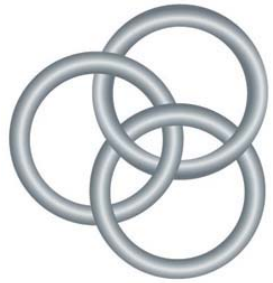
pen•math
@ORCCA



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
  - Editing
  - On-spot validation
  - Directly manipulation

Specifics Of Pen-Based Math Approach

- **larger alphabet**

  A, **A**, ***A***, Ã, *a*, α, ∝, ∞,…

- **no fixed vocabulary**

  LambertW(k,x), WrightOmega(z),..

- **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**

  $\left\langle \begin{matrix} \alpha \\ \beta \\ \gamma \end{matrix} \right\rangle, \sqrt{x^2 + y^2 + z^2}$

- **multiple notations**

  C_n^m vs. $\binom{m}{n}$, $\tan x$ vs. $\operatorname{tg} x$

- **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

Goals

In this poster

- **We will not**

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

- **We will**

  - Investigate the topic of an *interface* to pen-enable math software
  - 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:**

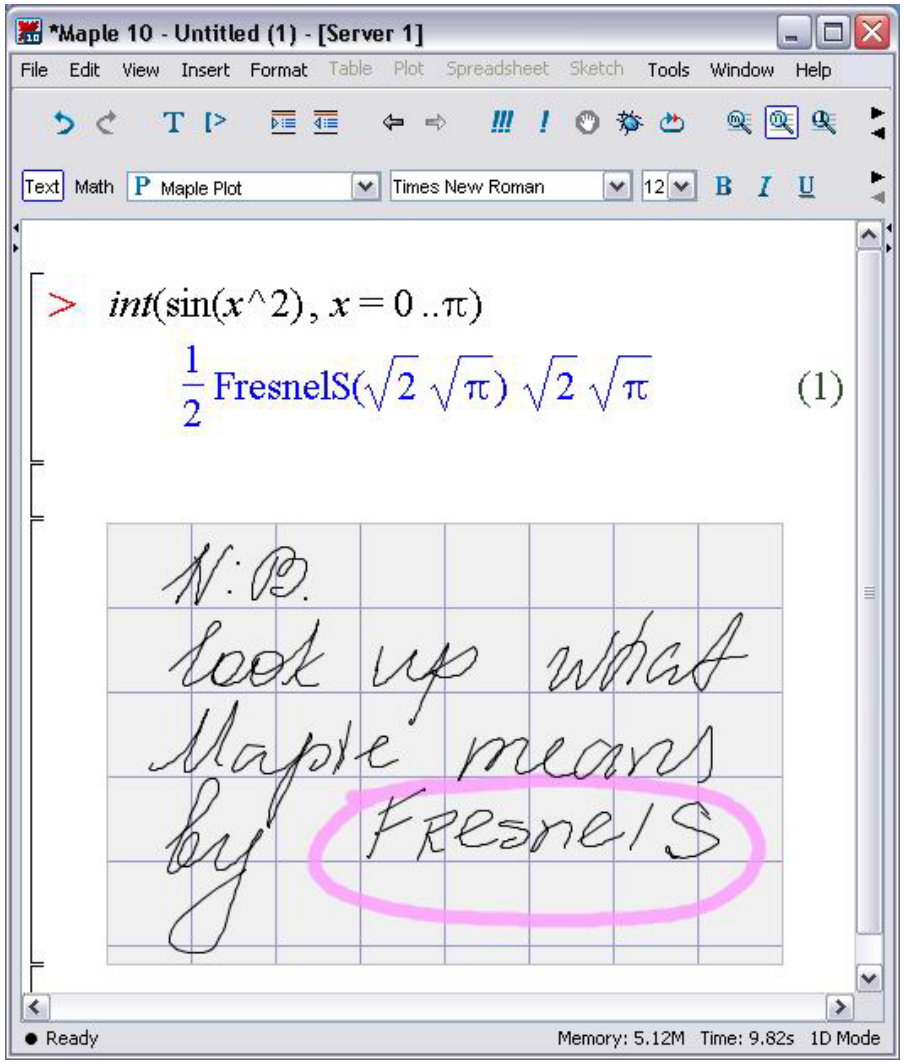
  Define the target audience that will use this interface:
  - We do not restrict the audience only to math systems users
  - We also include uses of rich text editors and document processors
  - We do not restrict the audience to one hardware/ software platform

