

A Language for Building Web Interfaces to Mathematical Software

Rachel Sun

Supervisors:

Professor Wolfgang Schreiner and Professor Elena Kartashova

Department:

Research Institute For Symbolic Computation (RISC) & Institute For
Analysis

SUPPORTED BY FÖRDERUNG DER WISSENSCHAFTLICHEN
FORSCHUNG (FWF)

Outline

Motivation

Goals And Problem Definition

Literature Review And Background

Tools

Approach

Current Situation

Problems And Future Work

Preliminary Results And Timeline

A Quick Demo

Conclusion

Motivation

Why is a Language for Building Web Interfaces to
Mathematical Software facility needed?



I Will Explain... !

A language for Building Web Interfaces to
Mathematical Software

Motivation

Just Imagine as a **mathematician**:

You come up with an algorithmic solution to a problem

- ▶ Solution can be implemented in any language
- ▶ Solution can be written using existing mathematical software

And you want to publish it on the internet:

- ▶ Making it accessible to a broader audience (as a web application) without rewriting the solution

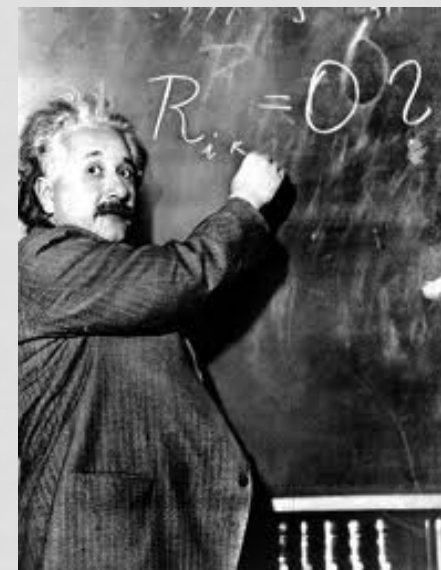
All you need is to tell a generic framework:

How you like it to be displayed?

How to perform the computation?

And a web application is automatically generated!

Won't it be nice?



Motivation

Just Imagine as a **student** or a **researcher**:

You do not have a copy of a particular mathematical software on the local system...

You go online and there you find a bunch of web-based applications

- ▶ that solve some kind of problem domains
- ▶ without having to install any software
- ▶ available anytime, anywhere



Won't it be nice?

A More Clearer Illustration

What You HAVE:

- ✓ Mathematical solution to a particular domain
- ✓ - written in any language
- ✓ - using existing software

What You DON'T HAVE:

- ✓ Specific knowledge how to write a web application

What You WANT:

- ✓ Publish it to the Internet
- ✓ Share your knowledge to a broader audience

What You DON'T WANT:

- ✓ To rewrite solution to adapt to web Technologies

Goals And Problem Definition

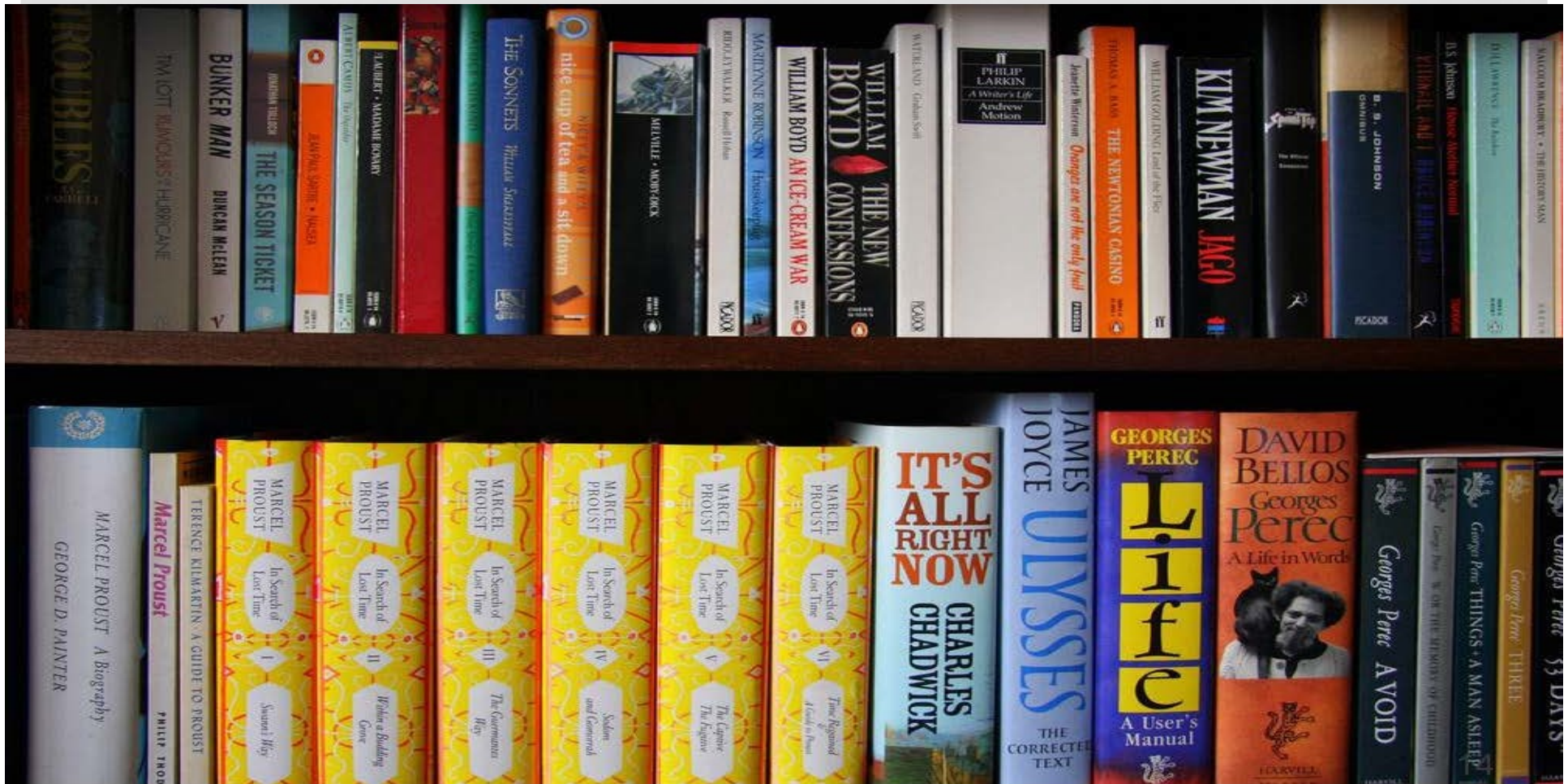


A language for Building Web Interfaces to
Mathematical Software

Goals And Problem Definition

- ▶ **Goal:** Design and implementation of a framework to create a dynamic web-based mathematical application that solves a class of mathematical problems.
- ▶ A generic framework which is independent of a particular mathematical domain problem
- ▶ All the mathematically computations can be performed in background by existing mathematical software
- ▶ Mathematical programmers needs to provide a workflow and an interface description, and necessary programs to the framework
- ▶ The actual web application will be generated from these information and the service is deployed on the server

Literature Review And Background



A language for Building Web Interfaces to
Mathematical Software

Current Approaches In Mathematical Web Application:

- ▶ Service Oriented Mathematical Systems
- ▶ Mathematics on the Web
- ▶ E-Learning Environments
- ▶ Virtual Laboratories
- ▶ Web Technologies

Service Oriented Mathematical Systems

- ▶ Aims to accelerate rapid application development by service reuse and discovery.

Example Applications:

▶ **Netlib**

- A general-purpose repository comprising a collection of free software libraries for mathematics computing via user interfaces.

▶ **NetSolve**

- A remote procedure call (RPC) based middleware system that integrates software resources distributed across a network.

▶ **Monet**

- A framework to discover mathematical web services dynamically based on published descriptions.

▶ **MathBroker**

- A software framework for brokering mathematical services.

▶ **MathWeb**

- Provides an infrastructure for linking mathematical services using a general distribution architecture.

A language for Building Web Interfaces to
Mathematical Software

Mathematics On The Web

Wolfram Research

Mathematica

- ▶ Interactive scientific computation software that offers the combination of numeric, algebraic, geometric and logic algorithms together with web services.



Wolfram webMathematica 3 More about webMathematica »

What is webMathematica?
webMathematica adds interactive calculations and visualization to websites by integrating Mathematica with the latest web server technology.

- Client technologies such as JavaScript and Java applets work seamlessly with webMathematica.
- Generate and display graphics and visualizations dynamically.
- Adjust your calculations interactively.
- Use familiar web interface elements such as buttons, drop-down lists, and text fields to control computations.
- webMathematica custom tags integrate strongly with the standard Java web technologies Java Servlet and Javasever Pages.
- Use sliders and other dynamic features to adjust parameters and generate new results interactively.
- Display output on the web page or create downloadable reports in PDF, notebook, or other formats.
- Use templates to give a consistent,

webMathematica

- ▶ A service that merges the computational power of Mathematica with the web.
- ▶ Intended to provide specialized computations over the web.

A language for Building Web Interfaces to
Mathematical Software

Mathematics On The Web

Wolfram Research



The screenshot shows the Wolfram Alpha website interface. At the top, the logo features a red starburst icon followed by the text "WolframAlpha" in a serif font, with "computational knowledge engine" in a smaller sans-serif font below it. Below the logo is a search bar with the placeholder text "Enter what you want to calculate or know about:". To the right of the search bar are two links: "Examples" and "Random". Below the search bar is a large, empty input field with a small orange square icon on the right side. Below the input field is a section with a light orange background. On the left, it says "Free online access to the Wolfram|Alpha computational knowledge engine:" followed by a paragraph: "Answer questions, do math, instantly get facts, create plots, calculators, unit conversions, scientific data and statistics, help with homework—and much more." To the right of this text is an illustration of several documents with various mathematical symbols and graphs. Below the illustration is a white button with the text "New to Wolfram Alpha?" and "TAKE THE TOUR »".

Wolfram Alpha

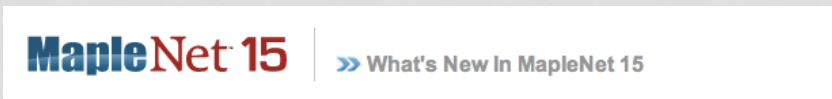
- ▶ An artificial intelligence based online answer engine with Mathematica running in the background.

