Framework For Pen-Based Mathematical Applications

Elena Smirnova and Stephen Watt

Ontario Research Centre for Computer Algebra,
University of Western Ontario
Introduction

• Growing popularity of pen-enabled devices such as the *Pocket PC*, *Tablet PC* and *interactive whiteboards* implies the need for handwriting recognition tools, including not only text, but mathematics too.

• Math input on pen-enable devices goes way beyond ordinary hand-written math on paper or regular whiteboard, because it can enjoy rich functionality of the software standing behind ink-capturing hardware.

• This may provide pen-entered math with useful features
  o Editing
  o On-spot validation
  o Directly manipulation
Specifcics Of Pen-Based Math Approach

- larger alphabet
  \( A, A, A, \tilde{A}, a, \alpha, \infty, \ldots \)

- no fixed vocabulary
  \( LambertW(k,x), WrightOmega(z), \ldots \)

- 2-dimensional structures
  \[ x^n, \frac{p}{q}, \sum_{i}^{N} a_i, \begin{bmatrix} a & b \\ c & d \end{bmatrix} \]

- large symbols for grouping
  \[ \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \sqrt{x^2 + y^2 + z^2} \]

- multiple notations
  \( C_n^m \) vs. \( \binom{m}{n} \), tan \( x \) vs. \( \tan x \)

- ambiguous notations
  \( \sin^{-1} x \) (arcsin \( x \) or \( \frac{1}{\sin x} \))
  \( \lg x \) (log\(_2\) \( x \), log\(_{10}\) \( x \) or ln \( x \))

These issues requires a new approach for pen-based software solutions for handling handwritten mathematics
In this poster

• We will not

address the subject of developing specific software for ink-aware math application

• We will

  o Investigate the topic of an *interface* to pen-enable math software
  o Suggest an architectural solution to enable such an interface.
Objectives

• Question we explore:
  If a pen-based interface for math is widely acceptable, how should its architecture be organized?

• Key to the decision:
  Define the target audience that will use this interface:
  o We do not restrict the audience only to math systems users
  o We also include uses of rich text editors and document processors
  o We do not restrict the audience to one hardware/software platform
State Of The Art

- Recently both math software packages and document processing applications have started to comprise ink-enabled features.

- Maple 10 and Word 2003 are good examples of software with basic pen-aware features.

- Pro and cons:
  + In both cases the attempts to enable ink are extremely handy for brief handwritten notes.
  - Both solutions are specific to the software product: they cannot be easily exported and reused in other applications.
  - Neither provides full ink support for handwritten mathematics.
Ink Features In Maple 10

\[ \int \sin(x^2), \ x = 0 \ldots \pi \]

\[ \frac{1}{2} \text{FresnelS}(\sqrt{2} \sqrt{\pi}) \sqrt{2 \sqrt{\pi}} \]  

(1)

N.B. look up what Maple means by \text{FresnelS}

scratchpad

character selector
Pen-Based Interfaces for Mathematics
Research Centre for Computer Algebra

Supported by
Microsoft Maplesoft

overlay inking
ink comments

change to Pen math logo

move this dir to D:\Unix
Available Technologies

• **WACOM driver for tablets**
  + easy to use interface (C++)
  + accessible from Java through JNI adaptors
  - hardware-specific
  - provides too primitive ink handling functionality

• **C#**
  + fully compatible with Tablet SDK API
  + native to Windows platforms
  + have a potential to became portable across platforms
  - cannot be exported as an ActiveX control to run inside MS Office applications
  - cannot be directly use within Maple architecture

• **Tablet SDK**
  + provides high-level support for ink management on Tablet PC
  + supported by .NET framework
  - is not portable across platforms
  - not directly available from Java
  - not available from Maple
Our approach must meet the requirement of **portability**

- **two-dimensional platform portability** of pen-based interface frameworks:
  - across platforms and applications
  - over time for any given (evolving) platform/application

- **digital ink portability**
  - can be achieved with InkML (universal ink format)
  - wrappers for device-specific ink interfaces

- **mathematical data portability**
  - OpenMath
  - MathML
Implementation Languages

- **C#**
  - **assignment**: ink collecting and processing,
  - **example of use**: connecting to Tablet SDK

- **C++**
  - **assignment**: low-level intensive computations
  - **example of use**: character recognizer, glyph feature determiner

- **Java**
  - **assignment**: high-level code for connecting with mathematical engine
  - **example of use**: math expression manipulation
 INVARIANT SOLUTION WITH REPLACEABLE “GLUE”

• Parts remaining invariant:

   A High-level math object manipulation code (Java)
   B Low-level digital ink analysis code (C++)

• Parts, depending on hosting system:

   1 Basic ink collecting software
     ▪ to support abstract ink representation
   2 Interlanguage linking code
     ▪ to connect low level C++ with high-level Java
   3 Interface code
     ▪ to embed pen-based math input in hosting application
Framework Components

③ Interface to Host Application

② Windows XP/ Windows CE/ Linux/ Mac OS X/Palm/... Framework

① Basic Ink collecting software

A JAVA

B C++
Instantiating The Architecture

We have instantiated the architecture for Tablet PC as follows:

① For **basic ink software**
   we used .NET-based Tablet PC SDK

② Specially designed **linkage mechanism** included
   - a number of .NET technologies (C#, managed C++),
   - COM interoperability features and
   - Java Native Interface (as described further)

③ **Interface to the hosting application**
   vary depending on the application
• To test the Tablet PC version of our architecture we use
  o for mathematical computing: *Waterloo Maple*
  o for document processing: *Microsoft Word*

• Then our framework components look like
Possible technologies to use for middleware:

- a number of commercial products (JNBridge, Ja.NET, Janeva)
- IKVM allows to run virtual Java machine inside .NET
- ORCCA architectural approach:

   (no 3\textsuperscript{rd} party software involved)
A solution for the host interface is as follows:

- interface to Maple
  - Java library, accessing COM object through JNI

- interface to MS Word
  - ActiveX control, accessing COM object via Win 32 C++
Pen-Math Interface In Maple 10

> solve(exp(x) > 7*x, x);

\[
\text{RealRange}\left(-\infty, \text{Open}\left(-\text{LambertW}\left(\frac{-1}{7}\right)\right)\right), \text{RealRange}\left(\text{Open}\left(-\text{LambertW}\left(-1, \frac{-1}{7}\right)\right), \infty\right)
\]

\[
\sqrt{1 - \frac{v^2}{c^2}} \frac{m_1 + m_2}{m_1 + m_2}
\]
Then the result we seek is given as

\[ \sqrt{1 - \frac{v^2}{c^2}} \]

\[ \frac{m_1}{m_1 + m_2} \]
• **Current results**
  
  - We have developed a software solution to enable a pen-based math interface on Tablet PC platforms.
  - This is compatible with Maple 10, MS Office (2000, XP and 2003).

• **Ongoing work in**
  
  - plugging recognizer tools to determine
    - structures of math expression
    - math characters
  - enabling math engine features
    - to validate math expressions
    - to allow direct manipulation on math formulae
  - instantiating our solution on other platforms.
Conclusions

• Our goal was
  o design a framework to allow wide use of pen-based math interfaces

• Our requirements were that
  o these interfaces be suitable for both math computing packages and document processing applications
  o the framework
    ▪ provides high-quality ink capturing and handling
    ▪ allows easy access to mathematical engine
    ▪ ensures future portability across and along platforms and applications

• Our results
  o showed feasibility of the goal sought
  o provided an architectural solution to enable an instance of the framework