State Of The Art

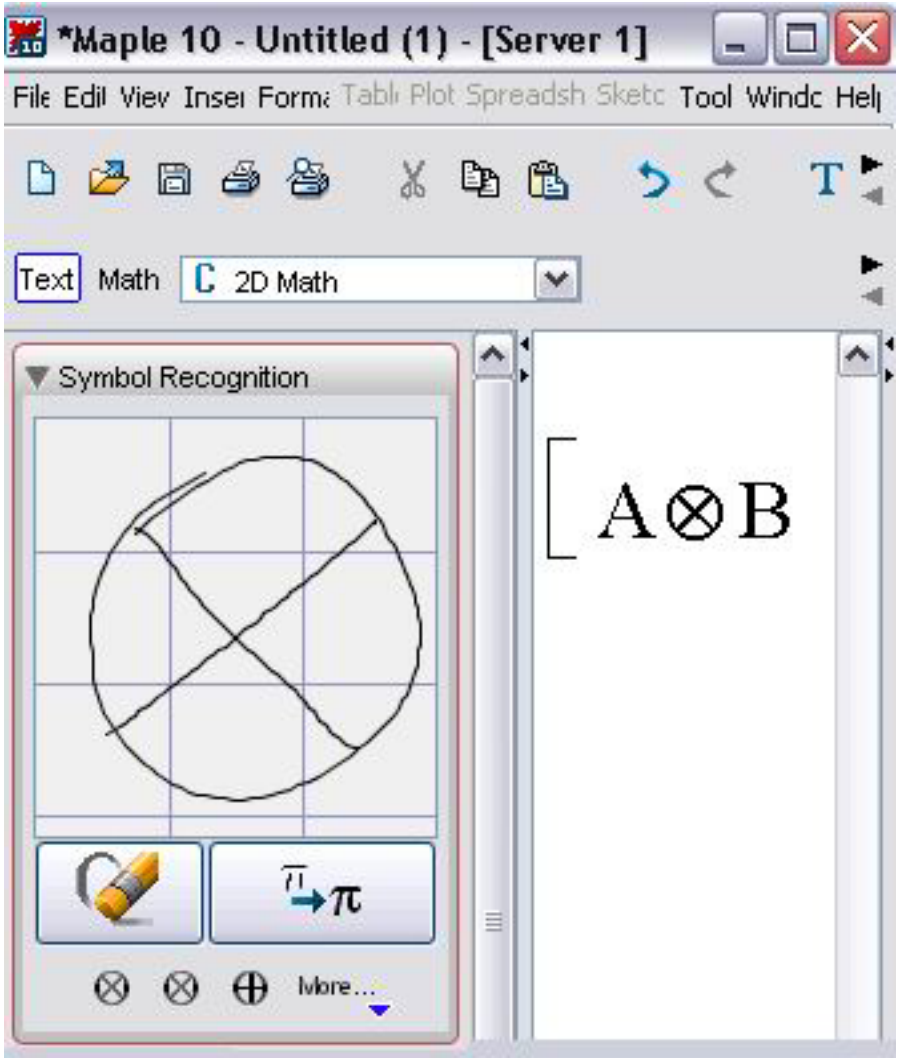
- Recently both math software packages and document processing applications have started to comprise ink-enabled features
- Maple 10 and Word 2003 are good examples of software with basic pen-aware features
- Pro and cons:

