

数式言語MathMLによる数学問題データベースの活用

メタデータ	言語: Japanese 出版者: 水産大学校 公開日: 2024-10-11 キーワード: Education; Data; Mathematics 作成者: 楢取, 和明, 青木, 邦匡 メールアドレス: 所属:
URL	https://fra.repo.nii.ac.jp/records/2011948

This work is licensed under a Creative Commons Attribution 4.0 International License.



数式記述言語MathMLによる数学問題データベースの活用

楫取和明*†, 青木邦匡*

Making use of a math problem database with the Mathematical Markup Language MathML

Kazuaki Kajitori*†, Kunimasa Aoki*

Abstract : We investigate benefits and catches of math problem databases with MathML functionalities based on our case studies in our math classes. We conducted online quizzes in the classes of Calculus in 2006-2008 using our system of math problem database (we call this QDB). The result was exactly what we expected, namely, we could develop the system in a short period of time and the students could get the evaluation of the quiz immediately after they finished and submitted the quiz. Also, we made a mid-term exam of Calculus and showed the answer on the web using our system. From these experiences, we found that QDB is ideal for some purposes, but also noticed that the environments for MathML are not sufficient yet, and that improvements of QDB can be done but only in a limited form. We think that we can make QDB more useful by adding functionalities even if they are for a rather narrow range of purposes and that the simplicity of the system should be maintained in order for the system to be user-maintainable.

ASFA keywords : Education, Data, Mathematics

はじめに

World Wide Web Consortium (W3C) のMathML 2仕様書¹⁾によれば, “MathMLはXML²⁾のアプリケーションで, 数式を記述し, その構造と内容を取り出すためのものである。MathMLの目標は, HTMLがテキストに対してするように, ウェブ上で数学を供給し, 受理し, 処理することである”。一般にMathMLのコードは機械で読み書きすることを想定したもので見た目には非常に冗長であるが, 一つ例を上げれば, $\sqrt{-1}$ を表すMathMLコードは,

```
<msqrt>
  <mrow>
    <mo>-</mo>
    <mn>1</mn>
  </mrow>
</msqrt>
```

である。<msqrt>はsquare root (平方根)を表すタグ,

<mrow>は横に並べるタグ, <mo>はoperator (演算子), <mn>はnumber (数)を表すタグである。

MathMLによって数式を表示できるブラウザ環境は現在でもまだ限定的ではあるが, MathMLによってウェブ上での数式使用の幅が広がりつつあることは事実である。実際にMathMLを活用しているウェブページはまだ少ないが, 金沢工業大のサイト³⁾では基礎的な内容の数学コンテンツをMathMLで表示している。また, 日本数学会の論文誌 Journal of the Mathematical Society of JapanはabstractをMathMLで提供している⁴⁾。

さて, 数学教育においては演習問題, 小テスト, 定期試験などのための問題作成が欠かせないが, 数学の問題をデータベース化して問題演習, 各種テストに利用できれば便利であろう。中学数学の問題データベースは商用では東京書籍の製品⁵⁾がある。これは5年契約の学校専売品であって, インターネット経由でデータベースから引き出した問題とその解答をPDFで出力できる。

2010年8月6日受付. Received August 6, 2010.

* 水産大学校水産流通経営学科 (Department of Fisheries Distribution and Management)

† 別刷り請求先 (corresponding author) : kajitori@sh-u.ac.jp

数学問題データベースのインターフェイスをウェブベースで作るとすれば自然とMathMLとのコラボレーションが考えられる。数学問題データベースにMathMLを応用した例は、数学教材製品のWebALT⁶⁾がある。Moodleベースのお試しサイトでは数学問題の表記は英語とフィンランド語である。データベースの管理に関しては不明である。

著者等は水産大学校において、4年前からMathMLと数学問題データベースのコラボレーションを使って、独自に数学科目におけるオンライン小テストや問題演習、試験の作成、試験解答の公開などを行ってきた。このシステムは我々自身の必要性から作成したものであり、身近な要請に沿って発展してきたものである。

本論の目的は、数学教育におけるMathMLと数学問題データベースの活用可能性と現状での問題点を本校での使用事例を通して論ずることにある。

数学問題データベースQDBとその使用事例

使用事例1：オンライン小テスト

水産大学校においては2006年度から2,3年間、学校の方針の一環として、いくつかの基礎科目の授業に対する補習授業を学生が受けるかどうかを判別するための小テストを科目ごとに行うこととなった。

