pen math @ORCCA

Western

A Context for Pen-Based Mathematical Computing

Elena Smirnova and **Stephen Watt**

Ontario Research Centre for Computer Algebra,
University of Western Ontario

www.orcca.on.ca/PenMath

Maple Summer Conference 2005

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 chalkboard, 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
  - Direct manipulation

Specifics Of Pen-Based Math Approach

- **larger alphabet**

  A, **A**, ***A***, Ã, a, α, $\propto$, ∞,…

- **no fixed vocabulary**

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

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**

  $$x^n, \quad \frac{p}{q}, \quad \sum_i^N a_i, \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- **large symbols for grouping**

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

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. $\mathrm{tg}\, x$

- **ambiguous notations**

  $\sin^{-1} x$ ($\arcsin x$ or $\frac{1}{\sin x}$)

  $\lg x$ ($\log_2 x$, $\log_{10} x$ or $\ln x$)

Targets for Pen-Based Mathematics

- The above issues require a new approach for pen-based software solutions for handling handwritten mathematics

- Ultimately we wish to have a pen-based platform for
  - mathematical expression entry,
  - mathematical editing and
  - calculation.

$$\frac{LambertW(z)}{z} = e^{-x}$$

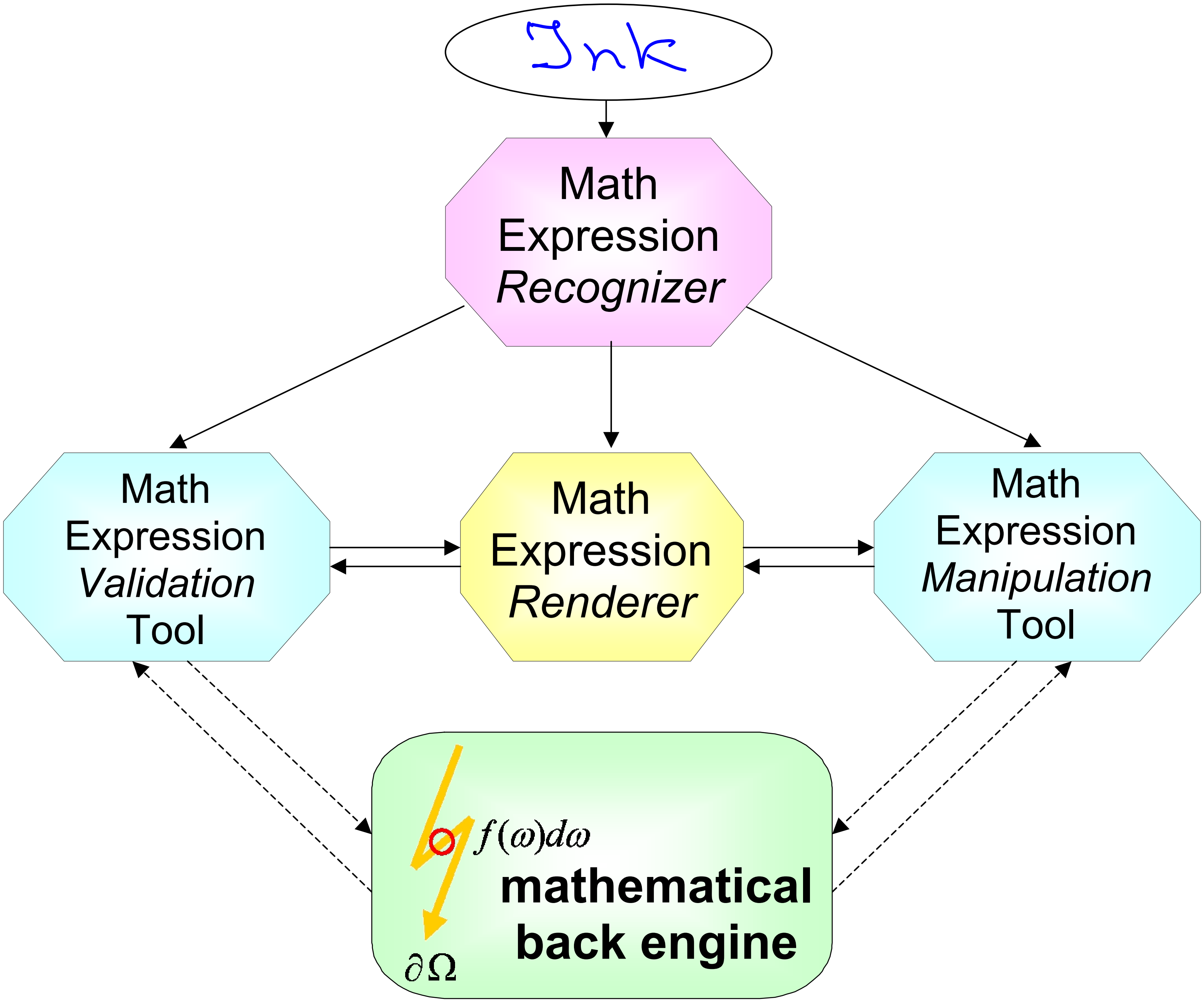
Targets for Pen-Based Mathematics

- The above issues require a new approach for pen-based software solutions for handling handwritten mathematics