Mathematics On The Web

Maplesoft

Maple

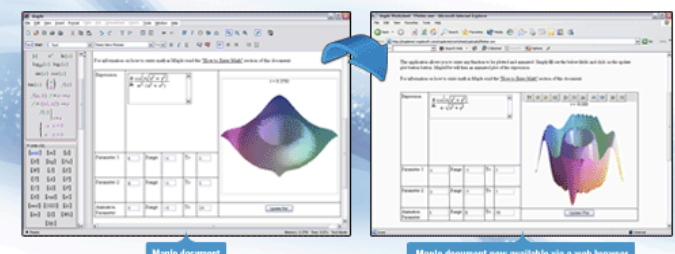
- ▶ Like Mathematica, Maple manipulates information in a symbolic or an algebraic manner.



MapleNet 15 | >> What's New In MapleNet 15

Bringing the power of Maple to your applications and web sites

The MapleNet suite of mathematical services brings the power of Maple to your applications and web sites. With MapleNet, you can add mathematical computations and visualizations to your web and desktop applications, share solutions over the web through interactive Maple documents, and develop rich technical web content.



Maple document | Maple document now available via a web browser



Maple 15

Overview | What's New? | Features | User Stories | Online Demos | How Does Maple Compare? | Us

Home : Maplesoft Products : Maple : Maple 15 for Professionals

Maple 15 is the essential technical computing software for today's technical

Math Engine | Smart Documents | Connections

Most Powerful Math Engine



Whether you need to do quick calculations, develop design sheets, or produce sophisticated high-fidelity simulation models, the world-leading Maple computation engine provides the necessary technology to dramatically increase your analytical productivity.

New in Maple 15!

- » [Speed up your Maple computations by taking advantage of all available processors](#)
- » [See more new features in Maple 15](#)

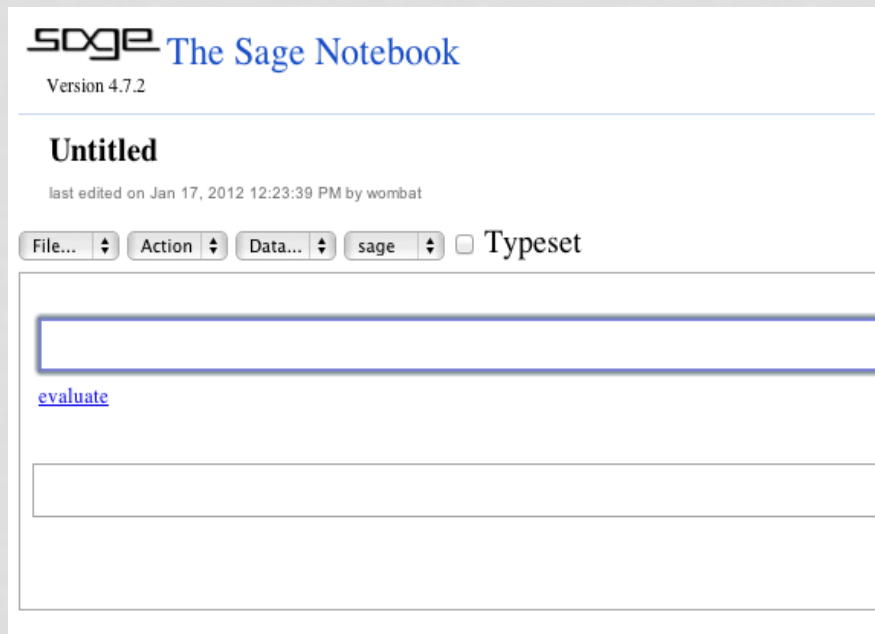
MapleNet

- ▶ With MapleNet Mathematical computations and visualizations are integrated into web and desktop applications.

A language for Building Web Interfaces to
Mathematical Software

Mathematics On The Web

Sage



- ▶ A free web-based Mathematical software system.
- ▶ Aims to create an open source alternative to Magma, Maple, Mathematica, and MATLAB.
- ▶ Uses Python programming language to support procedural, functional and object-oriented constructs.

E-Learning Environments



A language for Building Web Interfaces to
Mathematical Software

E-Learning Environments

- ▶ eFunda (engineering fundamentals):
 - ▶ Offers a convenient internet resource for online engineering computation.

The screenshot displays the eFunda website interface. At the top left is the eFunda logo. To its right is a navigation bar with links: Directory | Career | News | InfoStore | Industrial | SpecSearch® | Ask an Expert. Below this is a prominent 'GLOBALSPEC®' banner with the tagline 'The Engineering Search Engine®' and a search input field with a 'GO' button. A secondary search bar is located below the navigation menu. The main content area is organized into several columns of links:

- Materials:** Periodic Table, Steels, Aluminum, Polymers, Piezo, Corrosion, and more.
- Design Center:** Plastic Design, Gages, Springs, O-Rings, Sensors, Screws, I-Beams, and more.
- Processes:** Casting, Stamping, Molding, Annealing, Machining, Rapid Prototyping, and more.
- Unit Conversion:** Length, Mass, Pressure, Time, Base-N, Fractions, Hardness, Gage, and more.
- Formulas:** Beam, Plate, Composite, Vibration, Heat Transfer, Fluid Mechanics, and more.
- Mathematics:** Areas, Trigonometry, Taylor Series, Least Squares, Laplace Transform, and more.

