社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

数学文書点訳における文書構造処理インターフェイス

金堀　利洋†　　仲　　正幸†　　鈴木　昌和††

† 筑波技術大学 障害者高等教育研究支援センター　〒 305–8521 茨城県つくば市 4–12–7
†† 九州大学 数理学研究院　〒 812–8581 福岡県福岡市東区箱崎 6–10–1
E-mail: †{kanahori,naka}@k.tsukuba-tech.ac.jp, ††suzuki@math.kyushu-u.ac.jp

あらまし　現在，点字を墨字として編集できるエディタが利用されているが，数式においては数学記号の点字を墨字で表示する機能はなく，点字のまま編集しなければならない．そこで我々は数式文書の点訳システムを開発し，その報告を [5] にて行った．このシステムは数式も認識・編集可能な OCR システムと，新たに開発した数式点訳エンジン，そして数式の墨字表示も可能な点訳エディタから成っていた．今回，文章の点訳だけでなく，文書構造の点訳も行うためのインターフェイスを開発し，組み込みを行った．このインターフェイスを用いて文書構造を指定し，点訳エンジンはそれに合わせてマス空け，点訳記号の挿入などを行う．このインターフェイスと点訳エンジンの概要とその点訳手法を紹介する．

キーワード　数学文書，文書構造，自動点訳

Interface for Braille Translation of Mathematical Document Structure

Toshihiro KANAHORI†, Masayuki NAKA†, and Masakazu SUZUKI††

† Research and Support Center on Higher Education for the Hearing and Visually Impaired,
Tsukuba University of Technology　Kasuga 4–12–7, Tsukuba-shi, Ibaraki, 305–8521 Japan
†† Faculty of Mathematics, Kyushu University　Hakozaki 6–10–1, Higashi-ku, Fukuoka, 812–8581 Japan
E-mail: †{kanahori,naka}@k.tsukuba-tech.ac.jp, ††suzuki@math.kyushu-u.ac.jp

Abstract　Current Braille editor displays printed characters corresponding to Braille characters. However, for mathematical formulae, there is no practical system which displays mathematical symbols corresponding to Braille characters, so users must translate mathematical formulae viewing Braille characters. We are developing Braille translate system for mathematical documents and have reported at [5]. This system consists of mathematical document reader, mathematical Braille translator and mathematical Braille editor. The Braille editor can display mathematical symbols corresponding to Braille characters. In this report, we present new interface and new functions of our Braille translator to translate document structure. Using the interface, users mark up document structure in a document. After that, our Braille translator generates Braille document inserting spaces and Braille symbols corresponding to the document structure.

Key words　mathematical document, document structure, automatically Braille translating

1. はじめに

近年，文書の点訳において，コンピュータは大きな役割を果たしている．光学文字認識 (OCR) システムによる文書の電子化，自動点訳システムの導入により，それまで専門的な知識，技術が必要であった作業がある程度自動化された．また，文字認識や点訳の結果を，墨字の状態との対応を見ながら修正やレイアウトの調整などを容易に行える点字エディタが開発され，その結果，点訳作業のコストの軽減，点訳従事者の増加がもたらされた．しかし，数式，化学式，図表などを含んだ文書の点訳においては，上記の限りではない．まず，一般的に用いられている OCR システムは，これら数式，化学式などをほとんど認識することはできないだけでなく，その周辺の通常の文書の部分にも認識エラーを起こしてしまい，その修正に更なる手間がかかってしまう [1]．自動点訳においても，数式では，その点字による表記 (数式の構造表記，数学記号の点字表記) を行う場合，LaTeX などの数式構造を表記する書式を介して行うシステム (ex. [2]) などが用いられているが，やはり，その書式に精通している必要がある．また，通常の文書の点字エディタのように，数式の点訳結果と墨字の状態を対応させながら編集・修