- Ultimately we wish to have a pen-based platform for
  - mathematical expression entry,
  - mathematical editing and
  - calculation.

$$LambertW(z) = e^{-x} \cdot z, \quad z \neq 0$$

Interface for Pen-based Mathematics



Goals

In this talk

- **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

Outlines

The remainder of this presentation is organized as follows:

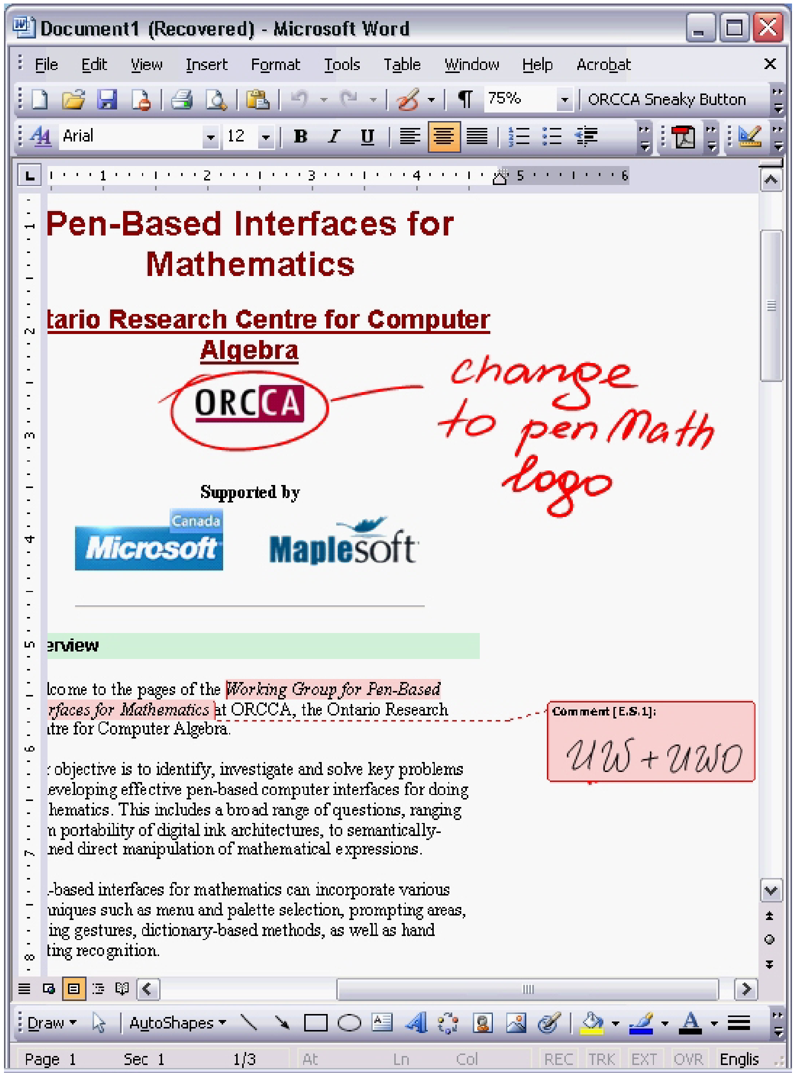
- **Overview of existing Ink technologies**
- **Portability objectives for pen-based frameworks**
- **The large-scale aspects of the architecture designed**
- **Some of the implementation issues**
- **Report on current and future work**

State Of The Art

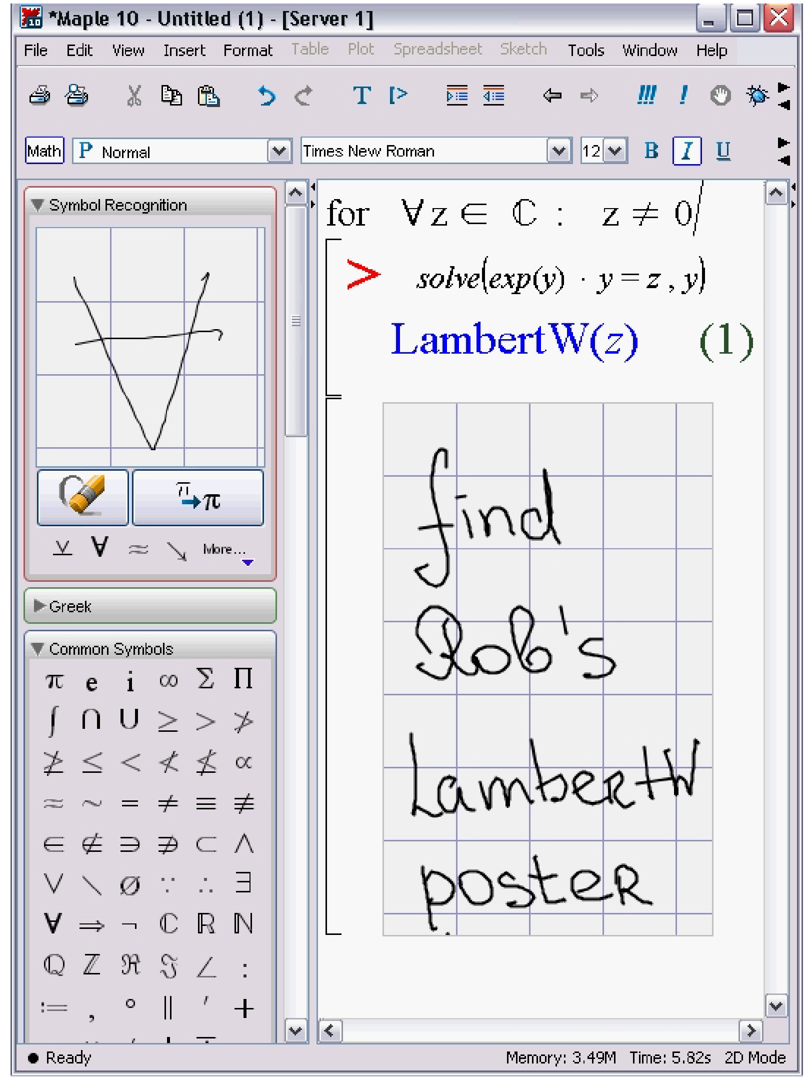
- Recently both math software packages and document processing applications have started to comprise ink-enabled features

