Authoring Tool for Mathematical Documents – Infty –

Toshihiro Kanahori*[1], Mitsushi Fujimoto†[2] and Masakazu Suzuki‡[3]

[1]*Research Center on Educational Media, Tsukuba College of Technology, 4-12 Kasuga, Tsukuba 305-0821, Japan*

[2]*Department of Information Education, Fukuoka University of Education, 1-1 Akamabunkyo-machi, Munakata 811-4192, Japan*

[3]*Faculty of Mathematics, Kyushu University, 36, Fukuoka 812-8581, Japan*

Abstract

This paper describes an authoring tool for mathematical documents. Users can input mathematical formulae by handwriting. Another inputting interface is our original front-end processor utilizing LaTeX commands. The written mathematical formulae are calculated by various computer algebra systems and output into files in the notation of MathML, LaTeX, HTML and in Braille codes. Providing these interfaces, the system realizes very easy intuitive methods to input mathematical formulae into a computer requiring no special skills; for example, about the MathML notations of mathematical formulae.

1. Introduction

We previously presented an integrated system for scientific documents including mathematical formulae [1] and [2]. Improvements of recognition rates and user interfaces have been obtained during the intervening period, and we recently released the OCR software 'InftyReader' to transcribe printed mathematical documents into digital data, and the editor 'InftyEditor' to edit mathematical documents and convert them into the various formats of MathML, LaTeX, HTML and Braille codes on web sites [7] and [8]. In this paper, we describe the outline of the Infty system as an authoring tool for mathematical documents utilizing mathematical user-interfaces.

*E-mail: kanahori@k.tsukuba-tech.ac.jp

†E-mail: fujimoto@fukuoka-edu.ac.jp

‡E-mail: suzuki@math.kyushu-u.ac.jp

The user interfaces of current computer systems are not convenient to input mathematical formulae; for example, the widely used data format LaTeX requires some learning to master the notation system, and it is not easy to understand the meaning of the written formulae at a glance from the LaTeX source. To realize an easier treatment of mathematical formulae in various formats, we have been developing a new handwriting interface along with a mathematical front-end processor to input mathematical formulae into a computer. Users can input mathematical formulae by handwriting in the *handwriting dialog*, and edit the input formulae on InftyEditor (**Figure 1**). Using our mathematical front-end processor, users can input various mathematical symbols and formulae with just a keyboard (**Figure 2**). The edited mathematical documents are saved into our

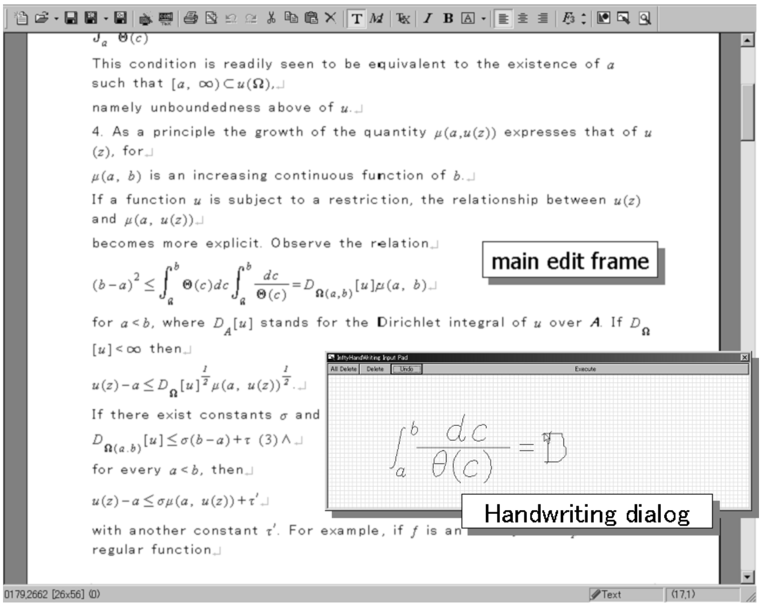

Figure 1: Snapshot of InftyEditor and handwriting dialog

original XML file (**Figure 3**) which can be converted into a file in any of the various notations of MathML, LaTeX, HTML[§] or in Braille codes, UBC (Unified Braille Codes) for English texts and Japanese Braille Codes for Japanese texts. InftyEditor can load the XML and LaTeX files, but we are still developing a loader for MathML.

Furthermore, mathematical formulae written on InftyEditor can be calculated using various computer algebra systems with an OpenXM communication controller that can absorb differences among the various computer algebra systems. Users don't have to learn complex commands and operations in the cases of computing for mathematical formulae.

