日本語教育のための表現意図出現パターン調査における文書データベースとＸＭＬの活用

中尾桂子，森下淳也
神戸大学大学院総合人間科学研究科
EMAIL:{nakao@ccs2000.cla.jm@}kobe-u.ac.jp

あらまし:小学校教科書をXML文書として記述することにより，その文中に出現する指示等の意図をパターンとして検出する．各文の表現意図は学習目標への道標として捉えられる．その手段として用いたXMLと関係データベースとの対比を議論する．
キーワード: 日本語教育,文章構造パターン,系列検索,テキストデータベース,XML

Text database for textbooks of primary school to probe the sequential patterns of the instructical expressions and it's analysis with XML.

K. NAKAO, J. MORISITA
Graduate School of Cultural Studies and Human Science,Kobe University

Summary: We probe the sequential patterns of instructical expressions appeared in textbooks of Japanese primary school. Instructical expressions in sentence will be an important signature towards understanding the learning issues, especially for foreigner's pupils in Japan. We also discuss the analysis with XML and compare with the case of XML and of relational database.

1. はじめに

ここ10年の間，外国人児童・生徒の編入により，学校現場は混乱してきた．

混乱の原因は，外国人児童の教育が，外国語教育，言語教育，基礎教育といった３種類の異なった教育がからみ合う複雑さによる．

昨今，国語教育へつなげるための日本語教育を解明するという目的で，学習用言語の解明を目指す語彙調査が進んでいる[1-3]．これは，児童の基礎日本語能力の充実をはかる上で有益な基礎研究である．

しかし，外国人児童への日本語の指導期間の短さや進学を考えて効率化とプラクティカルな面との関連を考慮するなら，即物的ではあるが，情報収集スキルの指導が必要である．

そこで，外国人児童への「教科学習につなげる日本語教育」における基礎研究の一環として，小学教科書の構成パターンを調べる．これにより，情報収集攻略法の指導方法の可能性をさぐる．

本の中のテキストをデータとして扱う場合に，その文章構造を考慮し，階層構造とテキストデータの関係を位置により関連づけたい．

過去に行った小学校教科書の文型調査を目的とする研究においては，テキスト或いはRDBMSを用いて解析を行なってきた[4]．RDBMSを利用すると，データの保全性が高く，同列に扱う集合に対する処理が効率良く実行できるためである．しかし，本論文の対象である階層構造と各データの関係付けにはコストがかかる．

階層構造を表現する汎用のデータ形式として，インターネットの普及から盛んに扱われるようになってきたものに，XML(eXtensible Mark-up Language)がある[5]．XMLはデータにタグ付けを行なうことで階層化し，データの持つ上下関係や順序関係を木構造で表現するデータ形式である．これは，汎用のデータ形式としてその構造の解析や変換を行なうソフトウェアが充実しているため，階層に依存する探索や順序関係を抽出するのに適している．

そこで，教科書構成の指標となる表現意図の出現パターンを調査するにあたり，小学校教科書を

XMLを用いて表現することにした．この表現意図出現パターン調査とは，教科書の階層の中での特定の意図を持つ文の出現順序を調べるものであり，XMLはその目的にかなったデータ形式であると考えた．

調査は，以下の手順で行なった．

(1)電子データ化した小学校教科書の文章に章節などの文章構成上の階層情報を付加する．その際，教科書の二次元的な広がりを表現するために教科書特有の構成単位として「ブロック」を定義することで，複数のストリームを文書構造の階層化に位置付けることにした．

(2)文章の内容毎のまとまりであるブロックに対してその末文の表現意図の性質の違いにより意味付けを行う．

(3)意味のまとまりで構造に基づき，出現パターンを調べる．

2章で教科書の構造とその文章とりあげ，表現意図とそのパターン化について述べる．3章でXML化文書データとその解析方法，並びに，木構造内での部分木検索について述べ，4章で文書データの取り扱いについて考察する．

