」があり、その後に「疏」が現われ、二行目の途中まで「疏」が続いている。

```
AAAAA 注 xxxxxx 疏 zzzz
      yyyyyy uuuu
vvvvvvvvvvvvvvvvvvvvvBBBB
wwwwwwwwwwwwwwwwww
```

これを LMNL でマークアップした例が以下である。注の著者は「著者 1」、疏の著者は「著者 2」とし、その時期を各々「唐」「宋」とする。実際には注疏の付加情報が更に必要であろうが、ここでは省略する。

```
[!lmnl version="0.2"]
[文献]
 [書名]{}
```

<sup>5</sup>例えば、唐代の地理誌である『元和郡縣圖志』の各種の版本では、このような割注が頻出する。図 2 を参照。

```
{版}{  
 [注 {時期} 唐{  
 [疏 {時期} 宋{  
 }  
 [巻]  
 [葉]  
 [表]  
 [行]AAAAA [注] [割注]xxxxx {割注}  
           [割注]yyyyy {割注} {注}  
           [疏] [割注]zzzz {割注}  
           [割注]uuuu {割注} {行}  
 [行] [割注]vvvvvvvvvvvvvvvvvv {割注}  
 [割注]wwwwwwwwwwwwwwwwww {割注} [疏]  
 BBBB {行}  
 ...  
 {表}  
 ...  
 [巻]  
 [文献]
```

## 3 マークアップ言語のデータモデル

本節ではマークアップ言語、XML と LMNL のデータモデルについて述べる。LMNL のデータモデルは後の節で電子テキストの概念モデルを設計する際の検討材料とする。また、XML のデータモデルは電子テキストの文書構造を定義する際に使用する。

### 3.1 XML のデータモデル

XML のデータモデルとしては多くのものが存在するが、ここでは XDM (XQuery 1.0 and XPath 2.0 Data Model) を取り上げる [9]。

以下のクラス図は XDM の概要を示したものである。

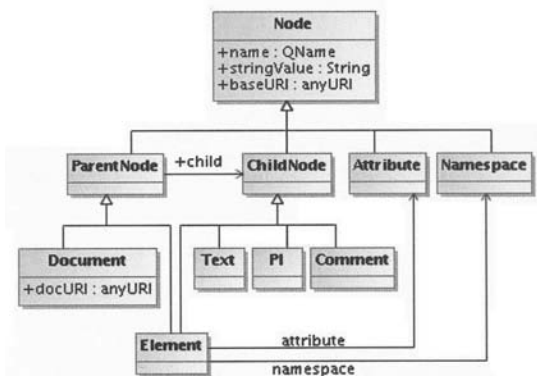


図 3: XDM

このクラス図について XDM の概要を述べる。

- XDM のインスタンスは本質的に Node が作る木である。Node は Document, Element, Attribute, Namespace, Text, ProcessingInstruction(PI), Comment を含む。
- 枝のラベルには三種、child, attribute, namespace がある。各々はどの枝を子として持てるかが決っている。
- Element は attribute と namespace の枝を持つ。
- クラス ParentNode と ChildNode は枝 child の両端として可能なクラスを、サブクラスとして持つ。即ち、Document は枝 child の親ノードにしかならず、Element は親ノード、子ノードの両方になれる。

言わば XPath, XQuery はこの XDM のインスタンスを走査できるように設計されていると言える。

XDM のインスタンスは色々な方法で構成されるが、ここでは文書から構成する方法を簡単に見てみよう。これは与えられた文書からパーザによって、文書に付与されたタグによるネスト構造を、XDM のインスタンスが作る木構造に変換する。例えば、ある要素 a がある要素の中にネストしていれば、b は a の子ノードになる。以下同様。

このように要素のネスト構造が文書の木構造を自然に表現している点が、XML 文書の特徴である。しかし、この特徴が逆に、古籍等の複雑な文書構造を持つテキストの電子化に XML を使用することを難しくしている。

### 3.2 LMNL のデータモデル

本節では Concurrent Markup のモデルとして、LMNL のデータモデルを取り上げ、古籍の電子化のために必要な機能について議論する。

文献 [10] の "LMNL data model" の内容を、本稿の必要な範囲で以下に纏める。文字、名前空間、局所名、拡張名はいずれも XML からの借用である。このモデルは Layer, Range, Annotation を基本概念としている<sup>6</sup>。その基本構成をまとめたものが、以下のクラス図である。

