

MONET: The Semantic Web for Mathematical Web Services

Walter Barbera Medina, Marc Aird and Julian Padget

Department of Computer Science
University of Bath

E-mail: {wbm, mapma, jap}@cs.bath.ac.uk

WWW: <http://monet.nag.co.uk>

Breakdown of the talk

- Maple: Technical details
 - Phrasebook
 - Communication with Maple.
- Invocation of Maple web services.
- Deployment of Maple web services.
 - Generation of Service Description (MSDL).
- Broker
 - Registration/Deregistration of services.
- Latest changes.
- Some Demos.

Maple: *Phrasebook*

- OpenMath: standard for representing the semantics of mathematical objects.
- Java Codec for encoding OpenMath as Maple

```
import nl.tue.win.riaca.openmath.codec.*;

public String encodeOMObject(OMObject fOMObject) {

    // Returns the integer if OMObject is an OMInteger
    if (fOMObject instanceof OMInteger) {
        OMInteger tOMInteger = (OMInteger) fOMObject;
        return tOMInteger.getInteger();
    }
    ...
}
```

Maple: *Communication*

- Over sockets (discarded).
 - Maple worksheet runs a procedure and opens a socket into Maple

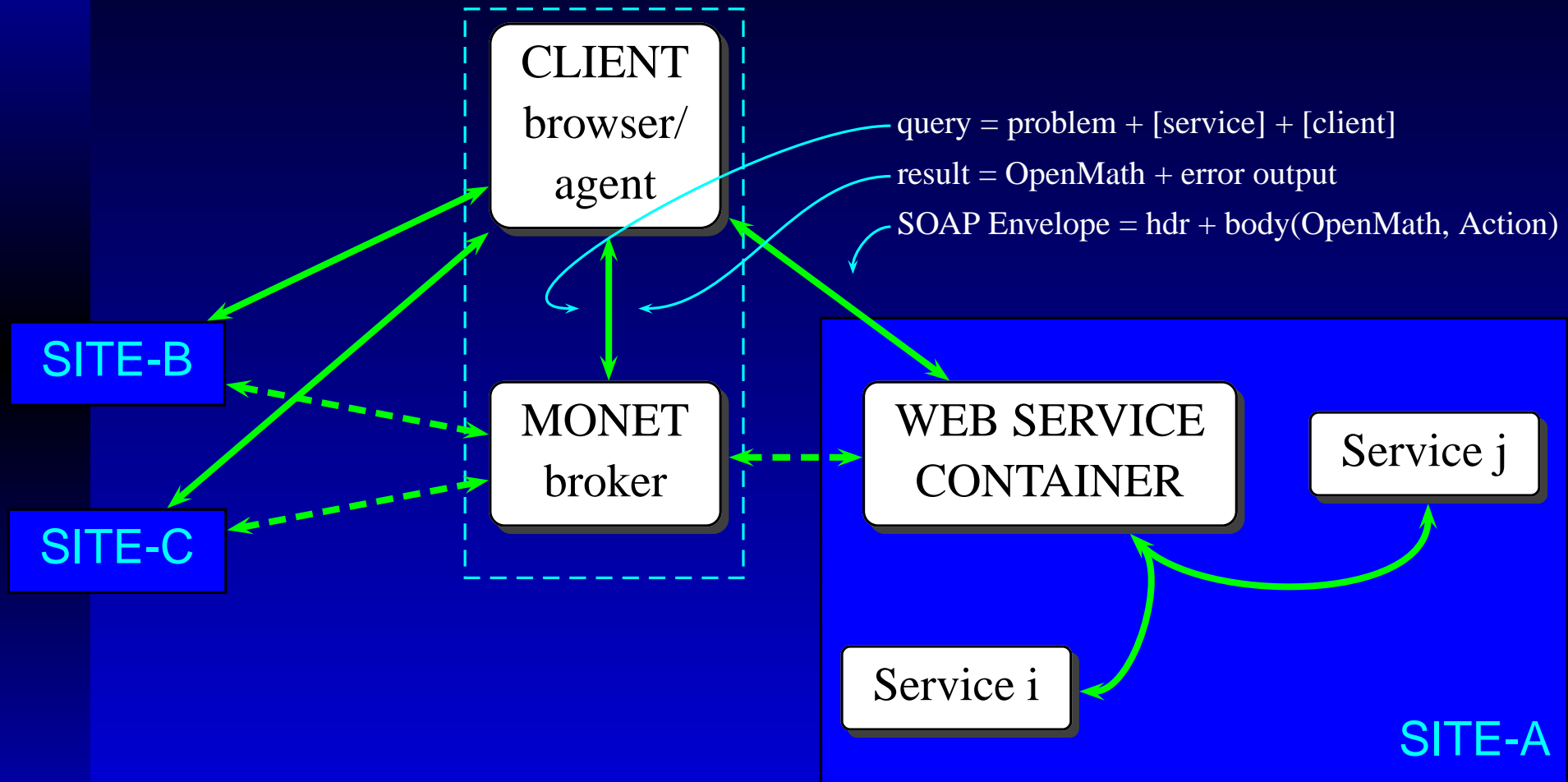
```
server := proc(sid)
  local a, b, c;
  use Sockets in
    a := Read(sid, 20):
    b := parse(a, statement):
    c := convert(b, string):
    Write(sid, sprintf("%s",))
  end use
end proc:
Sockets:-Serve(9999, server);
```