+ 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

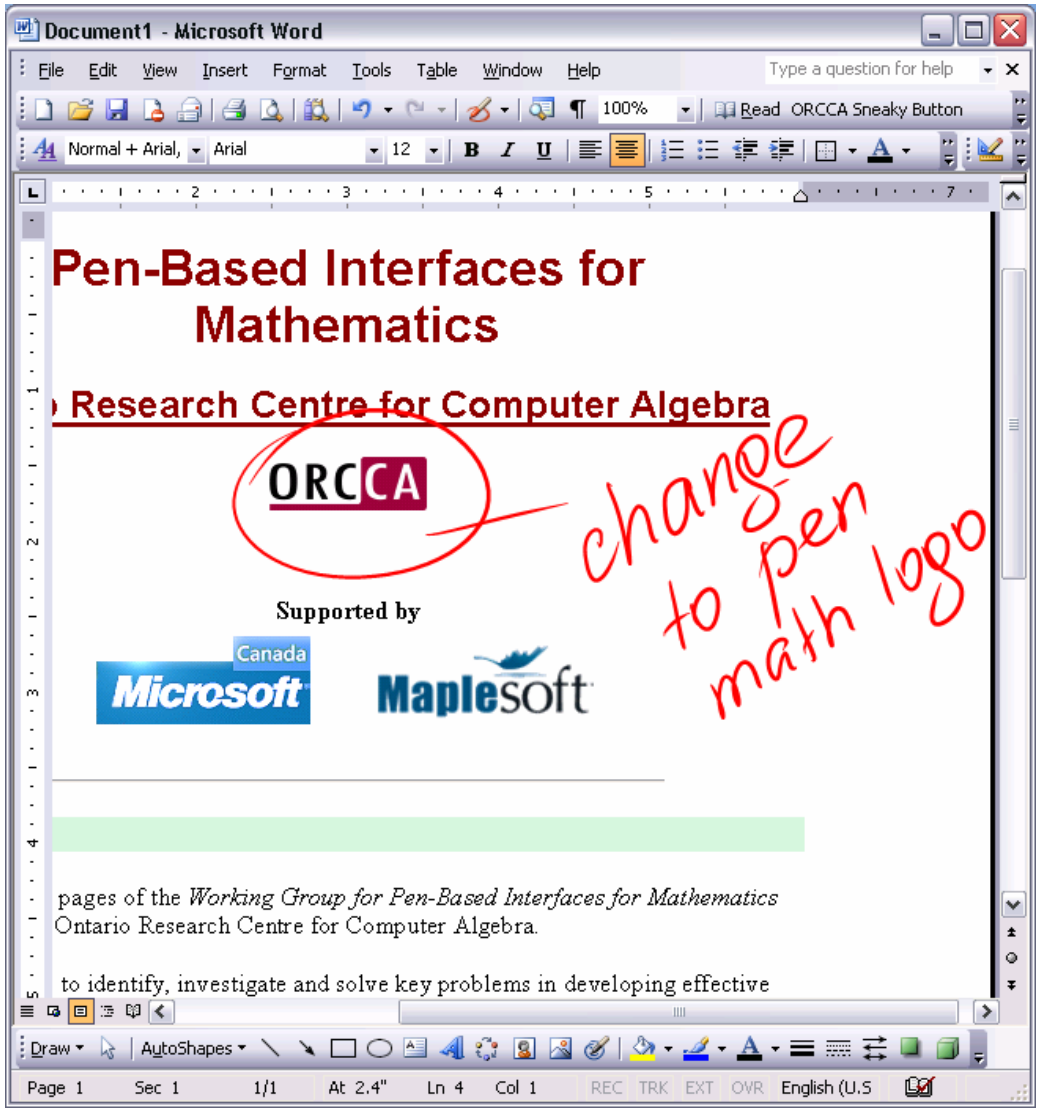


scratchpad

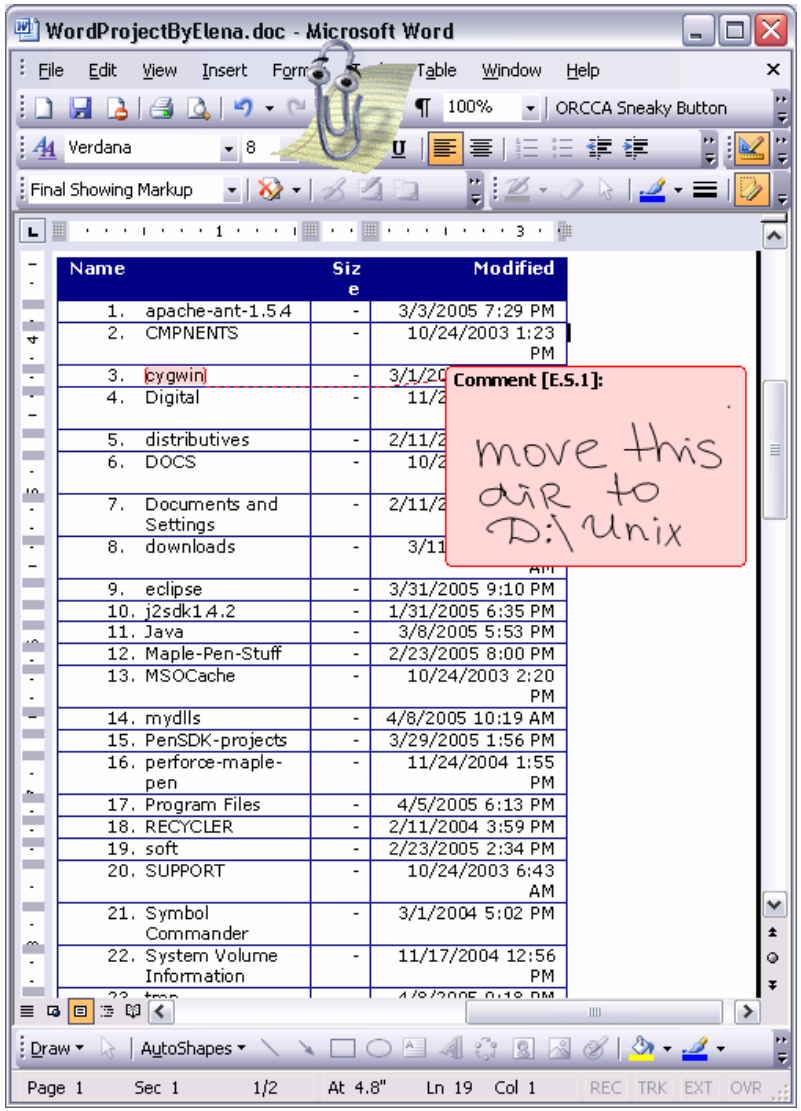


character selector

Ink Features In MS Word 2003



overlay inking



ink comments

Available Technologies

- **WACOM driver for tablets**

+	−
+ easy to use interface (C++)	− hardware-specific
+ accessible from Java through JNI adaptors	− provides too primitive ink handling functionality

- **C#**

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

- **Tablet SDK**

+	−