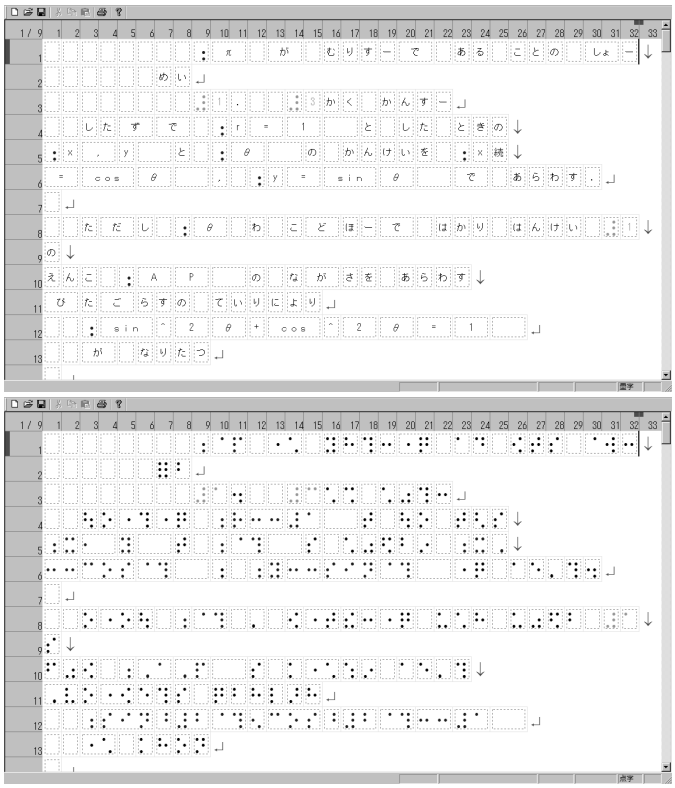

図 1　開発した数式点字エディタの画面 (InftyBrailleEditor)
実際の画面では，テキスト部は黒色，数式は青色で描画される
(上図:墨字表示 下図:点字表示)

正できるような実用的なシステムは存在しない．以上のような理由もあり，専門的な文書の点訳作業には手作業の部分が増え，大きなコストがかかってしまっている．

現在，我々は数式を含んだ科学技術文書認識システム *Infty* を開発しており，これは通常のテキスト部分の認識はもちろん，数式に関してはその構造解析，数学記号認識が可能である [3], [4]．更に，認識結果を編集・修正するエディタも備えている (図 3)．この Infty を用いて数式を含んだ文書の点訳システムのプロトタイプを開発し，報告を行った [5]．このシステムでは，通常の文書部分 (以下，テキスト部) はもちろん，数式部の点訳も行い，その結果も対応する数学記号や数式構造を墨字で表示・編集することが可能である (図 1)．

しかしながら，これらの機能でだけではまだ実用的な点訳システムとして不十分であり，特に点訳時に必要な，見出しや，箇条書きといった文書構造の情報を入力することが出来なかった．そこで今回，Infty の機能を拡張し，これら文書構造の入力を可能とし，更にそれら文書構造の自動点訳を行う機能を追加し，実用的な点訳システムの開発を行った．

2.　点訳の流れ

点訳作業は以下の複数の段階を経て行われる (図 2):

1) 認識・修正作業

点訳対象となる文書・本をスキャンし白黒の解像度 600dpi の画像にし，Infty の認識システム (InftyReader) を用いて認識を行う．認識した結果を，InftyEditor のインターフェイスを

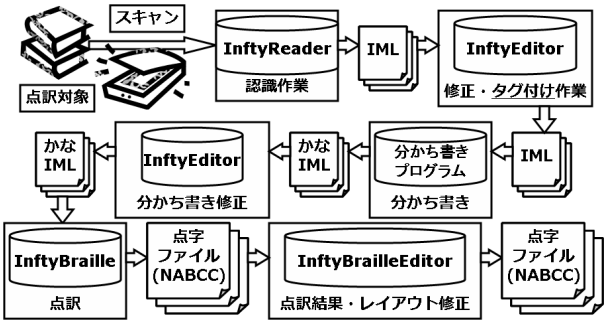

図 2　点訳作業のワークフロー

用いて修正を行っていく．また，InftyReader は PDF の認識も可能であるため，PDF 文書の点訳を行うこともできる．この時，次の “タグ付け” と呼んでいる文書構造の指定も行う．