- Sockets opened permanently (not session based).

Maple: *Communication*

- Over JavaMath (discarded).
 - Objective: Enable mathematical programs in Java to use computational capabilities of existing compute engines such as Maple or Mathematica.
 - Use of general purpose computer algebra package.
 - Cumbersome installation.
- Over Runtime java class (command line).
 - Stateless Client-Server interaction.
 - The execution takes place on the server side.

MONET: Architecture



Maple Invocation Web Service

- Creation of generic Maple code which enables loading specific maple problems code.
- Generation of an invocation web service using TOMCAT and AXIS.
 - Communication Client - Server through SOAP messages.
- WSDL:
 - Generation of a common WSDL.
 - Generation of one WSDL for each Maple web service created.

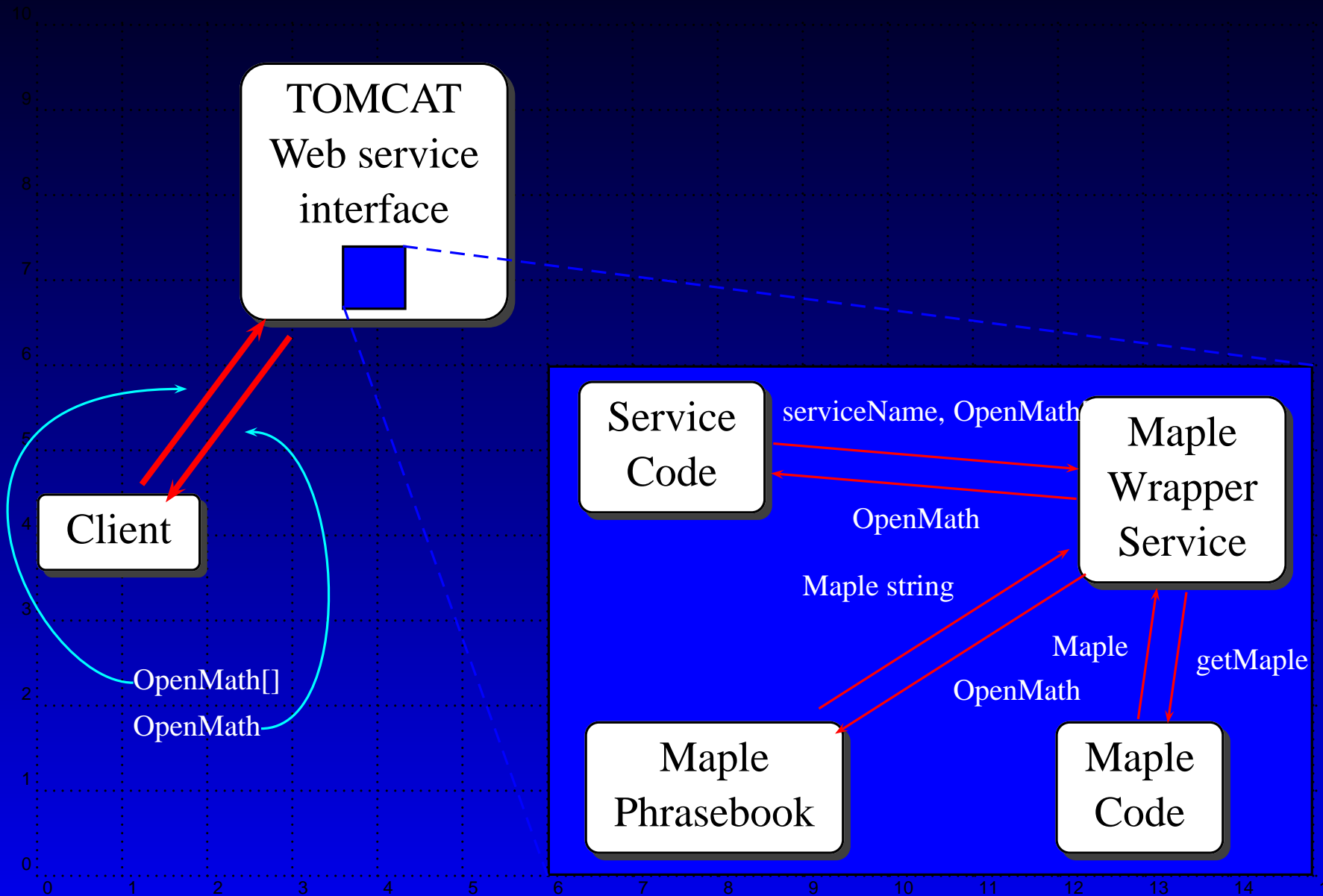
Maple Invocation Web Service

- Unique Invocation web service that handles any maple code.
- Use of the name of the maple service as one of the arguments.
- Consider service arguments (OpenMath) as:
 - Strings.
 - XML (DOM).
 - OpenMath Objects offers better semantics, but involves the implementation of serializers/deserializers of OMObjects.
- Unique WSDL. It relates to a single service which will be capable of executing any of the submitted Maple code.

Maple Invocation Web Service

- How it works..
- Call the service passing an array of OpenMath objects as inputs.
- Service creates a new maple file which:
 - Reads Maple Code and initialises it,