[§]In the HTML file, mathematical formulae are saved as image files.

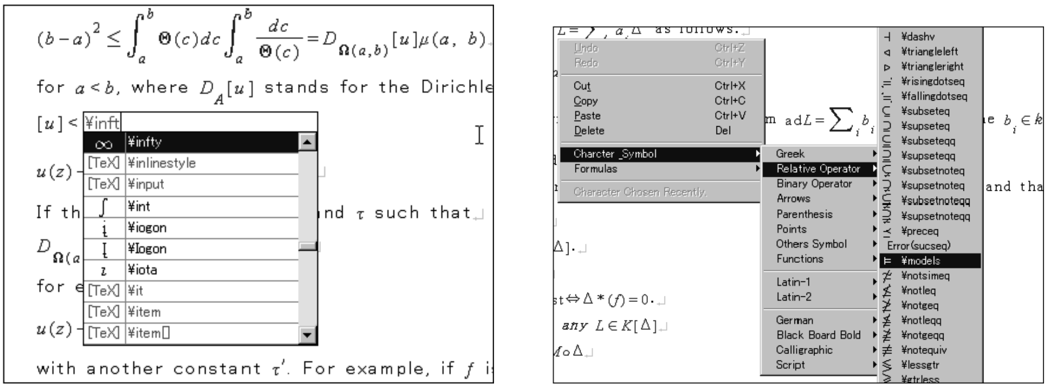

Figure 2: Snapshot of mathematical front-end processor (left side) and symbols menu (right side)

Presently, this system can recognize all the alphanumeric characters, almost all Greek letters, and the other symbols frequently used in mathematical formulae, and can analyze the formulae used in high school mathematics or in first year university mathematics, including fractions, square roots, subscripts, superscripts, integrals, limits, summations and the function names 'lim', 'log', 'cos', 'sin', 'tan', etc. For the moment, matrices are excluded. The structure of mathematical formulae may be nested. However, a deeply nested structure leads to an increase of small size characters and naturally increases errors of recognition.

The InftyReader is InftyEditor equipped with 'InftyOCR', an OCR engine, which can recognize various mathematical symbols (much more than the handwriting interface can) and mathematical structure (**Figure 3**). InftyOCR recognizes scanned images of pages of clearly printed mathematical documents and outputs the results of its recognition in the XML format. Users can seamlessly edit and convert the recognition results with InftyEditor. Using this system, it is possible to author scientific documents including mathematical formulae in the notation of the above formats without any real skills in or knowledge of the formats.

2. Similar Systems

Some similar software systems already exist; one is the commercial Scientific WorkPlace (see `http://www.mackichan.com/`), another one is the free software GNU TEXmacs (see `http://www.texmacs.org/`).

Scientific WorkPlace adopts unicode and TrueType fonts; therefore being a multilingual system. The software also includes a MuPAD kernel to compute mathematical formulae. TEXmacs is a wysiwyg (what you see is what you get) text editor for TEX or LATEX. Users can extend the functions of TEXmacs by using the extension language GUILE/SCHEME. TEXmacs can also be used as a graphical front-end for computer algebra systems PARI, Macaulay2, Maxima,

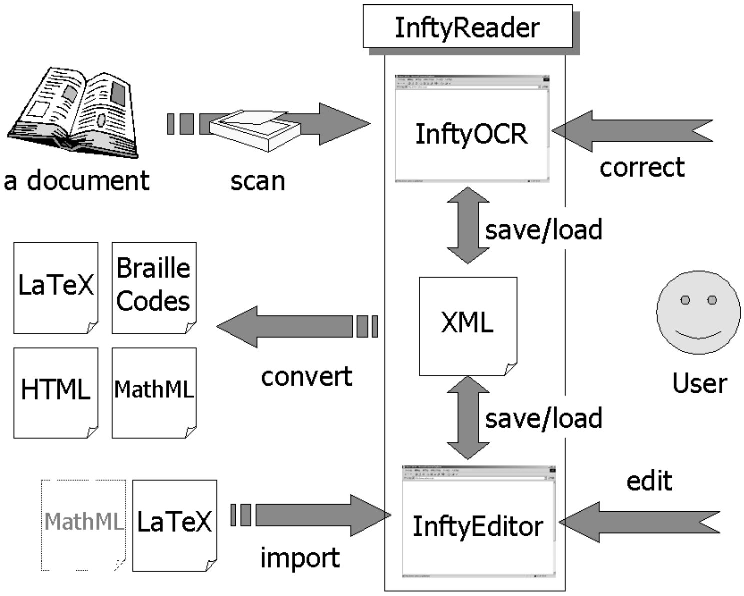

Figure 3: Overview of the Infty system

MuPAD, Maple and so on. However, neither software system has a handwriting interface for mathematical formulae.