At the bottom of the page, there are three featured sections:

- Renewable Energy:** Coverage of solar energy, solar architecture and sustainable buildings, wind energy (offshore and on shore), wave and tidal energy, biomass energy, renewable hydrogen and fuel cells.
- Wolfram webMathematica®:** A search bar with the expression $\int \text{Exp}[x] * \text{Sin}[x] dx$ and an 'INTEGRATE' button. Below it are links for Calculus (Integral, Derivative, Limit), Matrix (Inverse, Eigenvalue, Linear System), Plot (2D, 3D, Parametric), Numerical (Root, Curve Fit), and Transform (Laplace, Fourier).
- Mathematica Solutions:** A button labeled 'Click to Explore Mathematica Solutions »' with the text 'For Engineering, Medicine, Finance, Science, and More'.

A language for Building Web Interfaces to
Mathematical Software

E-Learning Environments

▶ ActiveMath:

- ▶ An artificial intelligence online collaborative learning environment for mathematics and other fields.



active π math

Content Software UdS Teaching Communities Guest - Login

ActiveMath Home

The ActiveMath group works at the frontiers of e-Learning. We enable learners to tap their full potential by keeping up their motivation and fostering a self-regulated learning process. We cooperate with teachers, tutors, psychologists and educators to improve learning and teaching in schools, universities and life-long learning.

Our research and development is focused on:

- Adaptive learning environments for mathematics and other fields
- Collaborative environments
- E-Portfolios and learning diaries

We develop and apply techniques from Artificial Intelligence such as:

- Semantic knowledge representations, user modeling, and planning
- Educational data mining
- Natural language processing

A language for Building Web Interfaces to
Mathematical Software

E-Learning Environments

▶ MERLOT:

- ▶ A free educational resource that comprises a collection of "learning objects" on the Web that refers users to useful Internet sites.

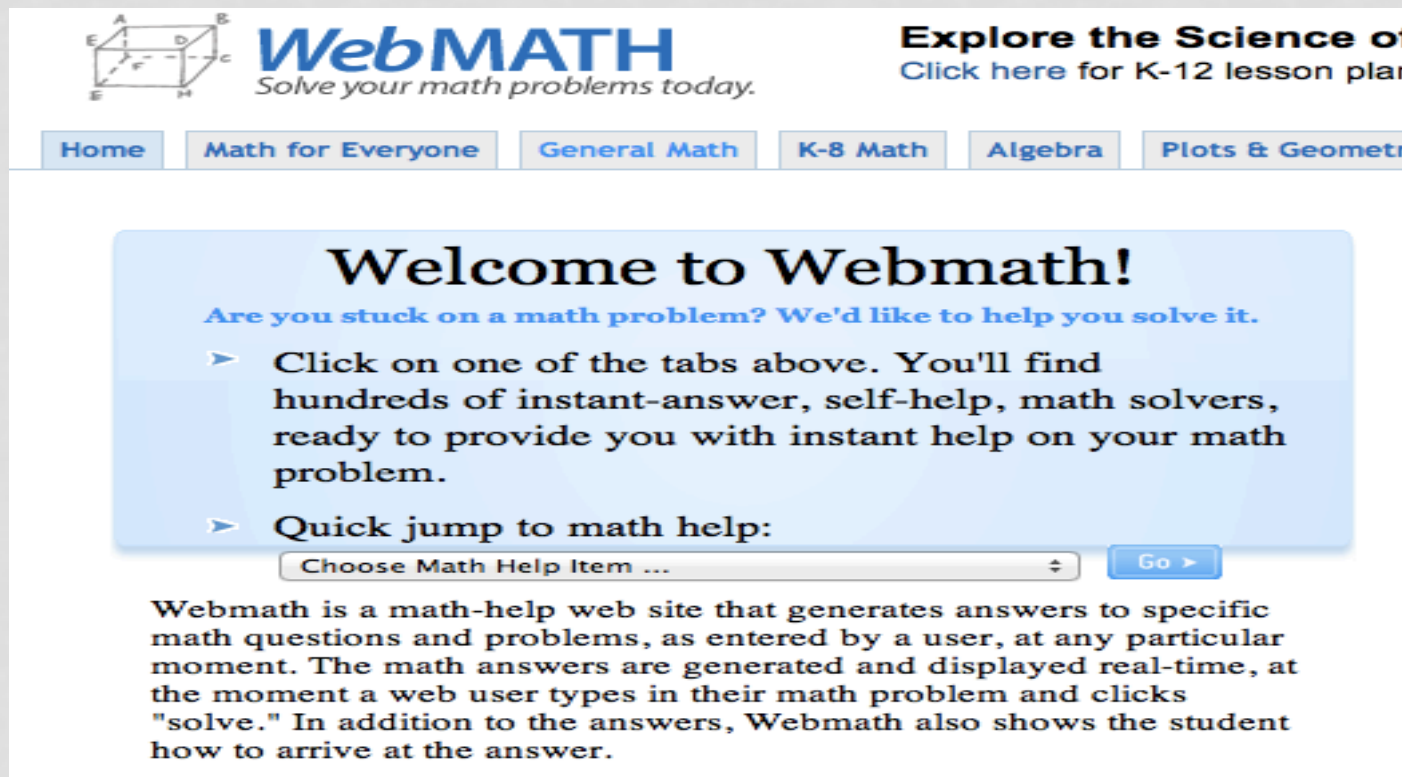
The screenshot shows the MERLOT website homepage. At the top left is the MERLOT logo with the tagline "Multimedia Educational Resource for Learning and Online Teaching". To the right is a search bar with the text "Search" and a button labeled "material". Below the search bar are navigation tabs: "Home", "Communities", "Learning Materials", "Member Directory", and "My Profile". A banner below the tabs reads "MERLOT on Twitter ** Become a MERLOT Peer Reviewer by Attending GRAPE Camp ** Listen to the Archived Webinar: Find...". The main content area is divided into several sections. On the left, there is a "Create Learning Materials with MERLOT Content Builder" box, a "Browse Collection" list with categories like Arts, Business, Education, Humanities, Mathematics and Statistics, Science and Technology, Social Sciences, and Workforce Development, and a "Browse Academic Support Services" box. The central area features a "Welcome to MERLOT" section with the tagline "Putting Educational Innovations Into Practice" and a description of the site's purpose. Below this is the "Exploring MERLOT" section, which includes links to "Learning Materials", "Personal Collections", "Learning Exercises", "Colleagues Across Disciplines", and "Guest Experts". At the bottom, there is a "Visit a Discipline Community" section with a list of disciplines including Agriculture and Environmental Sciences, Biology, Business, Chemistry, Communication Sciences and Disorders, Criminal Justice, Engineering, English, Faculty Development, Fire Safety, Health Sciences, History, Information Technology, Library and Information Services, Mathematics, Music, Physics, Psychology, Statistics, Teacher Education, Technical Allied Health, and World Languages. On the far left, a "News & Announcements" box displays "What's New in MERLOT" with statistics: "32,380 materials, 370 new materials, 101,110 members, 748 new members".