著者らが担当する科目である基礎解析学I（微分積分の基礎的内容）もその対象になっていたため、その方法を検討した。まず、1年次科目なので4月のオリエンテーションかその直後に小テストを行い、その結果次第で補修授業を受けるかどうか決めるため、早く採点結果を出す必要があった。そこで、記述式はあきらめ、穴埋め式のテストにすることにしたのであるが、それでも200余名分の採点には時間がかかると思われた。そこで、テストはコンピュータ上でオンラインで行い、提出された解答データをリアルタイムで採点し、すぐ結果を表示する仕組みを作成しようと考え、作成にとりかかった。

作成にとりかかったのは3月であるが、ウェブの仕組みを基本に、MathMLと $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 及びRDB（リレーショナルデータベース）を用いて比較的短時間で作成できると見込んだ。オンライン小テストシステムの基本的要点は以下のとおりである。

1. 問題とその解答は後利用など利用性を考慮してRDB (MySQL) に格納し管理する。問題は $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ コードで書かれデータベースに格納される（詳しく

は後述）。

2. 全体としてはウェブアプリケーション（サーバはLinuxベース、スクリプトはPerlで記述）であり、学生はブラウザでこのアプリケーションにアクセスする。各学生は、初期画面で自分に指定されたクラスを選び問題画面（すべてのクラスで同一）に入る。
3. データベースから取り出された小テスト用の穴埋め問題はMathML含むXHTMLコードに変換されブラウザ上に表示される（詳しくは後述）。受験学生は穴埋めを行い（解答をして）提出ボタンを押して提出する。下は、問題ページの一部である（ウィンドウは小さくしてある）。

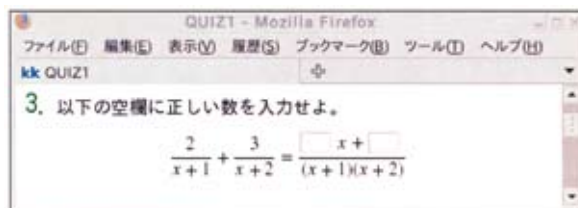


Fig. 1. The window of the online quiz

4. 提出された解答はウェブサーバに送られ、データベースの正しい答えと照合され採点結果（各問ごとの正誤と総合得点）がブラウザに戻される。

MathMLコードは直接手入力するには冗長すぎるので、入力には専用のエディタを使うか、他のコードで入力しておいてMathMLコードに変換するかという方法を採用。ここでは、数式の記述に関しては定評のある $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ を使用している。大学の数学教員には $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ はかなり普及しているので、問題と解答の入力に $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ を用いることは大学の数学教育の場ではある程度の一般的妥当性を有すると思われる。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ からMathMLへの変換プログラムは、`matexpl`⁷⁾という公開されたPerlコード（オリジナルは現在ウェブ上で在り処が不明）に少し手を入れたもの⁸⁾を用いた。

数式内の穴埋め解答欄は、`{i1}`というコードを $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の数式コードに埋め込んで、MathMLに変換するときこれを`<html:input name="i1"/>`というタグに置き換えることによって実現される。単なる`<input>`タグではMathML内では入力欄にならない。穴埋め部分が複数なら、`{i1}`, `{i2}`, `{i3}`, ... というコードを埋め込めばよい (Fig. 1, Fig. 2を参照)。穴埋め問題を登録するとき解答として`i1==3`などPerlで評価 (eval) される式を登録しておい

て、解答者の送信する値を解答のパラメータ ($i1$ など) に代入して評価することで正解・不正解を判断する。

上記の数学の問題と解答を格納したデータベースが本題の数学問題データベースのプロトタイプである。本事例の時点では最小限度の機能と少量の問題しかなかったが、その後機能、問題数とも増やしていったものである。この本論で対象とする数学問題データベースをQDBと呼ぶことにする。QDBの基本的構成は以下のとおりである。