2) タグ付け作業

認識結果を修正した文書に対して，行単位や文字単位で文書構造の指定を行っていく作業であり，今回新しく追加した機能である．例えば，行単位の文書構造とは，タイトルや，見出し，脚注，更には著者名やその所属といった情報を，そのブロックにタグを付けることによって指定するようにしている．また，文字単位では，箇条書きの行頭の印や，‘問 1’ といった行頭の見出し，また目次の見出しや目次のページ番号などの行中の一部にあたるものに対してタグを付けて指定する．この作業に関しては，節 3. において詳しく説明する．この作業によって指定された構造を元に，点訳時に文書構造の情報も点字に反映させる．

3) 分かち書き処理と修正作業

タグ付けされた漢字かな･数式交じりの文書を一旦，IML(Infty Markup Language) という XML としてファイルに保存し．そのファイルに対して分かち書き処理を行う．分かち書きした結果は漢字の部分はかなに分かち書きされているので，かな・数式交じりの IML ファイル (かな IML ファイル) となる．前回の報告 [5] においては，点訳作業において広く用いられている EXTRA [7] を使用していたが，今回は我々が独自に開発している分かち書きエンジンも使用できるようになっている．この分かち書きエンジンは，数学文書に特化して開発されている．更に，分かち書き結果に誤りのある可能性のある箇所をマークアップする機能を持っており，また認識したページ画像上の座標情報も持たせたままにすることが出来る [6]．この機能を利用して，我々の分かち書きエンジンを使用した場合，InftyEditor 上で分かち書きの修正を行う際に，誤りがある可能性のある箇所を赤く表示し，ページ画像上で分かち書きする前の漢字と分かち書きした結果を対応付けて表示するようにした．これによって修正作業時には，元のページ画像上の漢字を見ながら，赤く表示された箇所の分かち書き結果だけを確認し修正することで，分かち書き作業の効率を上げることが可能となった．

4) 点訳処理

上記の分かち書き作業によって作られた，タグ付きのかな IML ファイルを，我々が開発した点訳エンジン (InftyBraille) を用いて点訳する．点訳処理は，文字の点訳と文書構造 (タグ)