A language for Building Web Interfaces to
Mathematical Software

E-Learning Environments

▶ WebMath:

- ▶ A free online mathematical tutoring assistance site to help students with their mathematics proficiencies.



The screenshot shows the WebMATH website interface. At the top left is a 3D cube diagram with vertices labeled A, B, C, D, E, F, G, H. Next to it is the WebMATH logo with the tagline "Solve your math problems today." To the right, there is a link to "Explore the Science of" and a button for "Click here for K-12 lesson plan". Below the logo and tagline is a navigation bar with tabs for "Home", "Math for Everyone", "General Math", "K-8 Math", "Algebra", and "Plots & Geometr". The main content area features a large blue box with the heading "Welcome to Webmath!" and the text "Are you stuck on a math problem? We'd like to help you solve it." Below this is a list of instructions: "Click on one of the tabs above. You'll find hundreds of instant-answer, self-help, math solvers, ready to provide you with instant help on your math problem." and "Quick jump to math help:". Underneath the instructions is a search bar with the text "Choose Math Help Item ..." and a "Go" button. At the bottom of the page, there is a paragraph describing Webmath as a math-help web site that generates answers to specific math questions and problems, as entered by a user, at any particular moment. The math answers are generated and displayed real-time, at the moment a web user types in their math problem and clicks "solve." In addition to the answers, Webmath also shows the student how to arrive at the answer.

Virtual Laboratories

▶ Virtual Computing Lab (VCL)

(Peng Li, Lee W. Toderick, and Philip J. Lunsford, 2009)

- ▶ An open source remote learning environment.
- ▶ Goals is to provide remote access scheduling and image management system to the students at East Carolina University (ECU).

▶ Virtual Math Teams (VMT)

(Baba Kofi Weusijana, Jimmy Xiantong Ou, Gerry Stahl, and Stephen Weimar)

- ▶ Offers service for synchronous collaboration learning and group cognition within a rich integrated Internet-based environment.

▶ Distributed Control Lab (DCL)

(Andreas Rasche, Frank Feinbube, Peter Tröger, Bernhard Rabe, and Andreas Polze, 2005)

- ▶ Designed for teaching and research demonstration.
- ▶ Provides access to remote and real-time control experiments.

▶ The Virtual Networked Laboratory (VNL)

(Lee J. Leitner and John W. Cane, 2005)

- ▶ Replicates key aspects of traditional hands-on laboratory environment by providing a simulating remote laboratory experience.

▶ Virtual environment for learning networking (Velnet)

(Bruce Kneale, Ain Y De Horta, and Ilona Box, 2004)

- ▶ Enables laboratory experiments to be performed over network connections on a single desktop host computer.

Summary of The Current Approaches

- ▶ Need specific knowledge for the service oriented programming and web application.
- ▶ Cannot use arbitrary programming languages for the solution, such as in Sage, Mathematica, Maple etc.
- ▶ Specific software or hardware requirements on the client like in some of the virtual laboratories solutions.
- ▶ Tailored for one specific problem domain like in the virtual laboratories solutions.

Web Technologies

Web Technologies	Description
HTML	<ul style="list-style-type: none">• Used to describe WWW content.• Not so strict concerning the structure of an HTML document.
XHTML	<ul style="list-style-type: none">• Define HTML as an XML.• Documents must be marked up correctly and 'well-formed'.• Gives 'semantic' to a given tag.• Separates the presentation from structure.
XML	<ul style="list-style-type: none">• A universal data representation markup language.• Aimed to encode semantics into web documents.• Facilitates information interchange and integration from heterogeneous systems
XML Schema (XSD)	<ul style="list-style-type: none">• Define the possible structure and contents of an XML format.• A validating parser can check whether an XML instance document conforms to an XSD schema or a set of schemas• Check the correctness of node hierarchy and data-types in the XML

Web Technologies

Web Technologies	Description
PHP	<ul style="list-style-type: none">• Language for creating interactive web pages.• Can be embedded into HTML.• Deployed on most web servers.• Executes scripts directly on the web server.• Supports many databases, such as MySQL, Oracle, Sybase, Solid and etc.
JavaScript	<ul style="list-style-type: none">• Client-side object-oriented scripting language for web pages.• Designed to add functionalities and interactivity to HTML Pages.• Embedded directly into HTML pages.• Script is run by the web browser.• Used to validate forms and communicate with the server.