+ provides high-level support for ink management on Tablet PC	− is not portable across platforms
+ supported by.NET framework	− not directly available from Java
	− not available from Maple

Portability Criteria

Our approach must to 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

Our Architectural Approach

Invariant Solution With Replaceable "Glue"

- **Parts remaining invariant:**

Ⓐ High-level math object manipulation code (Java)

Ⓑ Low-level digital ink analysis code (C++)

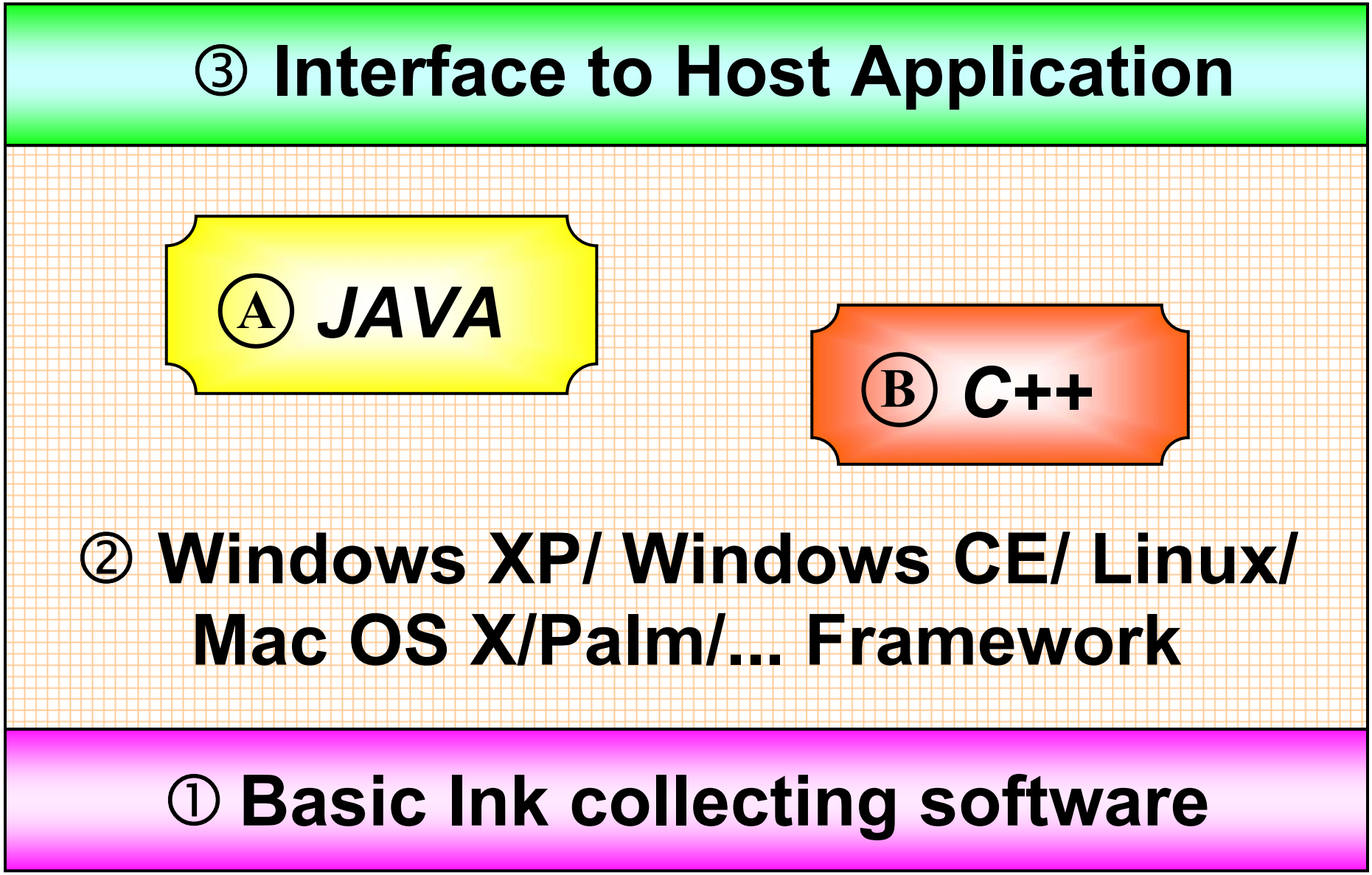
- **Parts, depending on hosting system:**