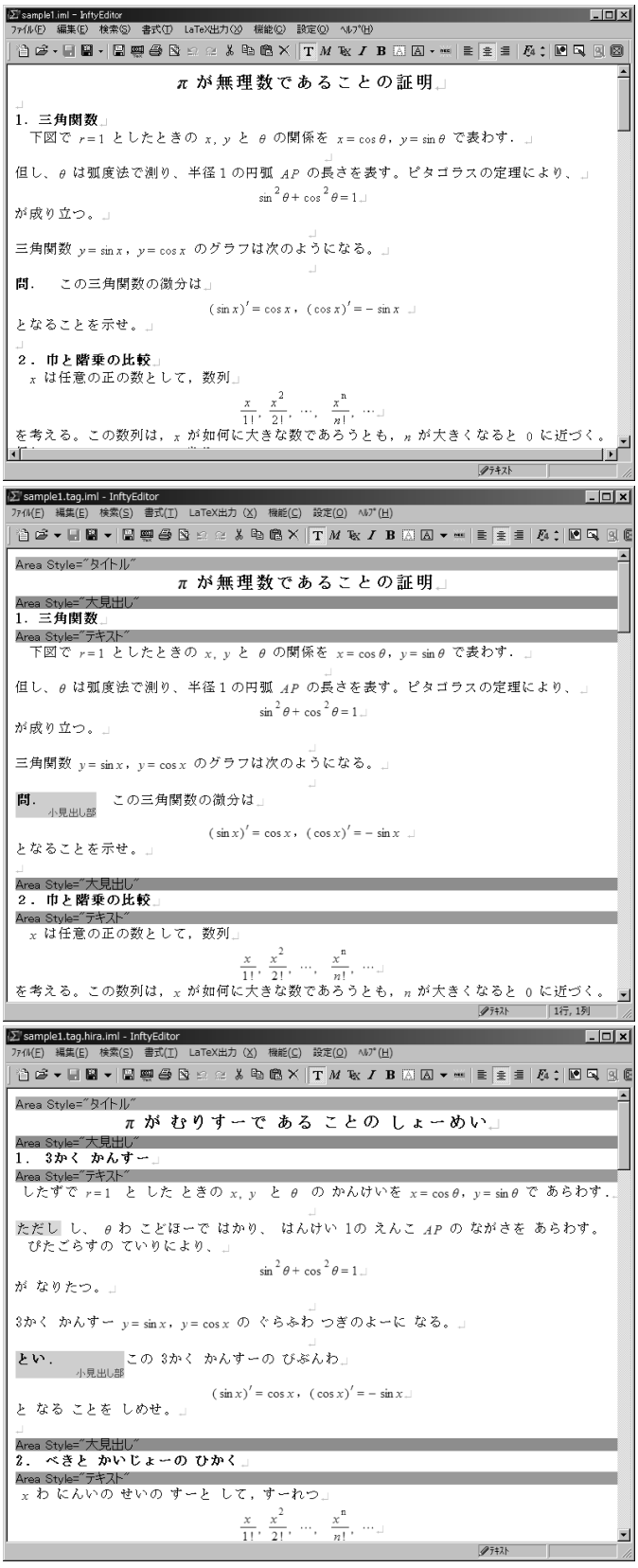

図 3　上段:Infty の編集画面 (InftyEditor),
中段：タグ付けされた文書画面,
下段：分かち書きした結果.
実際の画面では，テキスト部は黒色，数式は青色で描画される．また，分かち書きが誤っている可能性のある部分は赤く表示される (下段：「ただし し」の部分)

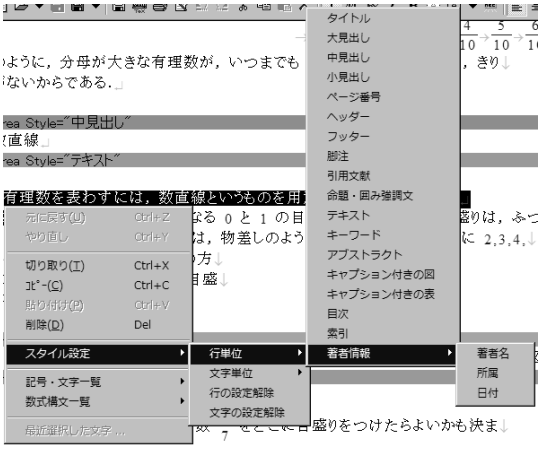

図 4　タグ付け時のコンテキストメニュー

の点訳の 2 つに分けられる．文字の点訳は，テキスト部と数式部毎に開発した点訳エンジンを用いて行い結果を統合する形を採っている [5]．この処理に加えて，今回新たに文書構造に合わせてスペースや点訳記号の挿入を行い，指定された点字の 1 行あたりのマス数に合わせて適切な位置で改行を行う．この新たな処理に関して，節 4. において詳しく説明する．

5) 点訳結果確認

点訳の結果を，我々が開発している数学文書点訳エディタ (InftyBrailleEditor:図 1) を用いて確認を行う．点訳に誤りがあった場合，特殊なレイアウトや点訳を除いて，このエディタで修正は行わず，点訳エンジンの修正・改良によって正しい点訳が行われるようにしていく．これによって，より精度の高い点訳エンジンを開発し，最終的には，タグ付けと分かち書きが正しく行われれば正しい点訳結果が得られるようになる事を目的としている．即ち，点字や点訳規則など一切知らなくても点訳作業が行える環境を作ることを目指している．