Web Technologies

Web Technologies	Description
Ajax	<ul style="list-style-type: none">• Used primarily for developing highly interactive web applications.• Offers a rich desktop-like user experience.• Allows asynchronous access of information to the server.
GWT	<ul style="list-style-type: none">• Makes it easier to develop interactive, AJAX web applications.• Java is the only language used in the web development program.• Compiles the Java source code directly into HTML and JavaScript code.• Run in all major web browsers.• No single line of HTML or JavaScript code is necessary.
Pyjamas	<ul style="list-style-type: none">• Is a port of GWT.• Compiles the Python source code directly into HTML and JavaScript code.

Tools



A language for Building Web Interfaces to
Mathematical Software

Tools

- ✓ Eclipse Indigo 3.7
- ✓ Pydev
- ✓ Python 2.7
- ✓ XML Schema
- ✓ XML
- ✓ lxml Toolkit
- ✓ Pyjamas 0.7
- ✓ Apache Web Server 2.2.20
- ✓ JSON-RPC 2.0

Tools

Why use `Pyjamas` over `GWTT`?

- ▶ We do not want Mathematicians to provide files (handler functions) written in Java.
 - Python is a lightweight and an intuitive language.
- ▶ To have Python script ability in `GWTT`, a Python interpreter is needed to be implemented in JavaScript to run in the web browser.
- ▶ Here comes the `Pyjamas` - a port of `GWTT`:
 - If there is already a compiler which translates python into JavaScript, why do we need an extra step?
 - It will hence as well translate the Python handler functions to JavaScript.
 - No extra interpreter needed.

Tools and Programming Knowledge For Mathematician

What you ONLY NEED :

- ✓ Basic knowledge in XML and Python
- ✓ Any text editor (for programming and XML editing)

What You DON'T NEED:

- ✓ Eclipse Indigo 3.7
- ✓ XML Schema
- ✓ lxml Toolkit
- ✓ Pyjamas 0.7
- ✓ Apache Web Server
- ✓ JSON-RPC 2.0

Approach



A language for Building Web Interfaces to
Mathematical Software

Files To Write And Provide

Two XML Files

- ▶ GUI Definitions
- ▶ Mathematical Server System-Calls Definitions

Two Python Scripts

- ▶ Client-side Handler
- ▶ Server-side Handler

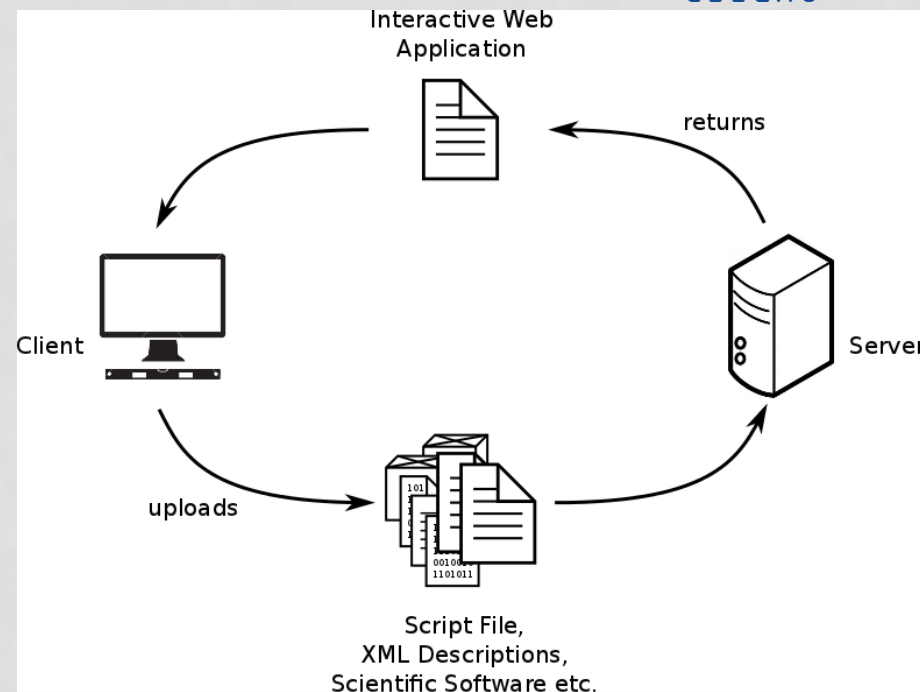
Mathematical Computation Files

- ▶ Arbitrary scripts, libraries, programs etc.
- ▶ Used to perform actual mathematical computation