3. Handwriting Interface

The outline of the handwriting formula recognition is orthodox, almost the same as [3]'s. However, there are two important points of originality in our method:

1. For character recognition, two individual handwriting character recognizers are used and the recognition results are determined by a vote.
2. To reduce errors in mathematical formula recognition (e.g. identification of individual handwritten symbols, mathematical structure analysis, and so on), the '*Automatic Rewriting Method*' has been developed and utilized here.

In this section, we describe the outline of our method of character recognition in our handwriting interface. The detail of the method is shown in [4].

One of the recognition methods uses the distribution of 8-direction elements of the strokes on 3×5 meshes of a character rectangle. The other method uses matching of segmented stroke sequences. Each of the two recognition methods returns three ordered candidates with costs. The *voting cost* of a candidate is taken to be the ratio of its cost to the third candidate's, and the final results of the recognition are determined by the ascending order of the sum of the two voting costs of the two recognition methods.

We introduce the outline of the original method we propose, the ‘*Automatic Rewriting Method*’. Distortion of input characters and the turbulence of the positions or the scales of the characters usually cause serious difficulties in a structural analysis of mathematical formulae, in which the positions and scales of the characters have special meanings. An error in character recognition or of the segmentation of the strokes into character units leads sometimes to fatal errors in the structural analysis of mathematical formulae. The labor then involved for the correction of this kind of error seriously disturbs the smooth input of mathematical formulae by an author.

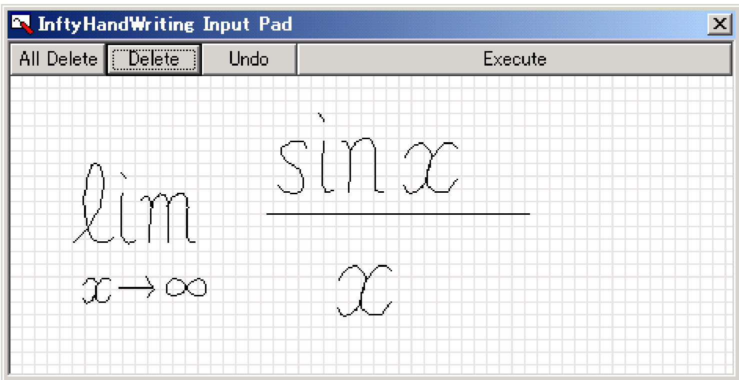

Figure 4: Infty Handwriting Dialog

Our *Automatic Rewriting Method* is introduced to overcome this difficulty. In this method, whenever the pen is up, the strokes are recognized. Each recognized character is automatically rewritten by neat strokes in an appropriate size and position (**Figure 4**). From this rewriting, the user can immediately identify each recognition error as it occurs, and thus can easily correct it.

The algorithms to determine characters and to select appropriate positions and sizes of the determinate characters are described in [4].

4. Command-based Interface

Displayed texts and mathematical formulae can be freely edited using ordinary editing operations such as cut, copy, paste and delete; and for mathematical symbols, our original front-end processor with LaTeX commands is provided. For example, a fraction ‘$\frac{1}{x^2+1}$’ can be input using just a keyboard in the following way:

1. input ‘`\frac`’, then a fractional line appears and the InftyEditor cursor moves to the numerator position of the fractional line,
2. input ‘1’, and press the enter key, then the cursor moves to the denominator position,

3. input ‘x’ and ‘^’, then the cursor moves to the superscript position of ‘x’,
4. input ‘2’, then the cursor moves to the right hand side position of ‘x^2’,
5. input ‘+’ and ‘1’, and press the enter key, then the cursor moves to the right hand side position of ‘$\frac{1}{x^2+1}$’, and the input is completed.

Users can input mathematical symbols using the symbols menu instead of LaTeX commands (**Figure 2**).

Some demonstration movies can be downloaded from the Infty Project web site [7].

5. Communication with Computer Algebra Systems

The previous version (Ver.1) of InftyEditor could communicate with Mathematica using MathLink [5]. In the current version (Ver.2), we planned to build the environment to use various computer algebra systems including Mathematica from InftyEditor. However, some differences made this difficult: (1) Each of the computer algebra systems has its own user interface. (2) Notations of mathematical expressions are different. (3) They use different function names, command names and user languages. (4) Some of them have a communication function, and others do not. Thus, we needed an infrastructure for the various mathematical software systems. We adopted OpenXM [6], which has the specification that defines the communication protocol and model in order to absorb the differences among various computer algebra systems, and its communication model is a client-server model. At present, mathematical software supporting OpenXM are Asir, kan/sm1, Mathematica, gnuplot, phc, TiGERS, Macaulay2, etc.