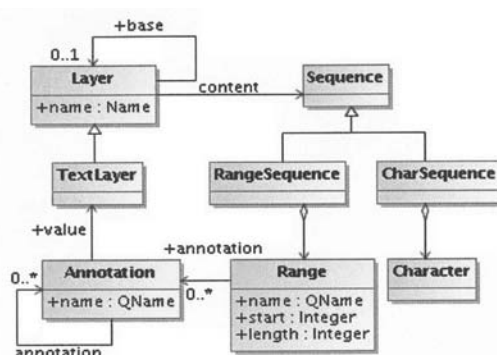


図 4: LMNL のデータモデル

クラス図を作成するにあたって、三つの基本概念に対して、TextLayer, Sequence 等のクラスを追加した。このクラス図の情報から導出される属性値については、クラス図から省略してある。

- Layer — Layer の保持する内容は、Range の Sequence か、Character の Sequence かのどちらかである。これは属性 content の値になる。また、基礎となる Layer を一つ持つことができる。これは属性 base で示される。
- TextLayer — base を持たない Layer として TextLayer を設けた。以下に説明するように、context に RangeSequence を持つためには base が必要である。
- Range — Range は Layer に含まれる Sequence 上の範囲を示す。これは構文的にはマークアップによって表現される「範囲」である。
- Annotation — Annotation は名前と値の対である。値は TextLayer の内容になる。その owner は属性 annotation の親である。
- Sequence — Range か Character どちらかの列を示す。Sequence のサブクラスとして、RangeSequence と CharSequence を設けた。

Layer の属性 content は Sequence であるが、その Sequence に Range が含まれている場合、Range の属性 owner layer の値はその Layer となる。

Range の保持する内容は、この owner layer の base から構成される。その様子を示したのが以下の図である。Range r の owner layer を Layer l1 とする。ま

<sup>6</sup>ここではクラスをマークアップの構成要素と区別するために英字を使用した。

た、 $l_1$  の base が Layer  $l_2$  である。 $r$  の属性  $start$  と  $length$  の値は、 $l_2$  の content である Sequence の最初の位置とその長さを示している。従って、 $r$  の保持する内容はこの Sequence の部分列になっている。

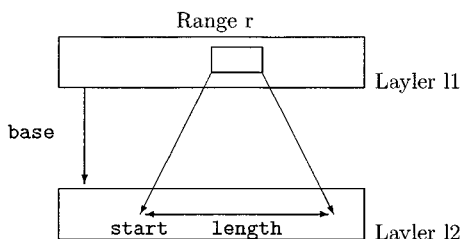


図 5: Range の内容

以下で LMNL のデータモデルとその構文についての問題点を纏めておく。次節でこれらについての解決を試みる。

- Layer の content は Sequence になっているので、その順序を自由に指定できる構文が望ましい。しかし、実際の LMNL の構文では、テキストに出現する順序しか指定できない。
- データモデルのインスタンスにおいて、複数の Range が Layer の content を通して、階層構造を構成しているが、その構造を構文から読み取ることが難しい。

## 4 電子テキストの概念モデル

前節で見たように多くの文書構造を持つ古籍を電子テキストとするためには、LMNL のモデルが持つ構造では不十分であることが分った。本節では LMNL のデータモデルは Range を定義するものとしてだけ扱い、文書構造は別に定義する方法によって電子テキストの概念モデルを構築する。まず本稿で使用する多構造文書の定義を述べる。

**多構造文書の文書定義** 多構造文書の文書定義は  $S = \langle R, \{S_i\}_{i \in I} \rangle$  である。但し、

1.  $R$  は LMNL の Range の名前名の集合である。
2.  $I$  を適当な有限加算集合とすると、 $S_i (i \in I)$  は XML スキーマ (本稿では RELAX NG[8] を使用する) である。但し、以下のような  $R$  による要素の定義を追加する。

$S_i$  には以下の定義を追加して良いものとする。ここで RANGE\_NAME は  $R$  からの Range の名前とする。

```
<element name="RANGE_NAME">
  <attribute name="ref">
    <text/>
  </attribute>
  ...
</element>
```

各  $i \in I$  に対して  $d_i$  が XML 文書であり、 $L$  が LMNL 文書であるとき、 $d = \langle L, \{d_i\}_{i \in I} \rangle$  を多構造文書と呼ぶ。文書定義  $S$  に対して文書  $d$  が妥当であるとは、以下が成り立つことである。