A Simple System Workflow

4. Web users will now be able to use the software for performing computation over a web-browser.

3. It then either returns a link pointing to the web application or an error message to the client.

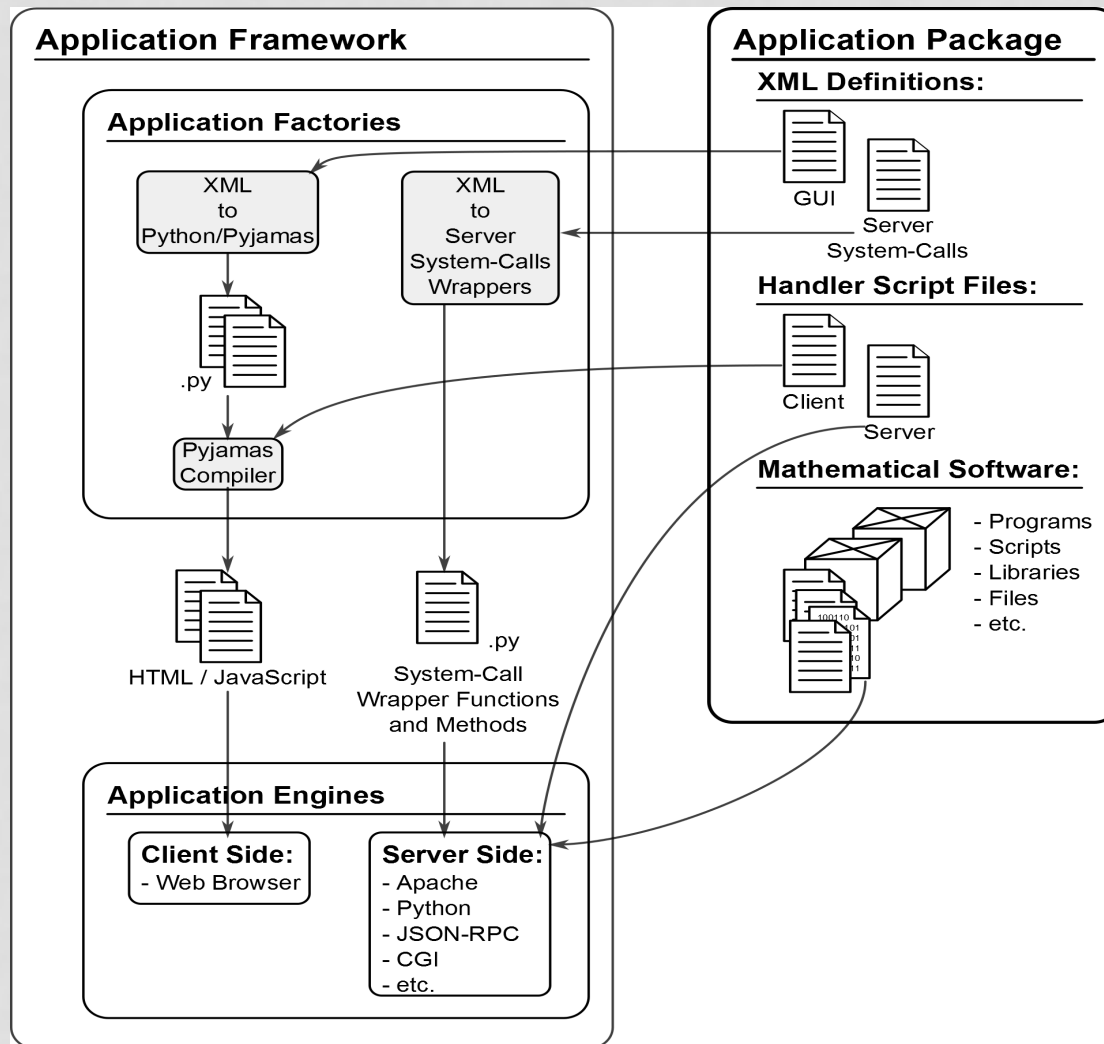


2. The server generates based on the uploaded files the interactive web application.

1. Client on his host computer uploads the necessary files for the framework to the server.

A language for Building Web Interfaces to
Mathematical Software

A More Insight Workflow



A language for Building Web Interfaces to
Mathematical Software

XML File:

GUI Definitions

- ▶ Define how the static web interface should look like.
- ▶ Define what methods are available in a remote service (JSON-RPC) and where to find them.
- ▶ Widgets that are currently supported:
 - ▶ Panel: Horizontal, Vertical
 - ▶ Widget: Button, CheckBox, RadioButton, ListBox, TextBox, TextArea, Label, HTML and Image.

XML File:

GUI Definitions Example

```
<GUI name= "MyFirstWebApplication" >
  <ServerCallBack name= "server" url=http://localhost/toyservice/cgi-bin/calculator.py
    methods= "add, sub" >
</ServerCallBack>
  <VerticalPanel>
    <HorizontalPanel>
      <Label text="Input 1 : "/>
      <TextBox name="input0"/>
      <Label text="Input 2 : "/>
      <TextBox name="input1"/>
    </HorizontalPanel>
    <HorizontalPanel >
      <RadioButton name="rb0" label="Add" group="group1" checked="True" />
      <RadioButton name="rb1" label="Sub" group="group1" checked="False"/>
      <Button label="Calculate" listener="calc_callback"/>
    </HorizontalPanel>
    <HorizontalPanel>
      <Label text="Output : "/>
      <TextBox name="output0" />
      <TextBox name="output1" />
    </HorizontalPanel>
  </VerticalPanel>
</GUI>
```

JSON Proxy object for handling JSON-RPC asynchronous communication to the server.

There can be multiple JSON Proxy objects.

Location of the Service.