2. 学習活動における意図

教科書の文章は学習活動の流れに基づいた事実や実験，それらのまとめなどで構成されている．教科書では，目標とする学習内容へと学習活動を通じて学習者を導くために，指示や注意等の表現形態を持つ文が随所にあらわれ，学習活動の流れにおける始まりや終わりといったなんらかの意図が表現されている．

本章では，出現パターン検索の手掛かりとなるこれらの表現と教科書やその文章の構成との関係について述べる．

2.1 教科書の構造

教科，学年の違いは多少あるが，小学校の教科書は文書であることから基本的には章節構造順に並んでいる．しかし，教科書は学習目的に応じた学習活動の手引きや資料といった役割を担う面があり，見やすさ，ポイントのフォーカスに配慮した構成となっている．このため，教科書の構造は二次元的な広がりを持つ．これを考慮して教科書テキストの構造を規定する．

2.2 ブロック

教科にもよるが，基本的に小説等と同様に教科書にも談話にストリームがある．しかし，実際には，各教科の１単元に相当する章の中には，本文ブロック，囲み記事ブロックや写真説明ブロックなどが並列されており，異なるストリームのまとまりが複数並べられた配置をとっている．小説等とは異なった特徴的な文書構成を持つ．

内容の異なる別々のストリームと見られるひとまとまり毎の文章は，通常の小説等に見られる章，節，部，段落という文書構造木にはない単位である．この単位を考慮した教科書文書の構成を定義するため，この物理的なまとまりは，教科書の章節以下の構成単位の１つであるとして，「ブロック」と名付ける．

図1にブロック毎に分かれた構成の一例を挙げる．図1は，見開きで見た小学校教科書が，視覚的にも，内容的にもブロックにまとまっていることを表した概念図である．

2.3 ブロックの性質

ブロックは内容毎にまとめられた１つのストリームであり，視覚的な単位である．このブロックを構成する文章は，同じ文体で同様の文末形態を持つ．しかし，中には，ブロックの最後の文だけが異なった文末形態を持つ場合がある．このような文の文末には，益岡のいう「伝達の態度」や「主観性」があるなんらかのモダリティが見られる[6]．

