OpenMath in SCiEnce: Evolving of Symbolic Computation Interaction

Sebastian Freundt
Peter Horn
Alexander Konovalov
Sylla Lesseni
Steve Linton
Dan Roozemond

22nd OpenMath Workshop, Grand Bend, July 9th, 2009.

www.symbolic-computation.org
European “Framework 6” programme,

Started April 2006, runs for 5 years,

Main purpose:

“to unite the European community of researchers in, and users of, symbolic computation. SCIEnce aims to promote the development of new software that is

- made more efficient by sharing components and expertise;
- made more interoperable in the modern Web services environment; and
- ready for the coming environment of Grid computing.”
• The Centre for Interdisciplinary Research in Computational Algebra,
  University of St Andrews, Scotland
• Research Institute for Symbolic Computation,
  Linz, Austria
• Ecole Polytechnique,
  Centre National de la Recherche Scientifique, Paris, France
• Computational Mathematicatics Group,
  Universität Kassel, Germany
• The KANT group,
  Technische Universität Berlin, Germany
• Discrete Algebra and Geometry group,
  Technische Universiteit Eindhoven, Netherlands
• Institute e-Austria Timisoara,
  Romania
• Maplesoft,
  Waterloo, Canada
• The Dependable Systems Research Group,
  Heriot-Watt University, Edinburgh, Scotland
What’s it all about?
What’s it all about?

Directly linking
Symbolic
Software

CAS1

CAS2

CAS3
What’s it all about?

Directly linking Symbolic Software

Link Symbolic Software to other Systems

- **SOAP-Clients**
  - Web-Apps, Java, C#, other CAS, ...

- **GET-Clients**
  - Scripts, other simple applications, ...

- **Humans**
  - Administration

SOAP

http GET

http html

Middleware

- **GAP**
- **KANT**
- **MuPAD**
- **Maple**

more to come ...

GAP 4

CAS1

CAS2  CAS3

Thursday, July 9, 2009
What’s it all about?

Symbolic Computation Software Composability Protocol

CAS1

CAS2

CAS3

GAP

KANT

MuPAD

Maple

more to come ...

Thursday, July 9, 2009
Protocol for communication between CASes

- OpenMath based
- Lightweight, simple sockets
- Basis for symbolic computation on Clusters and Grids
- Described in the “SCSCP standard (version 1.3)”, and scscp1 and scscp2 Content Dictionaries
- More details in 20 minutes.
Possible applications
Possible applications

- Cross-program: Software A can do things B can’t,
Possible applications

- Cross-program: Software A can do things B can’t,
- Cross-program: A can do things much faster than B,
Possible applications

- Cross-program: Software A can do things B can’t,
- Cross-program: A can do things much faster than B,
- Cross-platform: A is only available on *nix,
Possible applications

- Cross-program: Software A can do things B can’t,
- Cross-program: A can do things much faster than B,
- Cross-platform: A is only available on *nix,
- A is a pain to compile and install,
Possible applications

- Cross-program: Software A can do things B can’t,
- Cross-program: A can do things much faster than B,
- Cross-platform: A is only available on *nix,
- A is a pain to compile and install,
- etc...
OpenMath allows for different representations:

- XML
- OpenMath Binary
OpenMath allows for different representations:

- XML
- OpenMath Binary
OpenMath allows for different representations:

- XML

- OpenMath Binary
- OpenMath allows for different representations:
  - XML
  - OpenMath Binary

<table>
<thead>
<tr>
<th></th>
<th>XML Representation</th>
<th>Binary Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+2</td>
<td>`&lt;OMOBJ&gt;&lt;OMA&gt;&lt;OMS cd=&quot;arith1&quot; name=&quot;plus&quot;/&gt;</td>
<td>18 10 08 06 04 61</td>
</tr>
<tr>
<td></td>
<td><code>&lt;OMI&gt;1&lt;/OMI&gt;&lt;OMI&gt;2&lt;/OMI&gt;&lt;/OMA&gt;&lt;/OMOBJ&gt;</code></td>
<td>72 69 74 68 31 70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6c 75 73 01 01 01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>02 11 19</td>
</tr>
</tbody>
</table>