1. 各  $i \in I$  に対して、 $d_i$  が  $S_i$  で妥当である。
2.  $L$  に使用されている Range の名前が、全て  $R$  に含まれる。

但し、LMNL 文書には以下のような要素を使用することができる。ここで ID は LMNL 文書の identifier である。

```
<RANGE_NAME ref="ID"/>
```

多構造文書を以上のように定義するため、多構造文書のモデル (即ち、電子テキストのモデル) は XDM と LMNL のデータモデルを組み合わせたものになる。ここでクラス Range は XDM のクラス Element のサブクラスとする。

多構造文書のモデルでは文書構造を XML 文書で定義するため、問合せ言語として、Range の部分のみ拡張することによって XQuery を使用できる。

**図 2 のエンコーディングの例** 図 2 の『元和郡縣圖志』の一部の内容を多構造文書としてエンコーディングした例を、附録 A に述べた。A.1 は図 2 のテキストに LMNL でマークアップしたものである。多構造文書としては、二つの文書構造の例を述べた。但し、構造定義は RelaxNG のコンパクト構文を使用した。A.2.1 は物理的構造、A.2.2 は論理的構造の例である。「道」「州」「県」の地理区分に従った上下関係による論理的構造<sup>7</sup>、及び節 2.2 で問題提起した「割注」の論理的な読み方が示されている。

<sup>7</sup>唐代行政地理の語彙については、例えば文献 [2] を参照する。

## 5 校勘情報の記述に向けて

校勘情報として文字の異同を考えてみよう。写本やバージョンの情報をテキストに移す場合には、通常文字コードをもって行なう。これは一つの解釈であり、画像の位置情報と文字との対応関係等で記述する必要がある。また、異同を記述するためには、文字の包摂の問題及び校勘における文字の差違とは何かということを厳密にモデル化する必要がある。

文献 [7] の第七章「出校の原則と校記の書き方」には以下のようにある。

校勘の成果を整理して表現することは、一条一条単純に書き出せばよいというものではなく、分析・帰納を行って、出校は精選され、校記は精要で、叙例は簡明という条件を満たすようにしなければなりません。

この原則を電子テキストに当て嵌めることができるかという問題、及びこのような校勘情報の表現をどのようにするかということは、今後の研究を待たなければならない。これについては別稿に譲る。

## 6 おわりに

本稿は LMNL のデータモデル・インスタンスに多くの文書構造を付加する方法によって、古籍の電子テキストをモデル化することを試みた。また、多構造文書に必要な要素について検討した。但し、構文については更なる検討が必要である<sup>8</sup>。

文献 [3] では版本どうしの関係、例えば底本、参照等をモデル化した。しかし、校勘情報としては、文書のテキスト断片同志の関係としてモデル化する必要がある。本稿で述べた電子テキストの概念モデルは、このようなテキスト断片同志の関係を記述するための基礎になるであろう。今後の課題としたい。

**謝辞** 本稿の問題提起は文献 [5] に示唆されました。原稿を提供いただいた永崎研宣さんに感謝いたします。本稿の内容の多くは牛根靖裕さん、山田崇仁さんとの共同研究から生まれました。御二方に深い感謝の念を捧げます。本稿を書くにあたり以下の方々にお世話になりました。深く感謝いたします。安岡孝一さん、永田知之さんには唐代知識ベースのプロジェクトにおいてお世話になっております。守岡知彦さん、秋山陽一郎さんには日頃より人文学、計算機科学