意図を含む文を末尾に持つブロック内では，最終文に至るのまでの文は，最終文の根拠を説明するという流れになっている．つまり，ブロックの

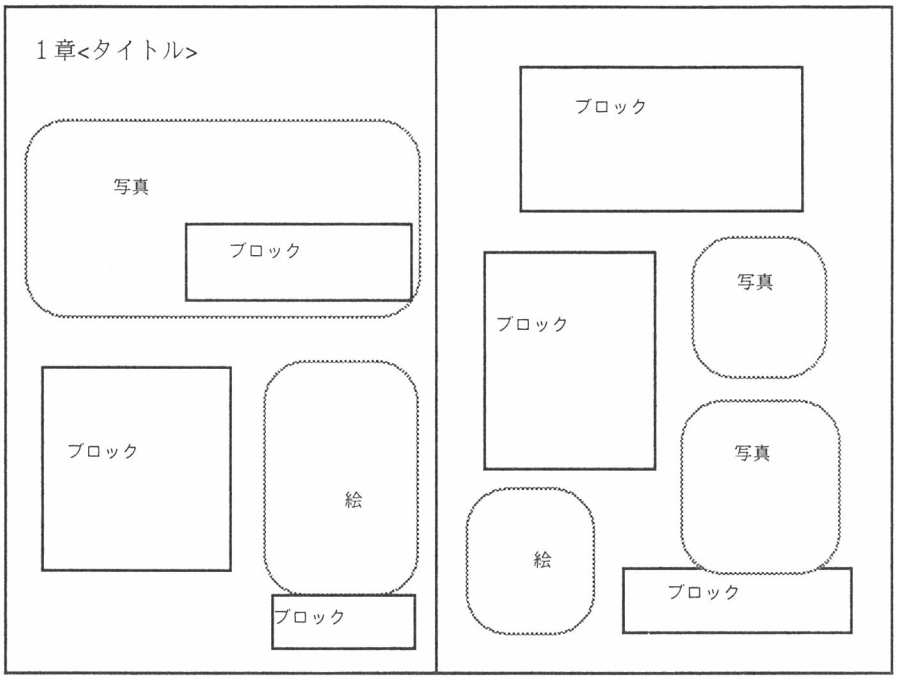

図1 教科書におけるのブロック単位の文章のまとまりモデル

最後に，文末に意図を持つ文があれば，ブロック内のストリームを１つにまとめた形で，ブロック全体がなんらかの意図をもっているように見せる機能があると考えられる．

以上のことから，ここでは，ブロックの最終文で他のブロックに対する意図の有無が判断できるとする．

なお，末尾の表現により，ブロックが学習活動の流れにおける意図を示す表す機能を持つと考えられることについては別の機会に述べる．

2.4 ブロックの属性

なんらかの意図を表しているブロックを有標とすれば，意図のない無標ブロックと区別できる．この区別を利用することで，ブロック単位の構成パターンを明らかにできる．なんらかの意図を表しているブロックの最後の文の表現形態は，決まった文末表現を持つ同じ形式の文が用いられていることが多い．特に，学習指示は教科それぞれ決まった文末形態を持っており，リスト化できる．このブロックの中の最後の文の指示表現を参照することでブロックに属性を付加する．

したがって，学習指示の文がブロックの最後にある場合，そのブロックは有標となり，その最後の文が持つ属性が，ブロック属性として付加される．無標の場合は最終文で指示文以外のものだということで判断できるが，学習指示以外の文についても「説明」文と「その他」という属性を与え，意図を含む表現のチェックを行い，遺漏がないようする．８種類の属性を図2に定義する．

A :指示・呼び掛け	・・・ましょう．てみよう．よう．
B :命令	・・・なさい．
C :指示・疑問	・・・だろう．だろうか．
D :指示・注意	・・・てはいけない．なければならない．
E :説明	・・・普通体
F :解説	・・・丁寧体
G :誘導	・・・終助詞
H :アドバイス	・・・～とよい．
I :その他	

図2 属性の定義

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<textbook>
  <section><title>物の燃え方と空気</title>
    <part kind="explain_body">
      <sentence skind="explain_body">は　‥　けが出ることがある．</sentence>
      <sentence skind="explain_body">し　‥　く燃えるようになる．</sentence>
    </part>
                          …
    <subsection><title>火と空気</title>
      <part kind="explain_body">
        <sentence skind="explain_body">はんごうすい　‥　ている．</sentence>
        <word>「隙間を作るとよく燃えるのは，どうしてなのだろう．」</word>
      </part>
      <part kind="inst_question">
        <sentence skind="explain_body">ロウソク　‥　付被せる．</sentence>
        <sentence skind="explain_body">一方の瓶には，蓋をする．</sentence>
        <sentence skind="inst_question">ロウソクの　‥　うなるか．</sentence>
      </part>
                          …
      <subsubsection><title>物をよく燃やす工夫</title>
        <part kind="explain_comment">
          <sentence skind="explain_comment">ゴミ　‥　されます．</sentence>
          <sentence skind="explain_comment">焼却　‥　ています．</sentence>
        </part>
                          …
        <part kind="explain_comment">
          <sentence skind="explain_comment">また　‥　一つです．</sentence>
        </part>
      </subsubsection>
    </subsection>
```

図3 XML化教科書

```
<xsl:template match="/">
  <xsl:for-each select="//part">
    <xsl:number level="multiple"
          count="section|subsection|subsubsection"/>
    <xsl:choose>
      <xsl:when test="@kind='inst_call'">A</xsl:when>
      <xsl:when test="@kind='inst_order'">B</xsl:when>
      ‥
      <xsl:when test="@kind='inst_advice'">H</xsl:when>
      <xsl:when test="@kind='els'">I</xsl:when>
    </xsl:choose>
  </xsl:for-each>
</xsl:template>
```

図4 パートの出現パターンを抽出するXSLTのひな形