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



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)

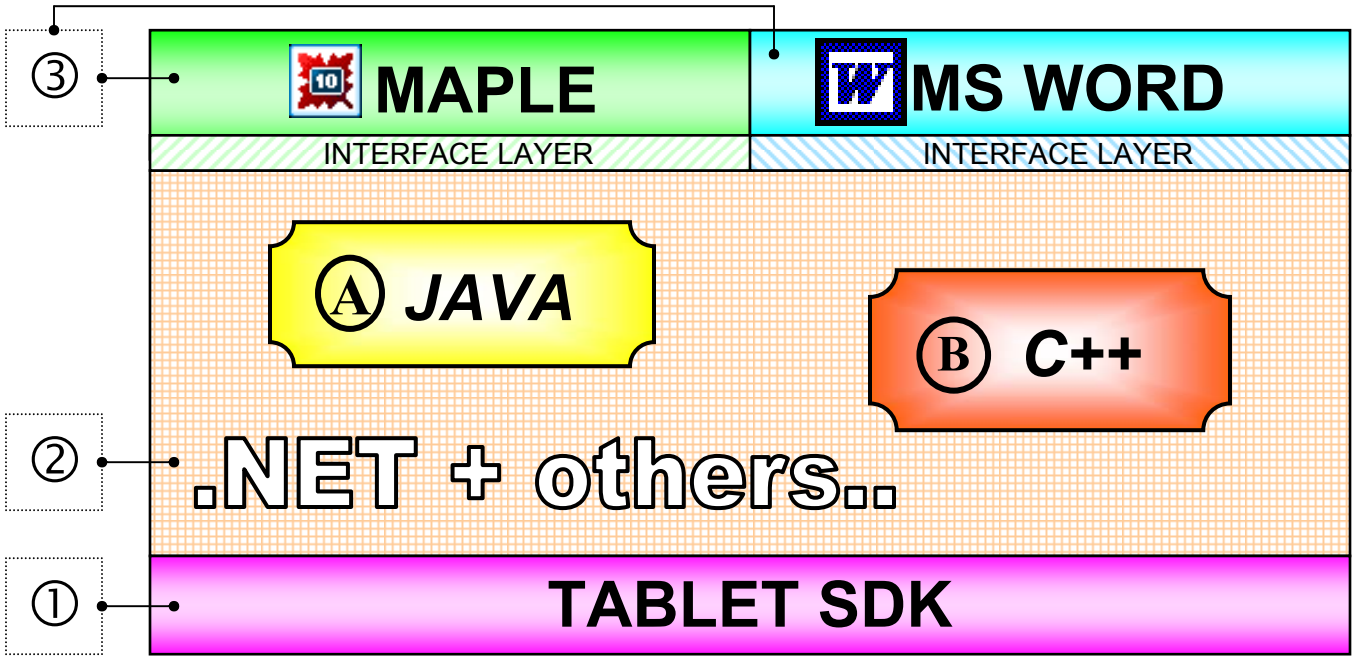
③ **Interface to the hosting application**

vary depending on the application

Testing Framework

- To test the Tablet PC version of our architecture we use
  - for mathematical computing: *Waterloo Maple*
  - for document processing: *Microsoft Word*