3.　文書構造の指定–タグ付け–

“タグ付け” とは，点訳に必要な文書構造を入力する作業であり，これを元に点訳エンジンは，マス空け，位置揃えを行う．タグ付けには，行単位と文字単位の 2 つがあり，それぞれ指定可能なタグを表 1 と表 2 に示した．実際に点訳する際には，用意しているタグほど細かくする必要はないが，文書の書誌情報を得るために用意しているタグもあるためにこのように細かく指定するようにしている．

タグ付け作業は点訳用の機能を実装した InftyEditor を用いて行われる．指定方法は非常に簡単で;

（1） タグ付けする範囲を指定,

（2） 右クリックもしくはアプリケーションキーなどでコンテキストメニューを出す (図 4),

（3） メニュー項目 [スタイル設定] のサブメニューとして，[行単位] もしくは [文字単位] を選ぶ.

（4） [行単位] もしくは [文字単位] のサブメニューの中から，タグを選ぶ.

タグをつけると行単位の場合は，タグ付けした範囲の先頭にそのタグ名の入った行が挿入される．文単位の場合はタグ付けさ

れた部分の背景色が変わり，タグ名が表示される (図 3 中段). 索引の部分に関しては通常，“索引見出し” と “索引ページ番号” が大量にあるので 1 つ 1 つ手動で指定していくのは大変手間がかかるため，索引全体を “索引” タグで指定することで，“索引見出し” と “索引ページ番号” を自動的に抽出しタグ付けする機能を用意している．ここで注意するのは，現在のシステムでは “目次ページ番号” や “索引ページ番号” といったページ番号は，点訳対象の文書に記載されているページ番号であるので，点訳後の文書中のページ番号ではないという点である．

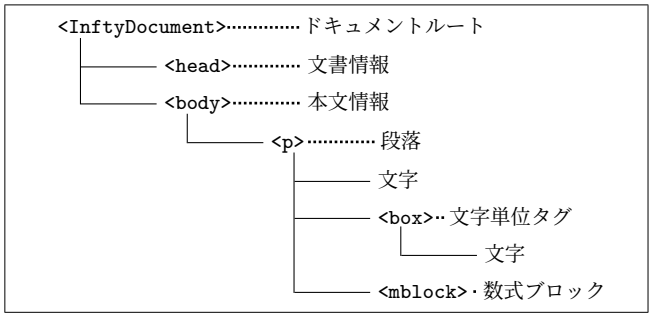

図 5　<box>は段落<p>中にあり，<box>はその中に文字列を持つ．また，IML 内での数式は<mblock>として切り分けられている

4. 文書構造の点訳

今回，文書構造タグを Infty のファイル形式 IML に導入し，かつ，既存の IML ファイルとの互換性を維持するために，元々文字を四角で囲むための<box>の style アトリビュートに文字単位タグ名を指定できるように拡張した．IML 中の<box>の位置を図 5 に示した．ここで，行単位タグは<p>の style アトリビュートにそのタグ名が保持される．例えば，

```
    π が無理数であることの証明
```

の 1 行が “タイトル” であると指定されていれば，

```
<p style="タイトル">
   <mblock><munit>pi</munit></mblock>が無理数であることの証明
</p>
```

と表される．また，複数行に渡ってタグ付けされている場合，例えば，

```
    y = f(x) が，x = p で，極大値，または極小値をとるならば

  f'(p) = 0

である．
```

全体が “命題・囲み強調文” であれば，

```
<p style="begin. 命題・囲み強調文">
   <mblock>
      <munit>y</munit><munit>=</munit>...
   </mblock>
が，
   <mblock>
      <munit>x</munit><munit>=</munit>...
   </mblock>
で，極大値，または極小値をとるならば
</p>
<p align="center">
   <mblock>...</mblock>
</p>
<p style="end. 命題・囲み強調文">
   である．
</p>
```