```
1.0.0:A:植え，育てていこう．
1.0.0:F:種芋を植えて育てる．
1.0.0:F:切って植えてもよい．
1.0.0:F:の図のように植える．
          :
2.2.0:F:そのままにしておく．
2.2.0:C:灰水は，どうなるか．
2.2.0:F:調べることもできる．
2.3.0:F:ができたからである．
2.3.0:F:ていることがわかる．
```

図7 意図パターンと文末表現の確認

```
1.0.0:A          1.0.0:AFFFFFF
1.0.0:F          2.0.0:FA
1.0.0:F          2.1.0:FCCF
1.0.0:F          2.1.1:EEE
1.0.0:F          2.2.0:FFCF
1.0.0:F          2.3.0:FF
1.0.0:F          2.3.1:CFFFF
2.0.0:F          2.3.2:FFFFF
2.0.0:A          2.4.0:FCFF
2.1.0:F          2.4.1:FF
2.1.0:C          2.5.0:FAFFFF
2.1.0:C          2.5.1:FFFFDF
2.1.0:F          2.5.2:FFFFF
2.1.1:E          2.5.3:F
2.1.1:E          3.0.0:FA
2.1.1:E          3.1.0:FFFCF
2.2.0:F          3.2.0:CCFF
2.2.0:F          3.2.1:EEEEF
2.2.0:C          3.3.0:FCFFF
2.2.0:F          3.3.1:FFFFF
2.3.0:F          3.4.0:FFIFFFA
```

図5 章節部毎のパート出現パターン

```
<xsl:variable name="bun">
  <xsl:value-of select="sentence[last()]"/>
</xsl:variable>
<xsl:value-of select=
 "substring($bun,string-length($bun)-9,10)"/>
```

図6 文末表現の表示

3. XMLによる文書データの解析

前節で述べた教科書の構造をXMLで表現し，文書構造の中に埋め込まれたブロックを探索して意図の順序情報を抽出する．XMLの構造検索はXSLT(eXtensible Stylesheet Language Transformations)による[7]．また，文書データのXSLTに基づく構造検索とデータベース検索の対比を行う．

3.1 教科書のXML化

タグ付けの手順は以下の通りである．

（1）章(section)や節(subsection)などの，文書の構成単位である内容の区別を目的としたXMLタグをテキストに付加する．

（2）文(sentence)にXMLタグを付け，文の意図を示す属性を付与する．

（3）ブロックとしてパート(part)タグを付ける．

（4）ブロック内の最終文を参照してブロックに属性を付加する．

以上の手順を経て，教科書がXML形式のデータとなる．例を図3に示す．紙面の都合上，構造を部分的に示した．

3.2 XML構造検索

章節内でのブロックの出現位置を調べる目的であるから，ブロックがどこにあるかを示すため，文書内の位置情報を得る必要がある．

上記3.1の手順で行ったXML化からは，有標ブロックの出現に着目してその出現のパターンを見いだすことが目的であるため，必要となる検索対象は異なった出現パターンが現れる箇所である．したがって，検索結果をユニークに提示し，教科単元に相当する章節内のどの部分に有標属性を持つブロックの出現をみることが主な目的である．もちろん，出現総数等の補充資料も作成する．

XSLTはXML形式のデータを別のXML形式のデータへと変換する記述するXMLのアプリケーション規格である．XMLの要素(ノード)に対する変換規則を同じくXML形式で記述するXMLのアプリケーションの一つとして規定されたもので，この変換規則によって，階層構造を持つXMLデータの構造に基づいた探索が行える．

基本的には，個々のノードに基づく変換規則のひな形(template)を定義して，変換するが，その階層構造のノードを指定するのにパターンで記述することができる．また，特定のノードが持っている値を引き出すにも，このパターンが用いられる．例えば，「1章2節の最初の文」といえば，