1. QDBはRDBのテーブル群problem, category, quiz, quiz_submitと、それらを管理するウェブアプリケーションからなっている。
2. categoryは問題が属するカテゴリを格納している。
3. problemのレコードは問題と答えとヒントを含む。問題は $\text{L}^{\text{T}}_{\text{E}}\text{X}$ で記述されて格納される(ただし、preambleと $\backslash\text{begin}\{\text{document}\}\backslash\text{end}\{\text{document}\}$ はなし)。
4. ウェブ上で表示するためにQDBから問題を引き出して使うときには、 $\text{L}^{\text{T}}_{\text{E}}\text{X}$ からMathMLに変換するプログラムを使う。
5. quizテーブルは、いくつかの問題からなる試験(本システムではクイズと呼ぶ)を格納しているテーブルである。quiz_submitは試験解答の提出記録である。
6. 管理用のウェブアプリケーションは、問題の作成・更新・削除、試験の作成・更新・削除、問題・試験の表示、解答・採点の表示、試験の集計などをブラウザのインターフェイスを使って行うものである。Fig. 2は、このアプリケーションの問題作成画面の一部である(図では見えない部分に解答とヒントを記入する欄がある)。

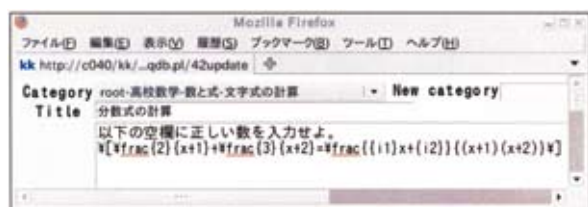


Fig. 2. The window for the problem authoring which produces the problem of Fig. 1

2006年度の小テストは、4月の12, 14日の二日に分けて水産大学の二つのコンピュータ教育用の教室(計約110台のコンピュータからなる)で行われた。

教室のOSはWindows XPで、ブラウザはMathMLへの

対応からFirefoxをあらかじめMathML用のフォントをインストールした上で指定した。初めて教室を使う一年生に教室の使い方を説明した上で、オンラインの小テストにとりかかった。問題数は20問、時間は90分の枠内で40分以上とれた。各学生には提出は1回かぎりであることを告げてあったので、提出した学生はその採点結果を確認して退出した。コンピュータのモニターとキーボードがある机の上で、計算を紙上でしながらではあったがさほど解答しにくそうには見えなかった。

集計結果は、受験者数212人、平均点74.03(100点満点)であった。個人の解答データもデータベースに残っているので確認できる。無効な解答データはなく、オンライン小テストが無事に終えたことを示している。

前もって3週間ほどの準備を要したものの、本番はトラブル無く進み、学生が解答提出直後に採点結果を知るといった目的も完全に果たされた。また、平均を出すなどの集計もデータベースの機能を使って簡単にできた(後掲のFig. 3に見える[クイズ結果]ボタンを押して集計したい小テスト番号を指定するだけ)。

2006年に引き続き、2007, 2008年にも同じ問題で年度初めに小テストが1年生に対して行われた。同じ問題で行ったのは、入学者の学力の年度による違いがあるかを見るという学校の方針の一環でもあった。ちなみに、2007年度の平均は68.47, 2008年度の平均は67.84であった。

本事例においては、 $\text{L}^{\text{T}}_{\text{E}}\text{X}$ コード内に $i1$ などの独自コードを埋め込みこれをMathML内の $\langle\text{html}:\text{input}/\rangle$ タグに変換することで、ウェブページの数式の中に穴埋めの解答欄を埋め込むことが可能になった。筆者らが現在においても重宝している $\text{L}^{\text{T}}_{\text{E}}\text{X}2\text{HTML}^9)$ を用いて $\text{L}^{\text{T}}_{\text{E}}\text{X}$ コードをHTMLページに変換しても数式は画像になるので、数式の中に穴埋めの解答欄は埋め込めない。その場合数式の中には変数で書いて、数式の外にその変数の値を問う解答欄を設けることになる。これでは、多くの場合問題がかなり見づらくなる。また、動的に数式を含むHTMLページを生成する場合、付帯する数式画像ファイルのファイル名を重複なく決めるにはアクセスするユーザごとに画像ファイル名を区別するなどせねばならない。これはサーバー側の処理も複雑化し、本事例のシステムの開発期間に大きく影響する。システムのシンプルさは我々のような教師レベルでの開発では重要である。

PDFで問題ページを表示することもできるが、この場合も数式内穴埋め解答欄ができないので、問題と解答欄と

の関連がよりいっそう見づらくなる。

また、本事例において、問題、小テスト及び解答データはデータベースに格納されているので、小テストの問題の組み換え、小テストの結果の集計には大変便利であった。本事例において数学問題データベースとMathMLは効果的に使用されたといえるだろう。