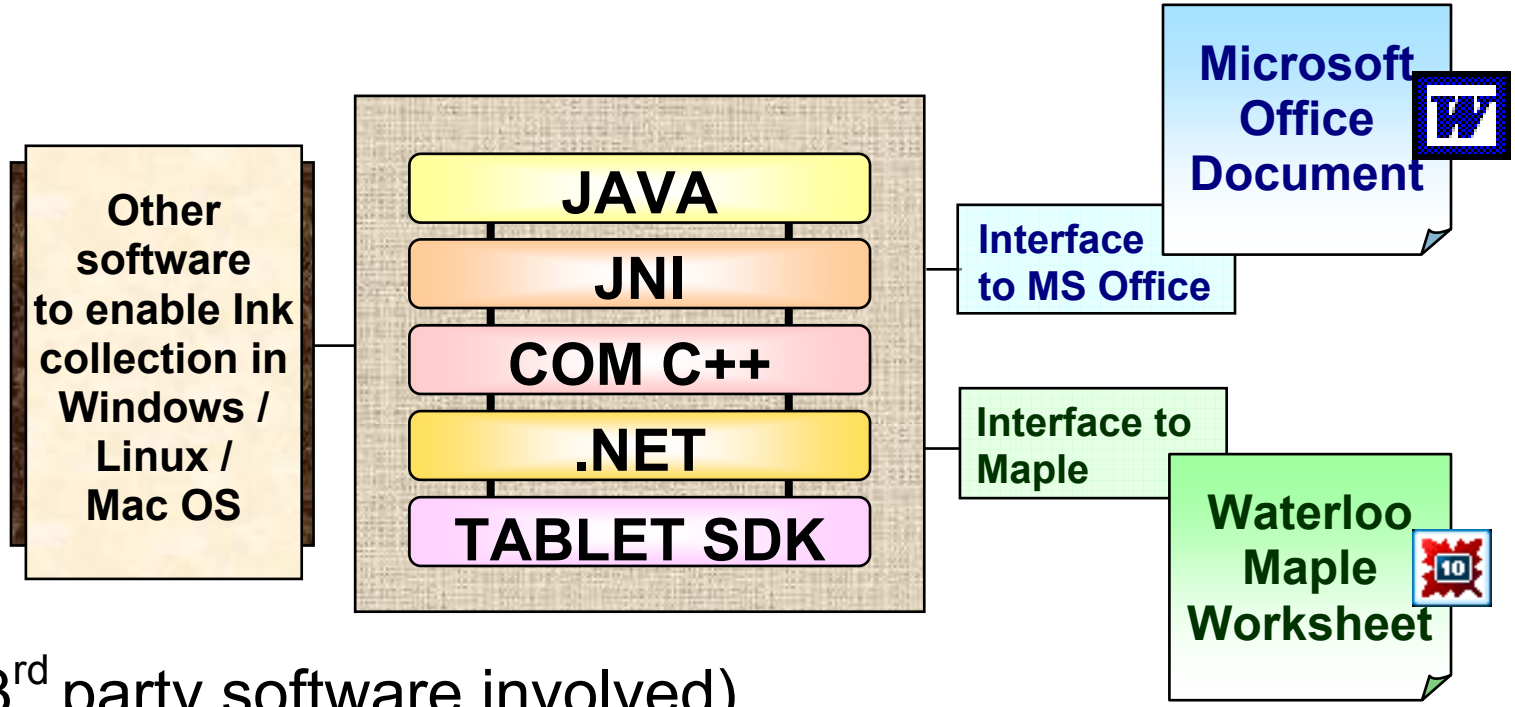
- Then our framework components look like



Linkage For The Test Framework

Possible technologies to use for middleware ②

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

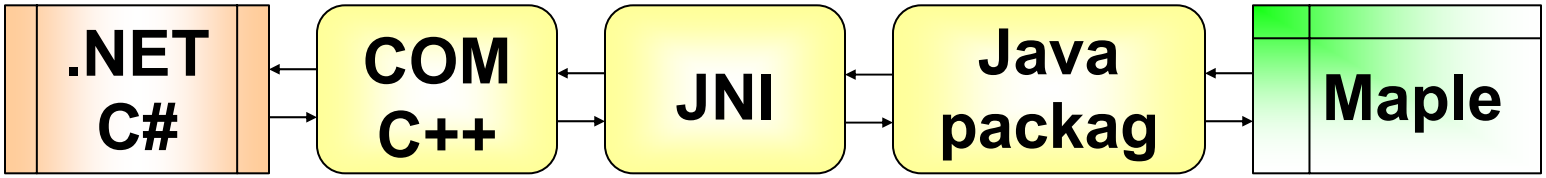


(no 3rd party software involved)