```
/textbook/section[position()=1]/
subsection[position()=2]/sentence[first()]
```

といったパターン記述が可能である．

一つのノードのひな形はその部分木を表している．そのため，そのひな形の変換規則として，相対的なパターンを記述することで，部分木の中の複雑な操作を行なうことも可能である．さらにノードに対してその要素の名前に基づき，階層構造の深さに依らない指定を行なうことができるので，構造を横断する探索が容易に指定できる．

実際のXML構造検索のためのXSLTによる変換は，IBM社のlotusXSLを用いた[8]．

3.3 パートの出現パターンの抽出

この変換規則を用いて，パートの種別属性(kind)を抽出する．そのためには任意の部分木にある全てのパートノードを選び，そのkind属性を取り出す．その際，位置情報として章節部のそれぞれの番号を付与する．実際の種別としては一目で意味が分かるようにキーワードが付けられているので種別を後の処理のために，図2で記述した記号への変換を施す．これを行なうものが図4である．細かな書式と途中は割愛した．

実行した結果(抜粋)を図5の左に掲載する．パート毎に，各章節部と種別記号が並んでいる．番号0は節部の前に置かれたパートを表している．この結果を章節部毎に，集約したもの(抜粋)が図5の右にあたる．

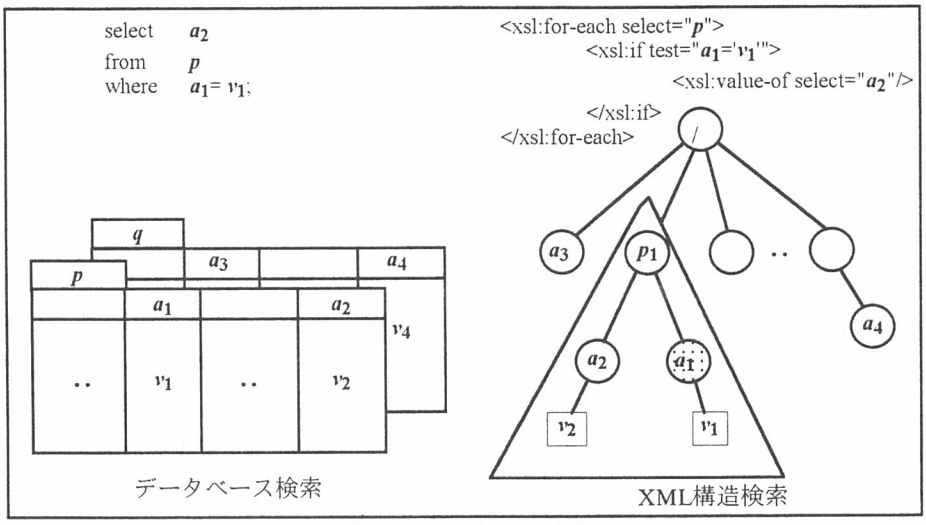

図8 データベース検索との対比

3.4 パートの出現パターンと文末表現の比較

XSLTもデータベースと同じように集約関数やカウンタなど，結果を加工する機能が備わっている．その機能を用いて，パートの出現パターンと文末表現を比較する．これはパターンがパートの最後の文の文末表現に拠っていることの確認を行なうためのものである．そのためには，パートノードに含まれる最後の文を取り出して，末尾だけを出力する部分を前出の図4に付け加える．この作業を行なうXSLTの部分を図6に掲げる．結果(抜粋)は図7のように出力される．

3.5 データベース検索とXML構造検索との比較

データベースにおける検索をXML構造検索と対比したものが，図8である．データベースにおいては，まず，参照すべき関係(p)を決定し，検索条件(a_1=v_1)に合致するレコードを求め，そのレコードの求める結果(a_2)の値を返す．これに対して，XML構造検索では，木構造の対応するノード(p)で指定される部分木の集合に対して，子ノード(a_1)が値，v_1を持つものを探し出して，その部分木に含まれる子ノード(a_2)の値を返すことになる．

しかしながら，木構造の探索は対象となるノードへの経路(a_1までの経路)と結果となるノードへの経路(a_2までの経路)で決まり，ここで示したような部分木に限らないので，このようなデータベース検索のアナロジーを越えて探索することが可能である．経路のパターンが記述できるならば，条件として指定されたノード(a_1)から親ノード(p)を遡って，目的とするノード(図のa_3やa_4)を返すことも可能である．例えば，教科書の例でいうと「星の観察」を含んだ節を探索し，それを含む章のタイトルを返す，「澱粉の抽出実験」の節の後に続く「結果の考察」の節を取り出す，等，特定のノードから遡ってノードを探す例は容易に見つかる．