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

Ink Features In Maple 10 and MS Word 2003



overlay inking and ink comments in MS Word 2003



scratchpad and character selector in Maple

State Of The Art: 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

Available Technologies

- **Digitizer Device Drivers (such as WACOM)**

+ easy to use interface (C++)
+ accessible from Java through JNI adaptors

− hardware-specific
− provides too primitive ink handling functionality

Available Technologies (2)

- **.NET / C#**

+ fully compatible with Tablet SDK API
+ native to Windows platforms

− cannot be exported as an ActiveX control to run inside MS Office applications
− is not portable across platforms
− cannot be directly use within Maple architecture

Available Technologies (3)

- **Tablet PC 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

Available Technologies (4)

- **Maple**

  - An interface for pen-based mathematics will be required to perform non-trivial transformations on its input.
  - It is inevitable that a pen-based mathematical framework will make use of a computer algebra system
  - For this we find Maple to be a suitable choice

Combining Available Technologies

- Even though these technologies individually provide high-levels of functionality useful in ink applications, none is completely suitable for our needs.
- In particular these elements to not all work together.
- Our solution will need to combine them in such a way that the final architecture
  - can provide **high-quality ink capabilities**
  - remains **portable across platforms**
  - provides **easy connection with applications**.

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

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

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

Our Architectural Approach

INVARIANT SOLUTION WITH REPLACEABLE "GLUE"

- **Parts remaining invariant:**

Ⓐ High-level math object manipulation code

Ⓑ Low-level digital ink analysis code

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

① Basic ink collecting software
  - to support abstract ink representation

② "Glue": Inter-Component Interface
  - to link (A) and (B) with (1) and (3),

③ Interface code
  - to embed pen-based math input in hosting application

Framework Components

③ Interface to Host Application

Ⓐ

High-level math object manipulation code

Java

Ⓑ

Low-level digital ink analysis code

C++

② "Glue": Inter-Component Interface

① Basic Ink collecting software

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

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)

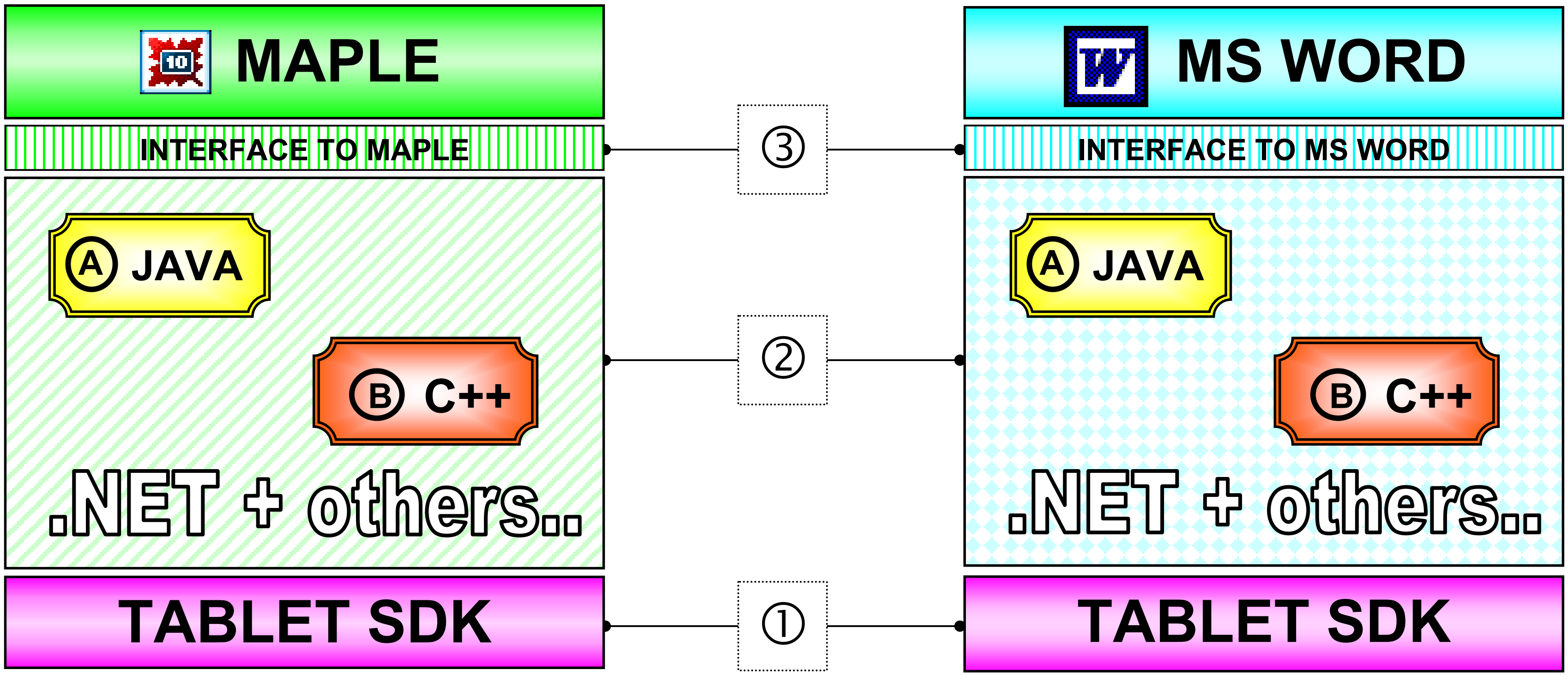
③ **Interface to the hosting application**