というように，タグ付けされた最初の行の<p>の style アトリビュートに “begin. タグ名”，最後の行の style に “end. タグ名” と入り，その間の行の style は指定されない．上の例での 2 つ目の<p>の align アトリビュートは，中央寄せで表示，という意味であり，タグとは関係ない．

以上のような形でタグ情報を保持した IML ファイルを，1 行 (<p>) ごとに以下の流れで点訳していく．ただし，行タグ “ページ番号” に関しては別処理を行う:

（1） テキスト部と数式部，そして<box>に分けて，それぞれ点訳エンジンにかける．<box>内の点訳は，<p>の点訳と同様に行う．テキスト部と数式部の点訳に関しては, [5] と同様の処理であるので，ここでは省略する．

（2） テキスト部と数式部，<box>の中の点訳結果を 1 つの行にまとめる．数式部の前には数符，後ろにはスペースを挿入する．<box>の点訳結果をまとめる際には，表 2 の点訳規則にしたがって，マス空け，点訳記号の挿入を行う．

（3） 点訳している行のタグと，表 1 の点訳規則に従って，行頭のマス空けなどを行う．この時，既に行頭にスペースが入っている場合は，それらは削除しておく．

（4） 点訳結果をまとめた 1 行のマス数 (点字数) が，指定された 1 行あたりのマス数 N より大きければ，N に収めるために行替えを行う．まず，N マス目から行頭に向かって以下の条件を満たす行替え可能位置かどうか調べていき，可能であれば行替えを行う．その条件とは;

- 数式中であれば，2 項演算子・関係演算子の前,
- テキスト部であれば以下の場所以外;

（a） 句読点・感嘆符・疑問符・中点の前,
（b） 括弧類・カギ類の閉じの前,
（c） 括弧類・カギ類の開きの後ろ,
（d） 数符・外字符の後ろ,
としている．

行頭までにこれらの条件を満たす位置が無い場合，数符・外字符の後ろ以外の場所であればその位置で行替えするようにしている．次に行替えした行に対して，まず，行タグの条件によってマス空けを行う．ただし，行替えした場合はマス空けの

数を 2 マス減らす．そしてまた，1 行のマス数が N マスより大きければ，N マス目から行頭に向かって行替え可能位置を探し，行換えを行っていく．これを 1 行のマス数が N マスより小さくなるまで繰り返していく．

ページ番号に関しては，行タグが “ページ番号” であった場合;

（1） 行中のページ番号を抽出する．現在，ページ番号として抽出するのは，1) 番号のみ，2)p-“番号” の 2 パターンとしている．これ以外の場合は，そのまま点訳して出力する．

（2） 番号が抽出できたら，表 1 の点訳規則に従って，ページ番号を出力し，次の行<p>の点訳に移る．

点訳時に使用する表 1 と表 2 にある点訳規則は外部ファイル (リソース) として与えられており，点訳文書を使用するユーザー合わせて適宜変更することが可能である．

以上の様にして，各行<p>に対して点訳を行い，各行の点訳結果を順に NABCC(北米点字コード) で点字ファイルを出力する．

5. 今後の課題

今回，我々が開発している数学文書点訳システムにおいて，文書構造を指定するインターフェイスと，指定された文書構造に合わせて点訳・レイアウト調整を行う点訳エンジンの手法の紹介を行った．この点訳システムではまず，科学技術文書認識システムを用いて数式を含んだ文書の認識を行う．その認識結果の修正，文書構造の指定，分かち書きの修正を我々が開発している数式エディタを用いて行う．またこの分かち書きの際には，広く用いられている EXTRA [7] だけでなく，我々が開発している数学文書用の分かち書きエンジンを使用することが可能である．このエンジンを用いた場合，分かち書き結果を，認識時の画像中の分かち書き前の漢字と対応させたり，分かち書きを誤った可能性のある箇所を赤く表示させ，点訳従事者の注意を喚起させることができる．点訳時には，指定された文書構造に合わせて，スペース，位置合わせ，点訳記号の挿入を行う．この点訳規則はリソースで与えられており，点訳文書のユーザーに合わせて変更することが可能である．数式の点訳結果も墨字の数学記号を表示し，数式の構造もわかり易い記号で表示することで，点訳結果の確認やそのレイアウト補正を可能としている．