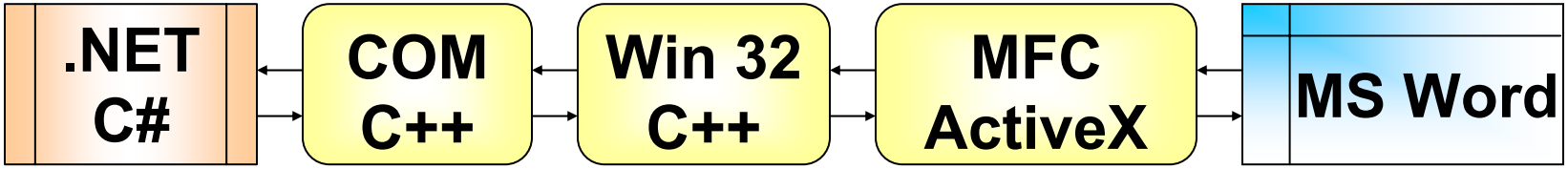
Interface To Hosting Application

A solution for the host interface ③ is as follows:

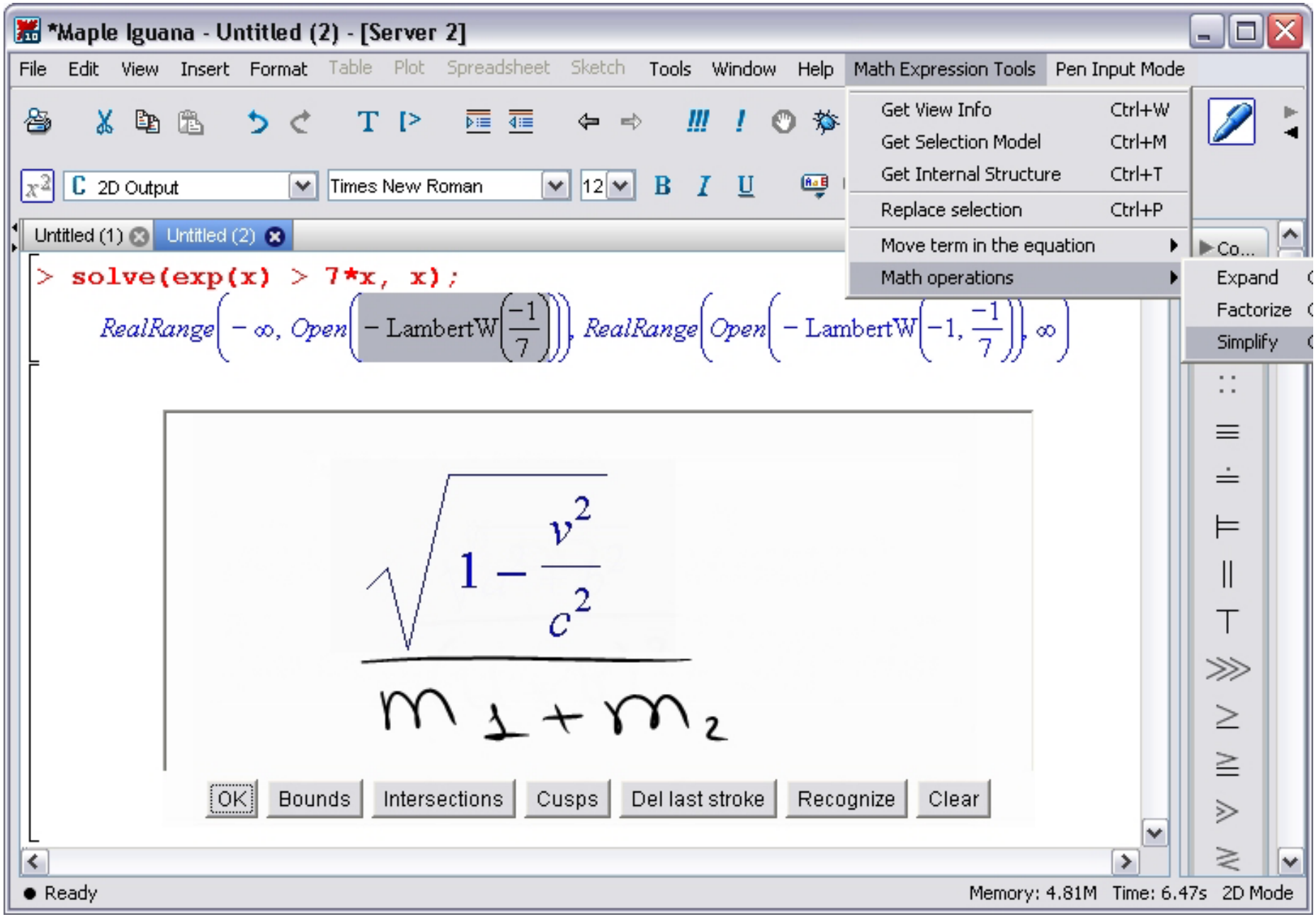
- interface to Maple
  - Java library, accessing COM object through JNI