vary depending on the application

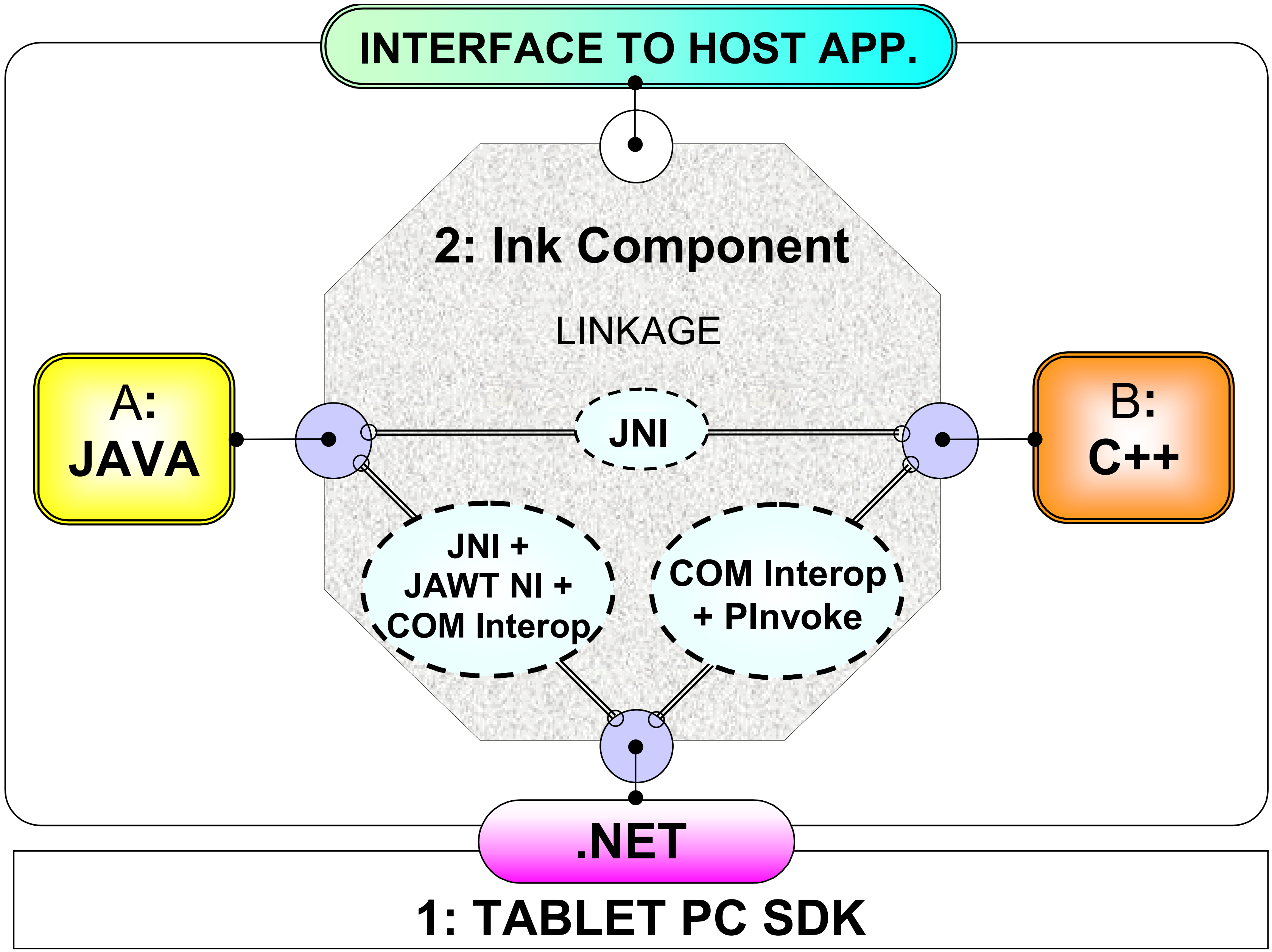
Testing Framework

- 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



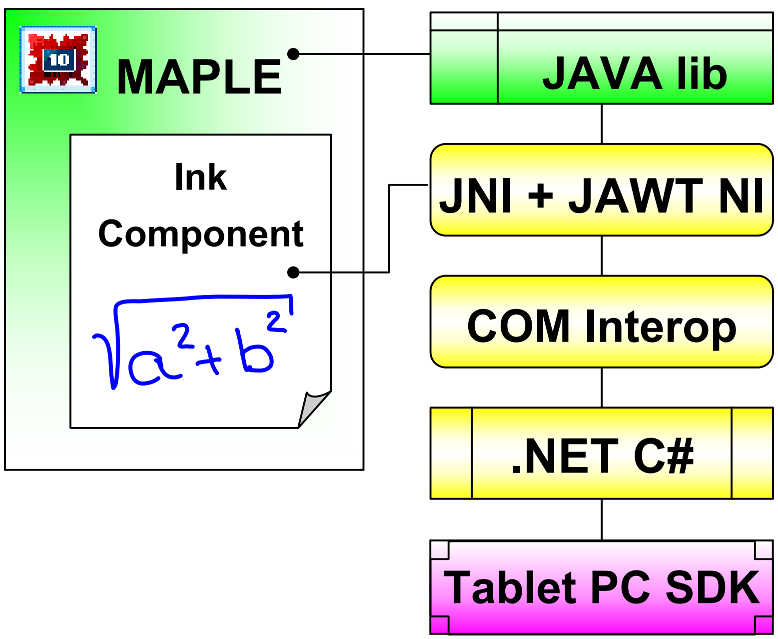