The main OpenXM server for InftyEditor is Asir; while the other OpenXM servers are called through Asir. The communication function of InftyEditor is provided as Windows Dynamic Link Libraries (DLL) and consists of an Infty–Asir translator, an OpenXM communication controller and a dialog box, which is used to execute calculations and to obtain results.

Figure 5 expresses the computing processes. The left side is for InftyEditor and the right for the OpenXM servers. Inputted data is converted to an Asir string; transmitted to Asir by the OpenXM protocol. Asir computes it (using other OpenXM servers if required), and converts the computation result to the Infty XML format. It is transmitted to InftyEditor and output in the dialog box. The user then pastes it on the editor.

Currently available computation functions are as follows:

- Evaluate (Asir)
- Summate (Asir)
- Expand (Asir)
- Factorize (Asir)
- Differentiate (Asir)

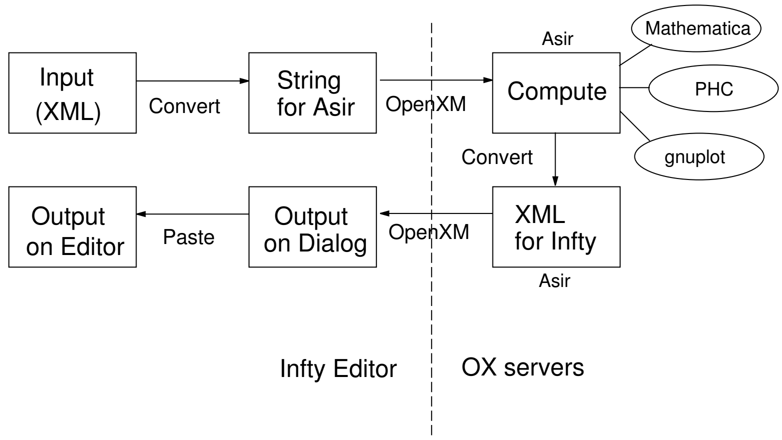

Figure 5: Computing Processes

- Integrate (Asir or Mathematica)
- Solve system of equations (phc)
- Execute Asir command or Asir user language (Asir)
- Plot
  - Explicit Plot: 2D and 3D (gnuplot)
  - Implicit Plot: 2D (ox_plot)

After choosing a mathematical notation, users can send it to the OpenXM servers by choosing a computing command from the calculation menu in the dialog box. The result is then displayed within the dialog box (**Figure 6**). Figure 7 is a snapshot of some calculation results on InftyEditor.