|  x  | `<OMOBJ><OMBIND><OMS cd="fns1" name="lambda"/>     | 18 1a 08 04 06 66    |
|     | `<OMBVAR><OMV name="x"/></OMBVAR></OMA><OMS cd="arith1" name="times"/> | 6e 73 31 6c 61 6d    |
|     | `<OMV name="x"/> `<OMS cd="nums2" name="pi"/> </OMA>` | 62 64 61 1c 05 01    |
|     | `<OMBIND></OMOBJ>`                                   | 78 1d 10 08 06 05    |
|     |                                                        | 61 72 69 74 68 31    |
|     |                                                        | 74 69 6d 65 73 05    |
|     |                                                        | 01 78 08 05 02 6e    |
|     |                                                        | 75 6d 73 31 70 69    |
| x→x·π | `<OMOBJ><OMBIND><OMS cd="fns1" name="lambda"/>     | 11 1b 19            |
|     | `<OMBVAR><OMV name="x"/></OMBVAR></OMA><OMS cd="arith1" name="times"/> |                          |
|     | `<OMV name="x"/> `<OMS cd="nums2" name="pi"/> </OMA>` |                          |
|     | `<OMBIND></OMOBJ>`                                   |                          |
OpenMath allows for different representations:

- XML
- OpenMath Binary

<table>
<thead>
<tr>
<th>Expression</th>
<th>OpenMath XML</th>
<th>OpenMath Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+2</td>
<td><code>&lt;OMOBJ&gt;&lt;OMA&gt;&lt;OMS cd=&quot;arith1&quot; name=&quot;plus&quot;/&gt; &lt;OMI&gt;1&lt;/OMI&gt;&lt;OMI&gt;2&lt;/OMI&gt;&lt;/OMA&gt;&lt;/OMOBJ&gt;</code></td>
<td>18 10 08 06 04 61 72 69 74 68 31 70 6c 75 73 01 01 01 02 11 19</td>
</tr>
<tr>
<td>$x \rightarrow x \cdot \pi$</td>
<td><code>&lt;OMOBJ&gt;&lt;OMBIND&gt;&lt;OMS cd=&quot;fns1&quot; name=&quot;lambda&quot;/&gt; &lt;OMBVAR&gt;&lt;OMV name=&quot;x&quot;/&gt;&lt;/OMBVAR&gt;&lt;OMA&gt;&lt;OMS cd=&quot;arith1&quot; name=&quot;times&quot;/&gt; &lt;OMV name=&quot;x&quot;/&gt; &lt;OMS cd=&quot;nums2&quot; name=&quot;pi&quot;/&gt;&lt;/OMA&gt;&lt;/OMBIND&gt;&lt;/OMOBJ&gt;</code></td>
<td>18 1a 08 04 06 66 6e 73 31 6c 61 6d 62 64 61 1c 05 01 78 1d 10 08 06 05 61 72 69 74 68 31 74 69 6d 65 73 05 01 78 08 05 02 6e 75 6d 73 31 70 69 11 1b 19</td>
</tr>
</tbody>
</table>

Bulky
OpenMath allows for different representations:

- **XML**
- **OpenMath Binary**

### Table

<table>
<thead>
<tr>
<th>Expression</th>
<th>XML Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1+2$</td>
<td><code>&lt;OMOBJ&gt;&lt;OMA&gt;&lt;OMS cd=&quot;arith1&quot; name=&quot;plus&quot;/&gt;&lt;OMI&gt;1&lt;/OMI&gt;&lt;OMI&gt;2&lt;/OMI&gt;&lt;/OMA&gt;&lt;/OMOBJ&gt;</code></td>
</tr>
<tr>
<td>$x \rightarrow x \cdot \pi$</td>
<td><code>&lt;OMOBJ&gt;&lt;OMBIND&gt;&lt;OMS cd=&quot;fns1&quot; name=&quot;lambda&quot;/&gt;&lt;OMBVAR&gt;&lt;OMV name=&quot;x&quot;/&gt;&lt;/OMBVAR&gt;&lt;OMA&gt;&lt;OMS cd=&quot;arith1&quot; name=&quot;times&quot;/&gt;&lt;OMV name=&quot;x&quot;/&gt;&lt;OMS cd=&quot;nums2&quot; name=&quot;pi&quot;/&gt;&lt;/OMA&gt;&lt;/OMBIND&gt;&lt;/OMOBJ&gt;</code></td>
</tr>
</tbody>
</table>
POPCORN offers an OpenMath-representation for humans:

- Integers, Floats and Strings as you expect:
  18, 0.6, 2.09e3, "22nd OpenMath Workshop"

- Symbols: cdname.symbolname

- Variables: $name, References: #name

- Application: arith1.plus(1,2,3)