QDBシステムの後利用としては、2006年度は当面、基礎解析学 I の自習用に、オンラインで問題を提示し、穴埋め問題に対しては採点をし、それ以外の問題に対しては解答を示すという機能を作り、基礎解析学 I の問題も登録した。しかしこの自習用問題演習システムは2006年度の基礎解析学 I の授業と平行して開発されたもので学生にはあまり使われなかった。

2007年度には次年度(2008年度)からカリキュラムが大幅に変わることになり、データベースの問題は新カリキュラムの科目にはそぐわないようになり、開発は中断された。

使用事例2：試験問題作成と解答公表

数学問題データベースQDBにおいて旧カリキュラム用の問題から新カリキュラムに即した問題を増やすうち、2010年度には、中間・期末試験をこのデータベースを使って作成すること、およびその解答の発表もデータベースから行うことを考えた。

小テスト用にすでに、複数の問題を指定して一つの小テスト(クイズ)として登録でき、登録されたクイズを $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ソースに落としてそのままコンパイルして印刷できるような仕組みはできていた(Fig.3参照)。これを中間・期末に使うのは問題さえ揃えば大変便利であることが分かった。使いたい問題の問題番号をコンマ区切りで書き、クイズとして登録したらtexsourceボタンを押して TeX ファイルとして保存する。このままでもコンパイルは通るが手を入れてから platex にかけてもよい。問題作成の手間も $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ コーディングの手間も省けて非常に短時間で小テストや定期試験のマスターが刷り上がる。



Fig. 3. The window for quiz manipulations

問題と解答の発表は、データベースに登録したクイズのページを各問解答表示ボタン付きで表示する機能を作成して、基礎解析学のウェブページ中のリンクから表示されるようにした(Fig.4)。



Fig. 4. The Calculus page

この中間試験の解答発表の見方については、MathML表示環境に依存するのでコンピュータ教室で説明(出席は自由)を行った。

中間試験の解答を発表してから中間試験の再試までの期間(6月22日から7月9日)において試験のページへのアクセスは、学内から198ヒット、学外から161ヒットであった(無論ここには著者らによるアクセスは含まれていない)。学生一人当たり何回かは同じ試験ページへアクセスすると考えれば、人数的にはヒット数の数分の一であろう。

学生は解答を見ようと思えばコンピュータ教室(夕方まで授業以外は開放)で見ることができるのであるが、理想としては自宅でもオンラインで参照できることであろう。

学外からアクセスしたIPアドレスの数は46であったが、DHCPを経ているとすれば同一学生が別のIPで何度かアクセスすることもありうる。実際アクセスしてきたIPにはほとんど同じものが見られる。本校学生の約半数が自宅からインターネットを使える環境にあるといわれている。中間試験受験者は299人、再試験対象者は127人であるから、学外からのアクセス率は高くなかったと思われる。

QDBでは試験の $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ソースを保存できるようにはなっているが、答まで含めて $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ に落とせるようには

なっていなかった。L^AT_EXソースに答をエディタを使って加えてDVIを作りPDFファイルに変換することは時間の関係でしなかった。

考 察

事例1では、MathMLコードにhtml:inputタグを埋め込んだ数式を使用することで、短い開発期間、並びにオンライン小テストのスムーズな実施及びその場での採点という当初の目的を果たすことができた。しかも、MathML使用のおかげでシステムが単純化したことは当初の開発期間の短縮化に役立っただけでなく、以後のシステムの発展・活用の上でも役に立つであろう。

本論のQDBシステムは、第一著者が4月に間に合わせるために急いで作ったものが元になっているが、第二著者もシステムの数々の制限にもかかわらずわずかな説明を聴いただけですぐに使えるようになった。第二著者もL^AT_EXユーザであること、システムがシンプルなこと、さらにサンプルがすでにあったからであろう。学生側はもちろん、教師の側でもL^AT_EXユーザであれば簡単に使いこなせるシステムであるといえる。

事例2では、解答の公開ではMathMLをめぐる環境がまだ十分ではないことが分かったが、登録された問題から試験を構成できるQDBの機能と、問題データをL^AT_EXコードで持っていることが試験問題の作成・印刷にとって大変便利であることが分かった。