Linkage For The Test Framework



Interface To Hosting Application

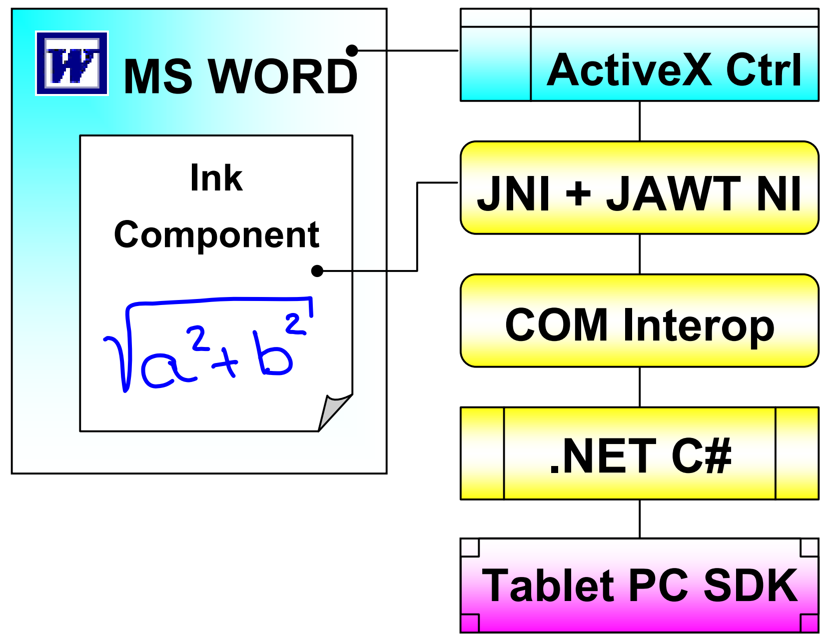
A solution for the host interface ③ is as follows:

- interface to Maple

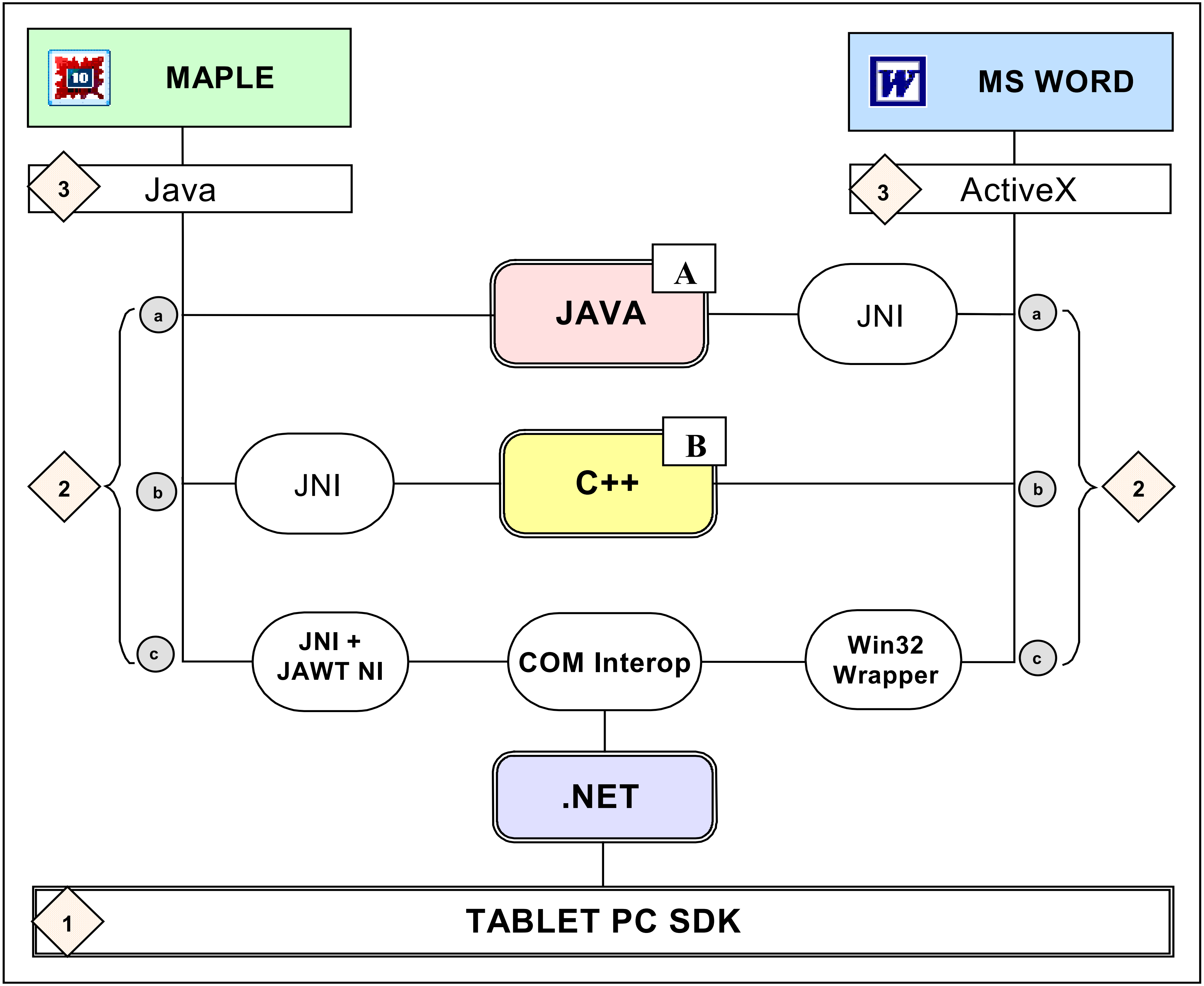


Java library, accessing .NET control through JNI

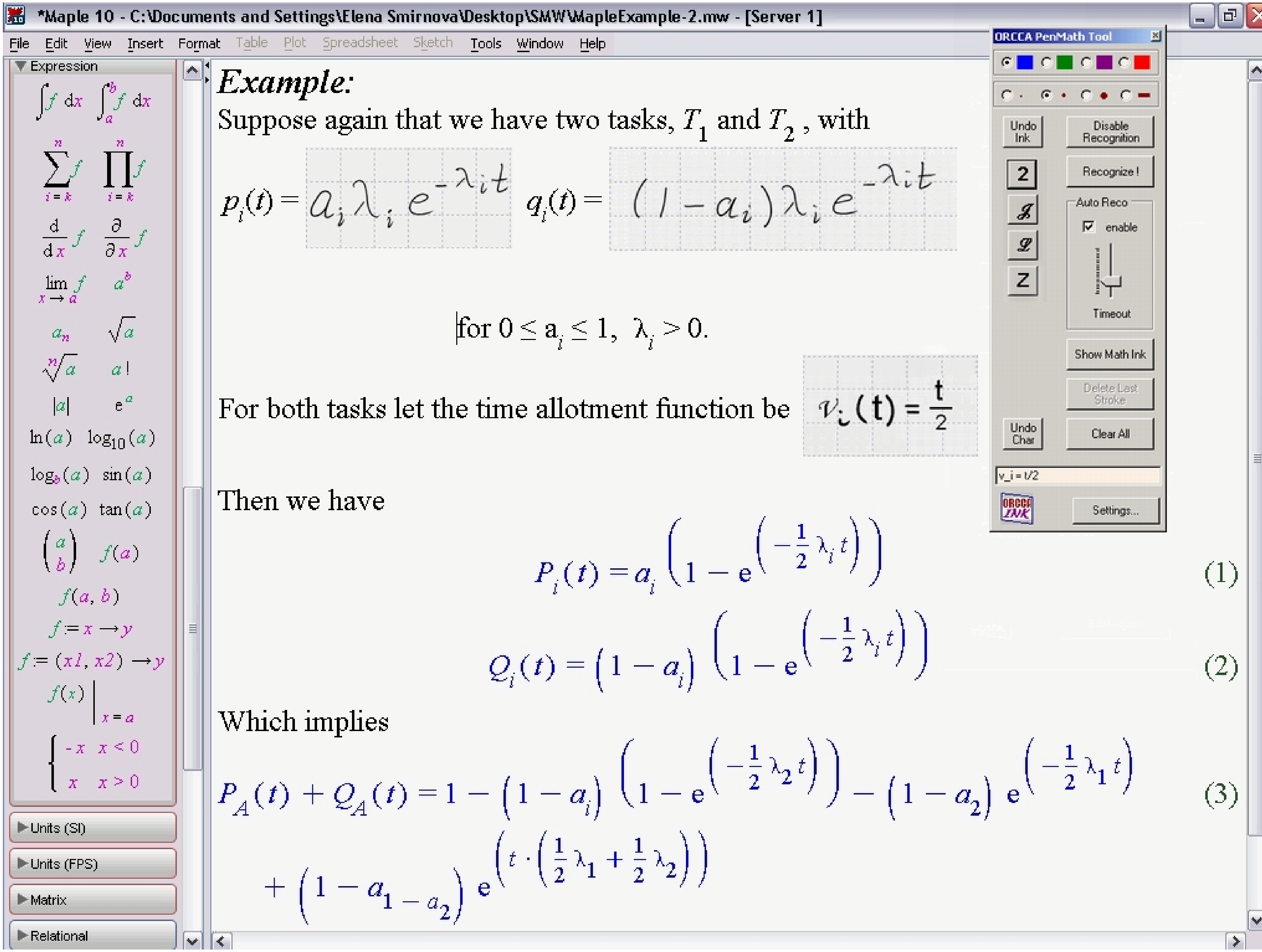
- interface to MS Word



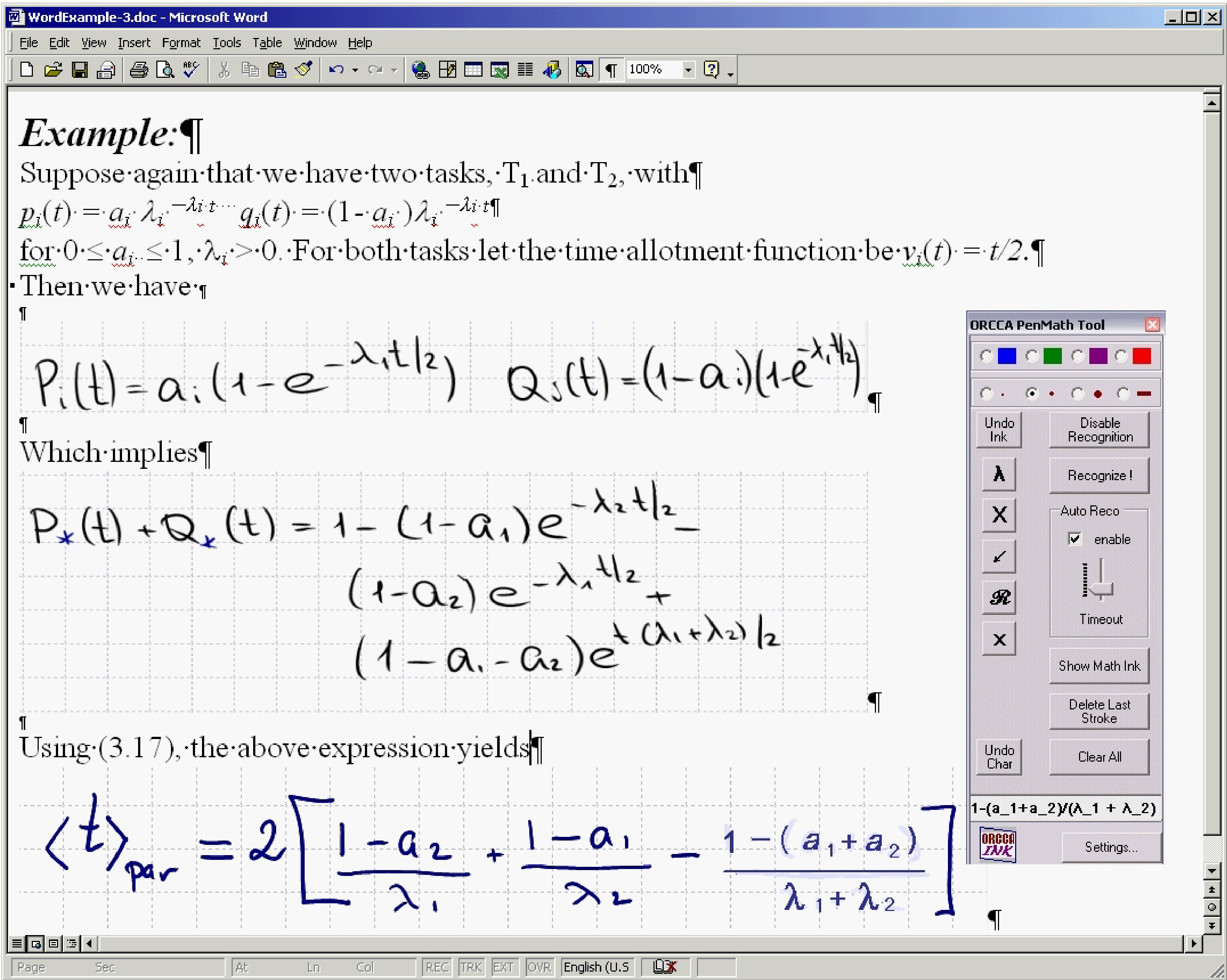
ActiveX control, accessing .NET control via Win 32 C++ Wrapper


MAPLE
10
MS WORD
W
3
Java
3
ActiveX
A
JAVA
JNI
a
a
2
b
JNI
B
C++
b
2
c
JNI +
JAWT NI
COM Interop
Win32
Wrapper
c
.NET
1
TABLET PC SDK