- Binding: fns1.lambda[$x -> $x + 1]

- Attribution: some.thing{aa.bb -> 1}

- Some abbreviations and infix operators
Possibly Only Practical Convenient OpenMath Replacement Notation
Possibly Only Practical Convenient OpenMath Replacement Notation

1 + 2

1+2

Thursday, July 9, 2009
<table>
<thead>
<tr>
<th>1 + 2</th>
<th>1 + 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x \rightarrow x \cdot \pi)</td>
<td>(\lambda[x \rightarrow x \cdot \pi])</td>
</tr>
<tr>
<td>Expression</td>
<td>Equivalent</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>$1 + 2$</td>
<td>$1+2$</td>
</tr>
<tr>
<td>$x \rightarrow x \cdot \pi$</td>
<td>lambda[$x$ -&gt; $x \cdot \pi$]</td>
</tr>
<tr>
<td>$17.6 \cdot e$</td>
<td>$17.6\cdot e$</td>
</tr>
<tr>
<td>Expression</td>
<td>Notation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>$1 + 2$</td>
<td>1+2</td>
</tr>
<tr>
<td>$x \rightarrow x \cdot \pi$</td>
<td>lambda[$x \rightarrow x*\pi$]</td>
</tr>
<tr>
<td>$17.6 \cdot e$</td>
<td>17.6*e</td>
</tr>
<tr>
<td>$\frac{22}{7}$</td>
<td>22//7{somecd.approx -&gt; 3.14}</td>
</tr>
<tr>
<td>Expression</td>
<td>Equivalent expression</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$1 + 2$</td>
<td>1+2</td>
</tr>
<tr>
<td>$x \rightarrow x \cdot \pi$</td>
<td>lambda[$x \rightarrow x \cdot \pi$]</td>
</tr>
<tr>
<td>$17.6 \cdot e$</td>
<td>17.6*e</td>
</tr>
<tr>
<td>$\frac{22}{7}$</td>
<td>22//7{somecd.approx -&gt; 3.14}</td>
</tr>
<tr>
<td>$\int_{0}^{1} \frac{1}{x^3 + \sin x} , dx$</td>
<td>calculus1.defint(interval1.interval(0,1), lambda[$x \rightarrow 1/(x^3 + \sin(x))]$)</td>
</tr>
</tbody>
</table>
Java Libraries for OpenMath and SCSCP
Java Libraries for OpenMath and SCSCP

org.symcomp.openmath
Java Libraries
for OpenMath and SCSCP

org.symcomp.openmath

org.symcomp.scscp

Thursday, July 9, 2009
Java Libraries for OpenMath and SCSCP

org.symcomp.openmath

org.symcomp.scscp

Standard Libraries
Java Libraries
for OpenMath and SCSCP

SCSCP enabled Java-Application

Standard Libraries

org.symcomp.openmath

org.symcomp.scscp
Java Libraries for OpenMath and SCSCP

org.symcomp.openmath

org.symcomp.scscp

Thursday, July 9, 2009
Java Libraries for OpenMath and SCSCP

- Representation and Manipulation of OM
- Many convenience methods
- Reads and writes different formats
- Extensible
Java Libraries
for OpenMath and SCSCP

- Representation and Manipulation of OM
- Many convenience methods
- Reads and writes different formats
- Extensible

- Wraps all SCSCP functionality
- Turning a Java-application into a SCSCP server/client is a one-liner
- Comes with many examples
What else?

- Grid and Cluster Infrastructure
- WUPSI, the Wonderful Universal Popcorn SCSCP Interface: a great testing and demo tool
- Webproxy, a Web based Administration and orchestration tool offering SOAP access
Demo....

Thursday, July 9, 2009
State of the systems

- GAP: Client and (single) server
- KANT: Client and server
- MuPAD: Client and server (using Java libs)
- Maple: (First prototype of) client and server
- Magma: Server (using Java libs)
- TRIP (celestial mechanics): Server and client (using their own, public, SCSCP C-library)
- .... more to come
Licensing and availability

- GAP: Free and open source; SCSCP/OpenMath libraries now included with GAP distribution
- KANT: Free; binaries from KANT homepage
- TRIP SCSCP C-library: open source, free
- Java libraries: Free, open source
- MuPAD OpenMath/SCSCP: Free;
- Magma OpenMath/SCSCP: Pending documentation...
- .... more to come
Thank you!

SCIENCE homepage
http://www.symbolic-computation.org/

The java libraries are available at
http://java.symcomp.org/