QDBシステムの当面の問題点とその解決方法について考えてみる。

1. 本システムはMathML関連でブラウザを選ぶ。MathMLによる数式が表示可能なブラウザでもブラウザごとに異なった環境整備（フォント追加、プラグイン追加など）が必要である。ウェブブラウザのMathMLサポートがまだ十分でないことは、教室における環境整備にとって負担になるし、学生が自分のパソコンで自主的に取り組むときの障害になる。また、MathML3のドラフト¹¹⁾の6.4を参照すると、“...その困難故に、デフォルトではMathMLの仕様は他言語の要素がMathMLの表現の中に入ることを許していないが、特定の複数言語混在文書用に、混在タグ使用がwell-denedでそのためのツールの開発が正当化されるような場合に、緩いスキームも用意されている”とある。とすると、<html:

input/>タグをMathMLコード内に埋め込んで解答入力欄を表示する本システムの方法は、特殊な（Firefox独自の）方法ということになるのかもしれない。

教室において教師による簡単なガイダンスのもとでなら理想的にMathMLベースの小テストを行えることから、現状のMathMLのブラウザ依存がなくなれば簡単なガイダンスさえ不要になり、学生の自主的な取り組みも期待でき申し分ない。

MathMLフォントが標準で装備され、また多くのブラウザでMathMLがサポートされ、なるべく共通のinputタグ埋め込み方法が確立されることを望みたい。

2. L^AT_EXコードをMathMLコードを含むXHTML(xml)コードに変換するプログラムは2006年当時も複数あったし、今(2010年)はもっとある。しかし、今でも任意のL^AT_EXコードを変換できるものはない。結局、L^AT_EXのサブセットを対象にすることになるが、このサブセットもプログラムによりまちまちである。現状QDBで使用しているmatex.plは使えるタグはあまり多くはないが十分実用的である。高校数学、微積分、線形代数の範囲で表すことのできなかった問題や解答は今のところない。ともあれ、どのプログラムを使うのであれどのタグなら使えるのかを知る必要がある。

itex2MML¹²⁾というL^AT_EXからMathMLへの変換プログラムは、itexというL^AT_EXのサブセットを提示してそれをサポートしている。コマンドラインのほかRubyからも使えるようになっている。math_ml.rb¹³⁾は類似の機能を持った純Rubyのプログラムである。LaTeXMathML¹⁵⁾は、JavaScript版（オリジナル）とPerl版がある。

MathMLのFAQ¹⁶⁾によれば、“MathML入力のためのシンタックスに対する要求は非常に広汎にわたる”、“さまざまなワーキンググループのメンバーが特定のツール用に入力シンタックスを開発している”、“入力シンタックスはcore MathML recommendationには含まれない”とある。よって、L^AT_EX類似の言語がMathML入力手段の標準になることはないだろう。しかし、L^AT_EXは数式入力システムとして妥当性があるだけでなく、QDBに登録した試験問題のL^AT_EXソースを落として定期試験のマスターとして

印刷するという場合に大変便利である。数式の表示品質はMathMLよりL^AT_EXの方が（少なくとも今のところ）高い。L^AT_EXソースを編集もできるから、QDBが完璧なマスターを作り出す必要はない（PDFなどで出力するのとの違い）。QDBシステムにとってL^AT_EXからMathMLへの変換プログラムとしてどのようなものが最適なのかを引き続き探っていきたい。

3. データベースに格納される問題数が増えてくると、ほとんど同じ問題同士をどうするかということを考えざるをえない。これら似た問題が増えてくると非常に冗長である。
- 例えば、つぎの不定積分を計算せよ、という問題があったとする。

$$\int xe^x dx$$

この問題のバリエーションとして、

$$\int xe^{-x} dx$$

という問題を使いたいとき、これを別の問題として登録するのは冗長であるということである。

これには、次のような処理が考えられる。問題を

$$\int xe^{p_1} dx$$

とし、 p_1 に依存する解答とともに登録する。実際に出題するときは、 p_1 の値を決めて出題する。RDB的には問題テーブルのほかに問題番号とパラメータ値からなる出題テーブルを設ければよいだろう。しかし、解答における p_1 を含む式は簡単な式（ p_1+2 とか）ならPerlで評価できるが、複雑な式ならせいぜい p_1 を単にその値に置き換えるか、数式処理系を使うことになる（それでも評価できない式は残るだろう）。システムが複雑化する上に完璧な処理は無理そうではあるが、複雑化の度合いを抑えつつ実用的な解決方法を探っていきたい。