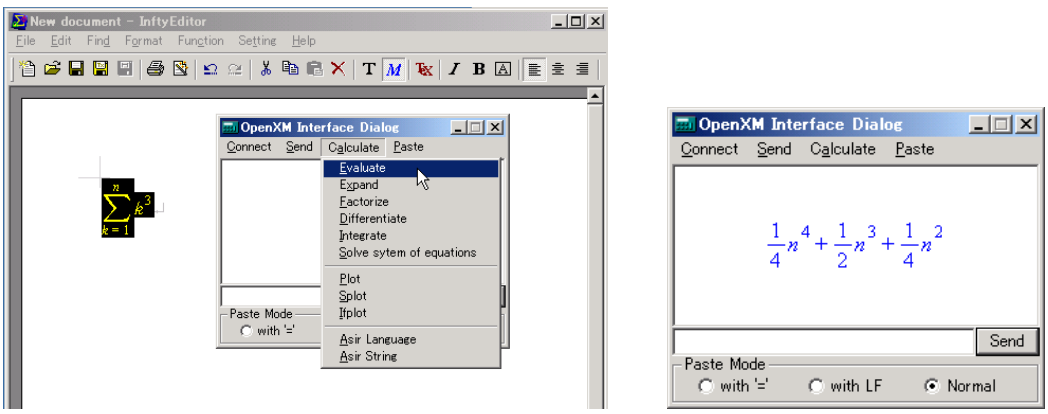

Figure 6: Calculation Menu and Results

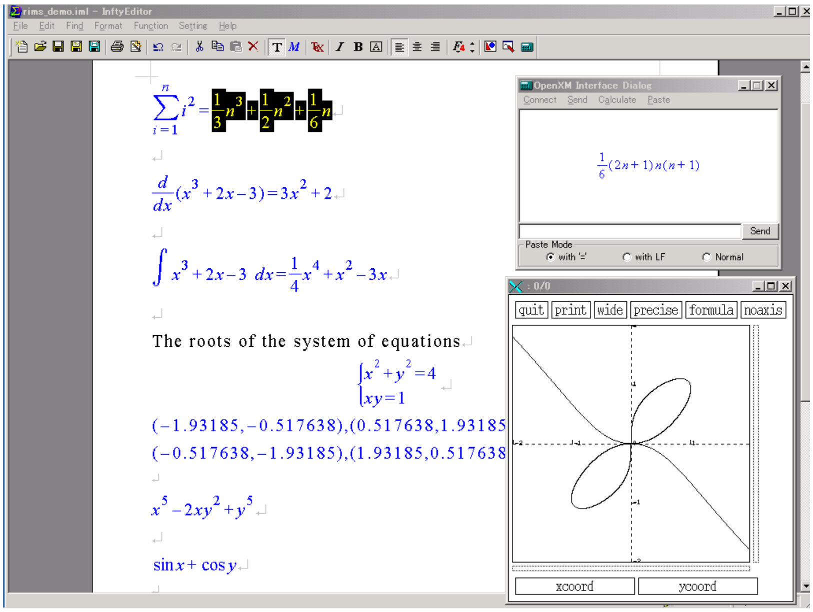

Figure 7: Snapshot of some calculation results

6. Conclusion

In this paper, we introduced our system for scientific documents that include mathematical formulae. We proposed a new handwriting interface and an original mathematical front-end processor as new mathematical user-interfaces. Our Automatic Rewriting Method improves the accuracy of structural analysis of handwritten mathematical formulae and realizes an easy and prompt correction method of recognition errors. We emphasized that an easy method for correction of the recognition results is extremely important to realize smoothly written mathematical expressions. In [4], we observed the efficiency of our method. Our mathematical front-end processor provides smooth keyboard-based input using LaTeX commands. The documents edited by InftyEditor are saved into another original XML format, which can be converted into MathML, LaTeX, HTML or Braille notations. Also, the mathematical formulae edited by InftyEditor can be calculated using various computer algebra systems via an OpenXM framework. Thus, users can translate, calculate and author mathematical documents without any special skills regarding the above formats and computer algebra systems.

7. How to get Infty systems

InftyEditor is a shareware, and can be downloaded from [8]. Additional DLLs for computing functions described in this paper are available from `http://www.InftyProject.org/calc/`. InftyReader can be downloaded freely from our application download site `http://www.InftyProject.org/download/`. For further information, please visit our site [7].

References

[1] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida and T. Kanahori, INFTY – An Integrated OCR System for Mathematical Documents, In *Proceedings of the 2003 ACM Symposium on Document Engineering*, ACM Press, 95–104, 2003.

[2] M. Suzuki, T. Kanahori, N. Ohtake and K. Yamaguchi, An Integrated OCR Software for Mathematical Documents and its Output with Accessibility, In *Computers Helping People with Special Needs (Proc. ICCHP2004)*, Springer LNCS **3118**, 648–655, 2004.

[3] B. Wan and S. Watt, An Interactive Mathematical Handwriting Recognizer for the Pocket PC, *MathML and Technologies for Math on the Web*, The 2nd MathML International Conference 2002, `http://www.mathmlconference.org/2002/`.

[4] T. Kanahori, K. Tabata, W. Cong, F. Tamari and M. Suzuki, On-Line Recognition of Mathematical Expressions Using Automatic Rewriting Method, In *Advances in Multimodal Interfaces (Proc. ICMI2000)*, Springer LNCS **1948**, 394–401, 2000.

[5] H. Okamura, T. Kanahori, W. Cong, R. Fukuda, F. Tamari and M. Suzuki, A Handwriting Interface for Computer Algebra Systems, In *Proceedings of the Fourth Asian Technology Conference on Mathematics*, ATCM, Inc., 291–300, 1999.

[6] M. Maekawa, M. Noro, N. Takayama Y. Tamura and K. Ohara, The Design and Implementation of OpenXM-RFC 100 and 101, In *Computer Mathematics (Proc. ASCM2001)*, World Scientific, 102–111, 2001.

[7] *Infty – Research Project on Mathematical Information Processing*, `http://www.InftyProject.org/`.

[8] *InftyEditor*, `http://www.caisystem.co.jp/infty/e/`.