これをデータベース検索において実現するならば，必要な情報を持つ別の関係(q)を検索に取り込み，追加された関係(q)と元の関係(p)の直積集合に対して，検索を施すことで，広がった範囲の属性を取り出すことになる．完全な記述をするためには，より複雑な関係記述を行なって，データを再構成する必要がある．しかしながら，これはデータベースのインピーダンス・ミスマッチにつながる．

このナビゲーションの有用性とインピーダン

ス・ミスマッチについては，既に，オブジェクト指向データベースに対する関係データベースの問題点として指摘されたものであるが，この文書データにおけるXML構造検索にはまさにこれが当てはまる．

4 考察

文書データをXML化して扱うことで今回の研究は行われた．多くのデータを処理するため，文書データをRDBMSに格納し，活用することや今回のXMLに基づく処理などいくつかの方法を試みている．

ここでは，文書データをRDBMSで扱う場合の問題点とXMLで扱う利点を述べる．

4.1 文書データとRDBMS

RDBMSでは多様性を持つデータを，行き届いた管理の下で，集合として保管する．検索は問い合わせ言語を利用し．検索結果は，「全体」集合の中から「検索条件に合致するもの」の集合が「関連データ」集合として行単位で同列に扱われて返される．

しかし，文書の内容を解析する場合，RDBMSを利用すると，そのメリットである保全性の高さや集合単位の処理であることにより，解析以前の処理にコストがかかる．

なぜなら，文書データは，正規化規則に従い，文を１単位で扱うか文の構成単位である形態素レベルで扱うかにより，別テーブルに格納仕分けることになる．テキストの構成成分に付随するバックグラウンド情報は，扱う単位によりそれぞれのテーブルに一緒に格納していてもよいが，分析結果を各テーブルに付随するデータとして扱う場合は，さらに別のテーブルに格納する必要がある．データを扱う切り口を変える場合は使用目的に応じて格納形式を変える必要があるためだ．

したがって，複数世代に渡るような親子関係やデータ全体における位置情報を談話機能情報と関連づけて格納するには，階層構造と各データの関係付けを細かく定義していかなければいけないことから，スキーマデザインが複雑になり，格納までの処理が容易ではないのである．

また，格納しても，構造についての情報は再加工処理を行う必要がある．ということは，データは，切り取ったり，複製したりと，ばらして扱うということになる．

しかも，問い合わせ言語を利用するため，スキーマが複雑になれば，問い合わせ言語による検索対象を指定する文が非常に複雑になる可能性がある．多くのテーブルにまたがる検索を行う際に，必要な情報を取り出すテーブルを見失わないようにと気を遣う．

以上の問題から，文書データをデータベースに保管してその内部の構成を深く解析していくには，あまり利便ではないのではないだろうか．

一方，XMLはデータを階層化して扱う．また，検索はXSLTで行うが，タグによりまとめられたデータの下位概念等の階層性を考慮して順に取り出すことや順序を再構成して表示することができる．DTDやXSLTを変更することで検索結果の返し方を容易に変更できるためである．

ただし，１つのXMLデータから部分木の集合が得られるとされているが，その処理方法や，検索で取り出したデータは部分木の集合のままであるため，検索結果の扱い方などは使用者がデータの性質によく配慮してデザインする必要がある．

また，以上に述べてきたことは，XMLの扱い方が容易であるということであり，RDBMSが持っているデータを安全に保存する，インデックスによる高速な検索ができる点に関してはXMLだけでは何の答えも与えられない．

4.2 文書データをXML形式で扱う利点

XMLで文書データを表現することで，従来個