4. 同種の問題をいくつか集めて大問とすることがある。大問中の小問と同じ問題が別に重複されて登録されているのはこれも冗長である。これには、大問というカテゴリを設けることが考えられる。例えば、

次の計算をせよ。

$$2^3 4^6$$

という問題と、

次の計算をせよ。

$$\sqrt{3}\sqrt{12}$$

という問題があって、これらを小問とする大問を登録して、

次の計算をせよ。

$$(1) 2^3 4^6 \quad (2) \sqrt{3}\sqrt{12}$$

と表示できるようにする。小問のレイアウトをどうするかは一般には難しいだろう。また、

次の計算をせよ。

$$2^3 4^6$$

という問題と、

次の計算をせよ。

$$\sqrt{3}\sqrt{12} \text{ を簡単にせよ。}$$

という問題はどうかまとめるのか。いろいろ制限は残ろうが、それを知りつつ使えばそれなりに便利ではあるだろう。

W3Cの仕様書やドラフトで認めているように、仕様の解釈や実装があまりにも複雑になるような機能は標準化不可能である。そして、特定の個々の応用において必要とされる機能をその応用のために個別に開発せざるをえない。また、QDBの機能もあまりに汎用性を求めるのは却って開発を難しくしシステムを複雑にする。しかし、上記のQDBの問題点はいずれも実用上重要であるので制限があってもなんらかの解決策は必要である。

QDBの開発は、RDB, L^AT_EX, Perl, Linux, Web, MathMLなどの1970年代以降開発され公開されてきたテクノロジーに負っている。これらのテクノロジーがあってこそ、ユーザレベルでユーザーズに応じたデータベースシステムの開発が可能になっているのである。

これらの中では新しいテクノロジーであるMathMLは数式表示（Presentation Markup）だけではなく数式の内容を記述する仕組み（Content Markup）も持っているなど、そのポテンシャルは現時点のQDBのMathML活用をはるかに越えている。たとえば、数式の検索¹⁷⁾や、（上述したような）数式処理との組み合わせによる動的な解答の作成などが考えられる。こうしたMathMLの活用を考えるなら、QDBがMathMLでデータを持つようにすることも

検討せねばなるまい。

数学教育を主なターゲットとして、教師としての視点から、システムをなるべくシンプルに保ちながらQDBを改良していきたい。

参考文献

- 1) MathML2.0仕様書, <http://www.w3.org/TR/MathML2/>, (2003).
- 2) XML1.0仕様書 (Fifth Edition), <http://www.w3.org/TR/2008/REC-xml-20081126/>, (2008).
- 3) KIT数学ナビゲーション, <http://w3e.kanazawa-it.ac.jp/math/>.
- 4) Journal of the Mathematical Society of Japan, <http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&>
- 5) 問題データベース 中学校 数学, 東京書籍.
- 6) WebALT - Web Advanced Learning Technologies, <http://www.webalt.com/>.
- 7) 田中研太郎 (matex.pl の作成者), <http://www.me.titech.ac.jp/miyalab/tanaken/index.html>.
- 8) matex.pl (modified), <http://c040.sh-u.ac.jp/kk/matex.pl>
- 9) LaTeX2HTML, <http://www.ctan.org/tex-archive/support/LaTeX2HTML/>.
- 10) HTML5 (Working Draft), <http://www.w3.org/TR/HTML5/>, (2010).
- 11) MathML3 (Working Draft), <http://www.w3.org/TR/MathML3/>, (2010).
- 12) itex2MML, <http://golem.ph.utexas.edu/~distler/blog/itex2MML.html>.
- 13) math.ml.rb, <http://www.hinet.mydns.jp/?mathml.rb>.
- 14) 黒田拓, MathMLライブラリの開発と今後の展開について, 京大数理解析研講究録, 1572, 142-155, (2007).
- 15) A Brief Description of LaTeXMathML, <http://math.etsu.edu/LaTeXMathML/>.
- 16) MathML FAQ, <http://www.w3.org/Math/mathml-faq.html>, (2002).
- 17) 岸本 et al. MathML を用いた類似数式検索方式の実現, DEWS2003 論文集, 6-P-07, (2003).