我々の最終的な目的は，点訳対象文書の認識結果の修正，文書構造の指定，分かち書きの修正が適切に行われていれば，点訳エンジンにかけるだけで，完璧な点訳文書が得られるシステムの構築である．これによって，全く点字を知らなくても点訳を行うことができ，点訳従事者の裾野を大幅に広げ，点訳にかかるコストを削減することが可能となる．しかしながら，実現にはまだいくつか課題が残っている．主なものとしては;

（1） 現在，目次・索引に現れるページ番号は元文書に記されているものそのものを用いている．点訳した際にはもちろん，ページが元文書とずれるために，点訳結果のページ番号との対応を解析して，点訳に反映させる必要がある．

（2） 現在，行列は 1 次元で表示している．行列に関しては読みやすい 2 次元での表記方法が望ましいが，行列の大きさ，要素の文字数などによって，改行する位置が複雑である．行列は分野によっては頻繁に現れる数式であるので，この問題に関しては早急に対応したい．

（3） 図表は現在，点訳の対象外にしているが，図に関しては，現在，数式を表現できる点図エディタを開発中である．また，表に関しては，行列ほど複雑ではないが，大きな表をいかにして読みやすく表記するか，といった問題があると思われる．表ももちろん頻繁に現れるので，行列と同じく早急に対応したい．

また，手入力を更に減らすために，ある程度文書の認識時に文書構造も取得する機能も現在，開発・実装中である．

点訳以外での文書構造入力インターフェイスの応用としては，LaTeX や HTML といった文書構造を持ったフォーマットへの出力が考えられる．Infty システムは既に LaTeX や HTML, MathML といったフォーマットへの出力をサポートしているが，文書構造をこれらフォーマットへの変換に用いることで，より質の高い出力結果を得られると期待できる．また，PDF の生成に用いることで，タグ付けされたアクセシブルな PDF 文書の生成にも応用できると思われる．

このように今回のシステムは点訳だけでなく，幅広く文書のアクセシビリティの向上に貢献が期待できるため，目標の実現を目指し，今後とも改良・開発を進めて行きたい．

文　献

[1] T. Kanahori and M. Suzuki, “Refinement of digitized documents through recognition of mathematical formulae”, the Proceedings of the 2nd International Workshop on Document Image Analysis for Libraries, pp.297–302.

[2] 楠 加奈子, 佐藤 浩史, 原 俊介, 大武 信之, “自動点訳システムのための LaTeX マクロ展開”, 電子情報通信学会技術研究報告, ET97-81, pp.1–8, 1997.

[3] M. Suzuki, T. Kanahori, N. Ohtake and K. Yamaguchi, “An Integrated OCR Software for mathematical Documents and Its Output with Accessibility”, *Computers Helping people with Special Needs*, 9th International Conference ICCHP2004, Lecture Notes in Computer Sciences 3119, Springer (2004) pp.648–655.

[4] Infty 公式ウェブサイト, “Infty Project”, `http://www.inftyproject.org/`.

[5] 金堀 利洋, 内田 智也, 高村 明良, 鈴木 昌和, “数式を含んだ文書の点訳システム”, 電子情報通信学会技術研究報告, WIT2006-5, pp.23–26, 2006.

[6] 橘美紗, 鈴木昌和, “数学文書点訳における日本語分かち書き仮名変換処理”, 電子情報通信学会技術研究報告, WIT2007-1, 2007, 掲載予定.

[7] (有) エクストラ, “高機能自動点訳ソフトウェア EXTRA”, `http://www.extra.co.jp/`.