Pen-Math Interface In Maple 10



Pen-Math Interface In MS Word



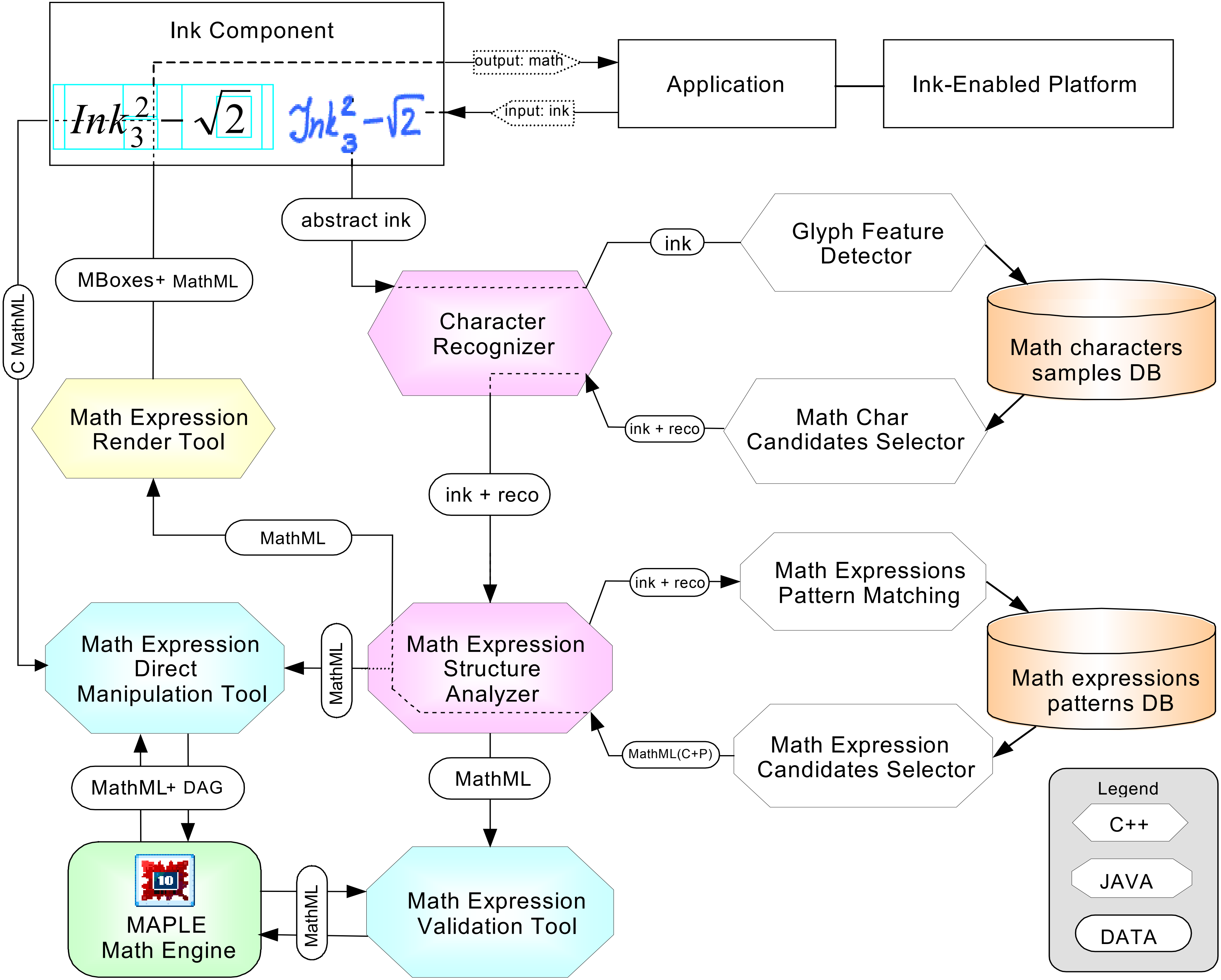
Progress Report

- **Current results**

  - We have developed a software solution to enable a pen-based math interface on Tablet PC platforms.
  - Successfully plugged a recognizer for a wide variety of math characters to the framework
  - We tested its compatibility with Maple 10, MS Office (2000, XP and 2003).

Future Work

- **Ongoing work in**
  - plugging tools to determine structures of math expression
  - enabling math engine features
    - to validate math expressions
    - to allow direct manipulation on math formulae
  - instantiating our solution on other platforms
    - for handheld devices
    - other operating systems.


Ink Component
output: math
input: ink
Application
Ink-Enabled Platform
abstract ink
MBoxes+ MathML
C MathML
Character Recognizer
ink
Glyph Feature Detector
Math characters samples DB
Math Char Candidates Selector
ink + reco
Math Expression Render Tool
ink + reco
MathML
Math Expressions Pattern Matching
ink + reco
Math expressions patterns DB
Math Expression Direct Manipulation Tool
MathML
Math Expression Structure Analyzer
MathML(C+P)
Math Expression Candidates Selector
MathML+ DAG
MathML
MAPLE Math Engine
MathML
Math Expression Validation Tool
Legend
C++
JAVA
DATA

Conclusions

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

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

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 believes** are
  - we have achieved these objectives
  - our solution can allow a more natural interface for mathematics in a variety of settings