 - Calls the Maple procedure using the interface name, passing the OpenMath arguments (which have been translated into Maple syntax).
- Execute the Maple code.
- Extract output (in MathML-C) and translate to OpenMath.
- Return OpenMath result to the client.

Maple: Invoking a service



Deployment of Maple Web Service

- Use of MSDL to describe the mathematical service.
 - Construction of Java objects using MSDL schema.
 - Dynamic generation of MSDL through JSP pages that conforms to the MONET schema.
 - MSDL files management (update/remove MSDL files)

Broker

- Installation of UWO Broker.
- Objective: register/deregister maple services.
- Add Broker registration/deregistration through JSP pages.

Latest changes

- Necessity of common approach regarding to deployment maple web services:
 - Generation of WSDL for each maple web service using WSDD files.
- MSDL modification. Use of OWL definitions.
- Improvement User help through OpenMath editor and Help web page.
- Registration/Deregistration of maple web services into the Registry Manager using JSP.
- Use of MEL (Explanation language) as the format of the result given by the invocation of services.

Demos

- MSDL Available:

`agentcities.cs.bath.ac.uk:8090/axis/maple-service/msdl/`

- Simple Symbolic Integration:

`MSDL-Simple_Symbolic_Integration.xml`

- Gauss Legendre Integration:

`MSDL-Gauss_Legendre_Integration_Using_Maple.xml`

- Maple Wrapper Service:

`http://agentcities.cs.bath.ac.uk:8090/axis/maple-service`

References

- MONET: monet.nag.co.uk
- AXIS: ws.apache.org/axis/
- TOMCAT: jakarta.apache.org/tomcat/
- JAVAMATH: javamath.sourceforge.net/