<sup>8</sup>例えば、LMNL 文書からデータモデル・インスタンスを作成するために要する計算量について、著者はまだ理解していない。

についての議論を通して多くの影響を受けております。最後に妻留美と新たな家族に感謝します。

## 参考文献

- [1] 秋山陽一郎, 守岡知彦, 浦田衣里: 階層的素性名を用いた異体字記述の試み, 情報処理学会研究報告, 2005-CH-67, 2005.
- [2] 牛根靖裕, 白須裕之, 山田崇仁: 唐代行政地理の概念モデル, 情報処理学会研究報告, 2007-CH-73, 2007.
- [3] 牛根靖裕, 白須裕之, 山田崇仁: 唐代資料引用のための語彙設計, 情報処理学会研究報告, 2007-CH-76, 2007.
- [4] 白須裕之: 歴史記述に対する概念分析の試み, 情報処理学会研究報告, 2007-CH-74, 2007.
- [5] 永崎研宣: 要素間の関連情報を基盤とする仏教文献デジタル・アーカイブの可能性, 情報処理学会研究報告, 2007-CH-75, 2007.
- [6] 陳国慶: 古籍版本浅説, 遼寧人民出版社, 1957. (邦訳 沢谷昭次訳: 漢籍版本入門, 研文出版, 1984.)
- [7] 倪其心: 校勘学大綱, 北京大学出版社, 1987. (邦訳 橋本秀美, 鈴木かおり訳: 校勘学講義, アルヒーフ, 2003.)
- [8] J. Clark, M. Makoto, ed.: RELAX NG Specification, OASIS, 2001.
- [9] M. Fernández, et.al.: XQuery 1.0 and XPath 2.0 Data Model(XDM), W3C Recommendation, <http://www.w3.org/TR/xpath-datamodel>, 2007.
- [10] LMNL.org: The Layered Markup and Annotation Language, <http://www.lmnl.net/index.html>, 2002.
- [11] G. Tummarello, C. Morbidoni, E. Pierazzo: Toward Textual Encoding based on RDF, Proceedings ELPUB2005 Conference on Electronic Publishing, 2005.

## A 附録：多構造文書の例

### A.1 LMNL 文書の例

```
[!lmln version="0.2"]
[文献=yuanhezhi
[title]元和郡縣圖志[]
[版]幾輔叢書[]
]
...
[卷=v04
[title]元和郡縣圖志卷第四[]
]
...
[葉=p16]
[表=a16]
...
[行=03][果=01]銀城縣{果}
[comment=c01]
[割注]中下北至{割注}[割注]州四十里 {割注}
[comment]本漢固陰縣地屬西河郡漢末大亂{行}
...
[行=07]屬麟州{行}
[行=08][州=z02]勝州{州}
[comment=c02]
[割注][text=t01]榆林{text} [text=t02]開元戶四千九十五 鄉十三{text}{割注}
[割注][text=t03]下府{text} [text=t04]元和戶{text}{割注}
[comment]{行}
...
{表}
...
[卷=v04]
...
{文献}
```

### A.2 構造定義の例

#### A.2.1 物理的な文書構造の例

```
default namespace =
"http://coe21.zinbun.kyoto-u.ac.jp/msd/physical"
start = element 文献 { rangeRef, 卷+ }
rangeRef = attribute ref { text }
卷 = element 卷 { rangeRef, 葉+ }
葉 =
  element 葉 {
    rangeRef,
    element 表 { rangeRef, lines },
    element 裏 { rangeRef, lines }
  }
lines = element 行 { rangeRef }+
```

```
<?xml version="1.0"?>
<文献
  xmlns=
    "http://coe21.zinbun.kyoto-u.ac.jp/msd/physical"
  ref="yuanhezhi">
  <卷 ref="v04">
    <葉 ref="p16">
      <表 ref="a16">
        ...
```

```
<行 ref="103"/>
...
<行 ref="107"/>
<行 ref="108"/>
...
</表>
<裏 ref="...">
...
</裏>
</葉>
</卷>
</文献>
```

#### A.2.2 論理的な文章構造の例

```
default namespace =
"http://coe21.zinbun.kyoto-u.ac.jp/msd/logical"
start = element 文献 { rangeRef, 道+ }
rangeRef = attribute ref { text }
道 = element 道 { rangeRef, comment?, 卷+ }
卷 = element 卷 { rangeRef, 州* }
州 =
  element 州 {
    rangeRef,
    comment?,
    element 県 { rangeRef, comment? }*
  }
comment =
  element comment {
    rangeRef,
    element text { rangeRef }*
  }
```

```
<?xml version="1.0"?>
<文献
  xmlns=
    "http://coe21.zinbun.kyoto-u.ac.jp/msd/logical"
  ref="yuanhezhi">
  <道 ref="...">
    <卷 ref="v04">
      <州 ref="z01">
        <県 ref="j01">
          <comment ref="c01"/>
        </県>
      </州>
      <州 ref="z02">
        <comment ref="c02">
          <text ref="t01"/>
          <text ref="t03"/>
          <text ref="t02"/>
          <text ref="t04"/>
        </comment>
      </州>
    </卷>
  </道>
</文献>
```