Layouts of GUI.

Widget with a "name" attribute can be accessed from within the client handler function.

Name of the client handler function. The event is fired when the corresponding button is clicked.

XML File:

GUI Definitions Example

Generated Web Application Graphical User Interface (GUI):



JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

Input 1 : Input 2 :

Add Sub Mul Div

Output :

foo ▾

Web user will not be blocked waiting for the result during the asynchronous JSON-RPC communication.

Python Script File: Client-Side Handler

- ▶ Handles all the events fired in the webpage.
- ▶ Use the name of the widget defined in the XML GUI file to get and set values.
- ▶ All the method calls upon a widget are wrapped inside a wrapper class.
- ▶ If there is any changes in the Pyjamas API in the future:
 - ▶ Only the wrapper classes need to adjust to the changes.
 - ▶ No changes on the method calls in the client handler file - method names stay the same.

Python Script File: Client-Side Handler Example

```
def event_Handler( gui ):
    a = int( gui.input0.text )
    b = int( gui.input1.text )
```

← Object for manipulating the GUI.
← Gets the values of text boxes.

Client-side Computation

```
    if gui.rb0.checked:
        gui.output0.text = a + b
```

← Ordinary value assignment:
Translated directly by Pyjamas
Compiler.

Server-side Computation

```
    elif gui.rb1.checked:
        gui.output1.text = gui.server.sub( a, b )
```

← Triggered by RPC: Call
the server's method
with appropriate
arguments.

← Set the response from the server to
the desired widget on arrival.
(asynchronous communication)

XML Files:

Mathematical Server System-Calls

Definition

- ▶ Define how to call the mathematical software to perform actual computation.
- ▶ Wraps mathematical system calls into python functions:
 - ▶ Can be reused in the server-side script/handler file that defines services.
- ▶ Work still in progress. ..

XML Files:

Mathematical Server System-Calls

Definition Suggestion

(not implemented yet)

"ComputeAdd()" will be the actual function name in the python server-side handler script.

a and b are the arguments of the function and c is the return value.

```
<Function name="c = ComputeAdd(a,b)">
  <Exec>
    <Prog>./bin/add $a $b</Prog>
    <Pipe from="stdout" to="c"><Pipe>
  </Exec>
</Function>
```

"Prog" defines

- which program (mathematical software) is used for the computation
- location of the program.
- how to place the arguments.

"Pipe" redirects the output of the program from stdout to the return value c.

The above XML code should be translated into something like this in Python:

```
def ComputeAdd(a,b):
    c = os.popen('/bin/add %s %s' % (a,b)).read()
    return c
```

Python Script File: Server-Side Handler

- ▶ Defines how to handle service requests on the server.
- ▶ Gives the mathematician all the possibilities of Python (conditions, loops etc.):
 - ▶ Provides Mathematicians with the freedom and flexibility in defining how services should be handled.
 - ▶ When to call mathematical software on the server, how many times, under which conditions etc.

Python Script File: Server-Side Handler Example

```
class CalculatorService(object):
```

```
    @ServiceMethod
```

```
    def add(self, a, b):
```

```
        return ComputeAdd (a, b)
```

```
    @ServiceMethod
```

```
    def sub(self, a, b):
```

```
        return a - b
```

```
    @ServiceMethod
```

```
    def mul(self, a, b):
```

```
        return a * b
```

```
    @ServiceMethod
```

```
    def div(self, a, b):
```

```
        return a / b
```

Methods with `@ServiceMethod` decorator are exposed as services to callers.

Calling the actual mathematical software in the background using the python wrapper function.

Perform computation directly inside the method.

Current Situation

- ▶ **Client-side** Communication:
 - ▶ XML GUI definition
 - ▶ Designed the XML GUI definition.
 - ▶ Half-way through designing and defining XSD for validation against XML GUI definitions.
 - ▶ Python client-side handler file
 - ▶ Redesigned to become more understandable.
 - ▶ Implemented wrapper classes to become independent of Pyjamas API.
 - ▶ Implemented the XML ---> Pyjamas converter in Python.
- ▶ **Server-side** Communication:
 - ▶ Python server-side handler file
 - ▶ Designed and implemented the prototype.

Problems and Future Work

Problems:

- ▶ How to display graph and other data (any multimedia content) with Pyjamas over JSON-RPC.
- ▶ How should the XML Mathematical Server System-Calls Definition look like.

Future Work:

- ▶ Convert XML Mathematical Server System-Calls Definition into Python function.
- ▶ Implement additional widgets and panels that might be needed.
- ▶ Software evaluation:
 - ▶ Perform tests with different kind of combination of widgets and different backend software.
 - ▶
- ▶ Server installation.

Preliminary Results & Timeline

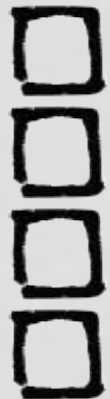


Literature Reviews

Tools Selection

Architecture Sketch

Prototype Development



Finish Implementation

Service Testing

System installation on JKU server

Completion of Thesis Writing

A Quick Demo

A language for Building Web Interfaces to
Mathematical Software

Conclusion

- ▶ There is a need for facilities in mathematical web-based applications.
- ▶ No simple general purpose framework so far that fulfills this kind of requirements.
- ▶ Beneficial for mathematical community and educators due to its simplicity.

Thank you!



A language for Building Web Interfaces to
Mathematical Software