タグ名	説明	点訳規則
タイトル	本，論文のタイトル	先頭 8 マス空け
大見出し	章，節など，その文書内での最大の見出し	先頭 8 マス空け
中見出し	大見出しの次に大きな見出し	先頭 6 マス空け
小見出し	中見出しより小さな見出し	先頭 4 マス空け
ページ番号	原文のページ番号	行頭から点字 {3,6} を続け，行末 6 マス目からページ番号を出力
ヘッダー	文書のヘッダー部	出力するかしないかを選択可．する場合はテキストと同様
フッター	文書のフッター部	出力するかしないかを選択可．する場合はテキストと同様
脚注	語句の補足説明	先頭に 1 行全て点字 {2,5} の行を挿入し，終わったら 1 行空ける
引用文献	引用された文献を 1 つずつ指定	テキストと同様
命題・囲み強調文	命題など文書中で強調されている段落	先頭に点字 {5,6}{2,3,5,6}，末尾に点字 {2,3,5,6}{2,3} を挿入
テキスト	タグ付けされない，文書の本文に当たる部分	行頭 2 マス空け
キーワード	論文などのキーワード部	テキストと同様
アブストラクト	論文などのアブストラクト部	テキストと同様
キャプション付きの図	図とキャプションをひとまとめにして指定	キャプションのみ，テキストと同様に表示
キャプション付きの表	表とキャプションをひとまとめにして指定	キャプションのみ，テキストと同様に表示
目次	目次全体を指定	実際は表 2 の目次見出し，目次ページ番号の規則に従う
索引	索引全体を指定	実際は表 2 の索引見出し，索引ページ番号の規則に従う
著者名	論文，書籍などの著者名を指定	テキストと同様
所属	著者の所属を指定	テキストと同様
日付	執筆，出版日などを指定	テキストと同様

表 1　行単位のタグの種類と点訳規則，点訳規則中のマス空けや点字などはリソースファイルによって指定されており，容易に変更可能

タグ名	説明	点訳規則
箇条書き 1	一番上のレベルの箇条書きのマークを指定.	先頭 2 マス空けマークは {3,5}{3,5}
箇条書き 2	2 番目のレベルの箇条書きのマークを指定	先頭 4 マス空けマークは {2,6}{2,6}
箇条書き 3	3 番目以降のレベルの箇条書きのマークを指定.	先頭 6 マス空けマークは {6}{2,6}
段落番号 1	(1), (2), …など見出しのついた箇条書きのラベルを指定.	先頭 2 マス空け
段落番号 2	2 番目以降の段落番号を指定.	先頭 4 マス空け
数式番号	数式番号を指定	先頭 2 マス空け，点字で “数式” と挿入し番号を表示
小見出し部	“問１．〜”，“例 〜” などの見出し部のうち，数学的な命題の見出しではないものを指定	見出し部の後ろに点字 {6}{3,6} を挿入し，更に後ろ 1 マス空ける
命題見出し部	“定理１．〜”，“証明 〜” などの行頭の 1 部が見出しとなるものを指定	見出し部の後ろに点字 {5}{2} を挿入し，更に後ろ 1 マス空ける
脚注マーク	† や ∗ といった脚注などと対応させる文字	指定した文字を点字 {5,6} と {2,3} で挟む
目次見出し 1〜3	目次の見出しを指定．ぶら下がりによってレベルを 3 つ用意	目次見出し 1 から順に先頭 2 マス，4 マス，6 マス空ける
目次ページ番号	目次見出しのページ番号を指定	目次見出しとの間を点字 {2} で埋め，行末から 6 マス目からページ番号を出力
索引見出し 1〜3	索引の見出しを指定．ぶら下がりによってレベルを 3 つ用意	目次見出し 1〜3 と同様
索引ページ番号	索引見出しのページ番号を指定	目次ページ番号と同様
e-mail	著者の e-mail アドレスなど	情報点字で出力 (開発中)
点訳者挿入	点訳者が特に説明を加える必要があると判断した場合に挿入	前後に点字 {2,3,5,6}{2,3,5,6} を挿入

表 2　文字単位のタグの種類と点訳規則，点訳規則中のマス空けや点字などはリソースファイルによって指定されており，容易に変更可能