別に設計せねばならなかった階層構造を汎用のデータ表現として記述できるようになった．汎用化されたことにともない，XSLTのような精度の高い変換のためのアプリケーションも登場している，それらによってもたらされている利点としては，

(1) 構造を保ったまま，データを格納できる．
(2) 文書データの特質がそのまま保たれているので，順序関係や上下関係を自在に探索できる．
(3) アノテーションフリーなデータとして扱うことができる．

等が挙げられる．ここで，(3)は説明を要する．XMLは元来，文書の内容を損なうことなく，マーク付けを行なうものであり，それが構造記述につながっている．しかし，内容を損なわないものであるなら，研究者などが行なう活動に伴う付加データも，元々のデータにマーク付けの形で埋め込んでも構わないことになる．

データとはそれ自身，貴いもので，決してあだや疎かに加筆等してはならないものであるという考え方を，この形式で払拭することが可能である．適当なルールを定めることで，データの利用者はアノテーションとしてデータ自身に加筆することができる．そのような加筆は，XSLT等を利用することで，取り除いたり，クローズアップすることが可能である．

5. まとめ

本論文では，教科書に書かれた文章を意図別パート毎に取り出すことから，その出現パターンを調べた．

その際，教科書中の階層構造に埋め込まれた文書パターンの出現順序を求めるため，構造を記述できるXMLで木構造を表現し，文書データに当てはめた．

小学校教科書の構成を踏まえ，(1)通常の文章構成である章，節や部，段落を利用して，文章に階層構造を持たせること．(2)視覚的単位であるブロックを定義し，通常の階層構造の下位に組み込むという２点に留意した．

下位に段落以下の階層を含む部分木であるブロックを探し，ブロックの中の最終文により属性として付加されたブロックの性質を検索結果として位置情報とともに出した．

表現意図の出現パターン調査において，教科書の構造，教科書テキストの階層化の記述方法，XMLの利用方法，系列検索について述べた．

そして，最後に，今回利用した方法で文書を扱うことについて考察を加えた．

今回の調査では，パターンの出力において，テキスト構造の中のブロックを全て同列に扱った．しかし，ブロックは全て同じレベルではなく，主流となる話の流れと副次的な話の流れがある．意図が表されていない無標のブロックをさらに分類するには，より具体的で忠実な教科書の定義が必要であり，そのためのXMLのデザインを考慮しなければならない．これらは今後の課題である．

参考文献

[1] 外国人子女の日本語指導に関する調査研究協力者会議, “外国人子女の日本語指導に関する調査研究＜最終報告書＞,” 東京外国語大学, Jun. 1998.

[2] 白鳥智美, “児童生徒に対する日本語教育のための語彙調査- 社会科教科書の語彙-,” 日本語教育学会平成１２年度春期大会予稿集, Mar. 2000.

[3] 工藤真由美編, “算数・数学教科書の日本語の考察- 日本語教育の観点から-,” 横浜国立大学教育学部, 1997.

[4] 中尾桂子・本村康哲, “日本語教育のための文データベースの構築,” 情処研報, 1998.

[5] http://www.w3c.org/XML/

[6] 益岡隆志, “モダリティーの文法,” くろしお出版, May. 1991.

[7] http://www.w3.org/TR/xslt.html

[8] http://www.alphaworks.ibm.com/tech/LotusXSL/