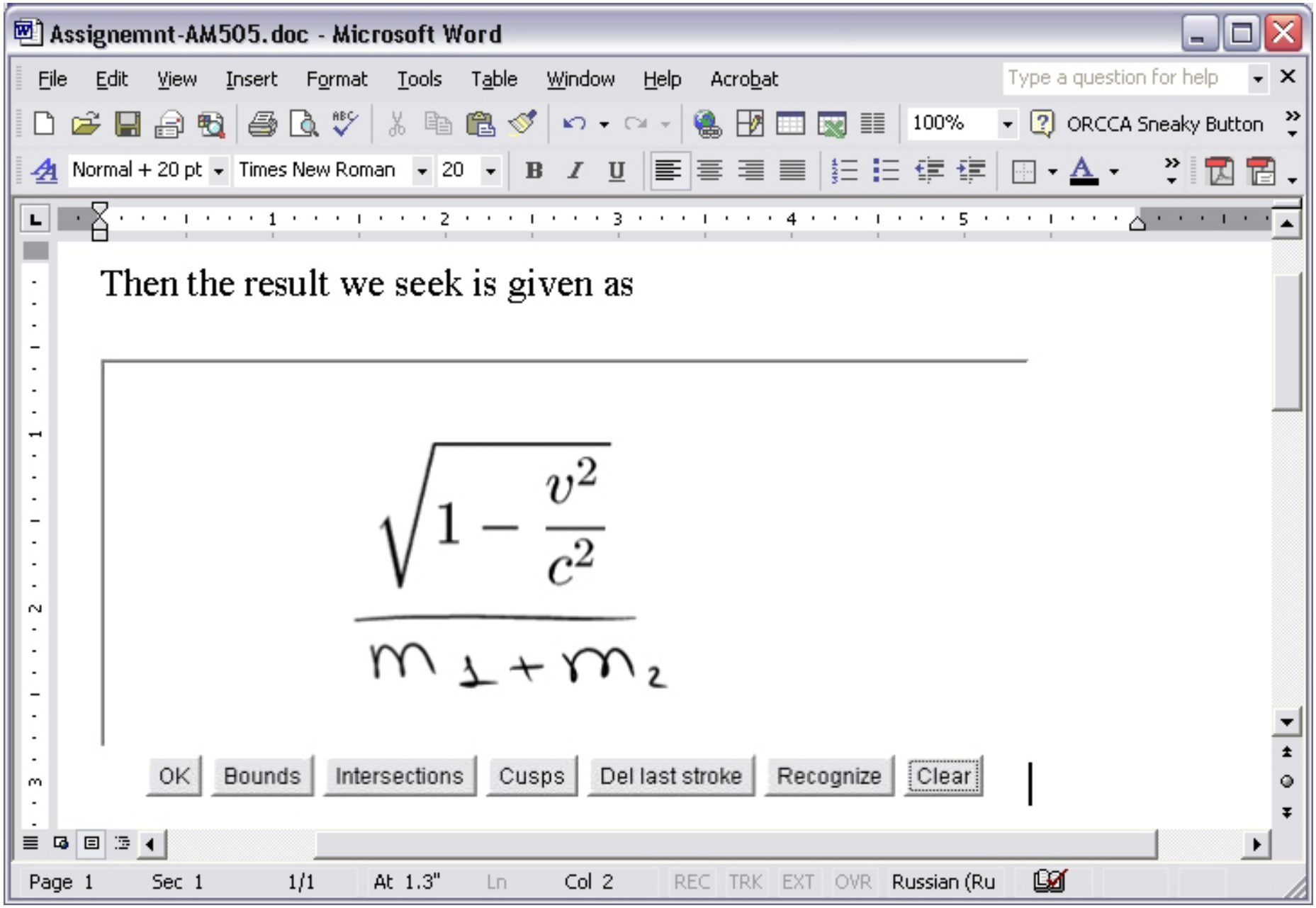
- interface to MS Word
  - ActiveX control, accessing COM object via Win 32 C++



Pen-Math Interface In Maple 10



Pen-Math Interface In MS Word



Status And Future Work

- **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
  - design a framework to allow wide use of pen-based math interfaces
- **Our requirements** were that
  - these interfaces be suitable for both math computing packages and document processing applications
  - 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**
  - showed feasibility of the goal sought
  - provided an architectural solution to enable an instance of the framework