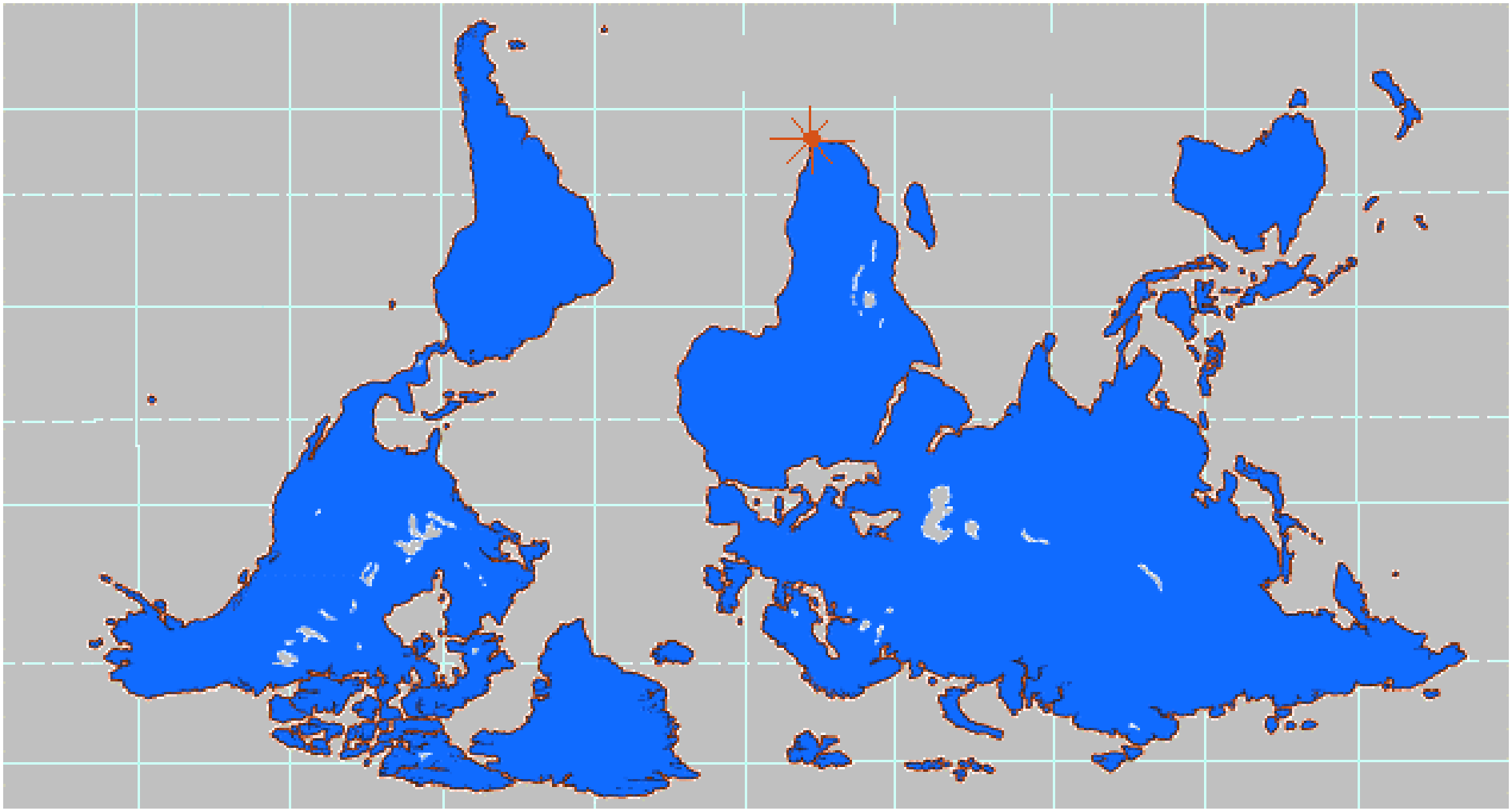
Applying

to

A Quick Geography Lesson



Cape Town, South Africa

Applying SOAP to OAI-PMH

Sergio Congia, Michael Gaylord,
Bhavik Merchant, Hussein Suleman

University of Cape Town, South Africa

September 2004

Overview

- OAI-PMH
- SOAP and Web Services
- Standardisation
- Experiments
  - Data Provider
  - Service Provider
  - Testing Tool
- Results
- Conclusions
- Future Work

OAI-PMH

- Open Archives Initiative Protocol for Metadata Harvesting
  - Transfers “stuff” (record) in XML from machine A to B,
  - Using an HTTP client-server approach,
  - When requested,
  - In discrete batches,
  - Over a period of time.
- OAI-PMH developed in the period 1999-2002.
- Emphasis was on semantics rather than syntax.

SOAP and Web Services

- SOAP is an XML encapsulation message transfer format.
  - Largely aimed at syntax of remote procedure calls.
- SOAP is a core part of the Web Services suite of "specifications".
  - Other specifications are: WSDL, UDDI, WS-Flow, etc.
- SOAP was initiated in 2000, adopted by WSI and standardised by W3C in 2003.
- Support exists in many development tools, but for differing versions.

The Big Question

- Why doesn't the OAI-PMH use SOAP?

Standardisation

- When is a standard stable to use?
  - WSI recommends a different SOAP version than W3C. There are non-trivial differences (e.g., XML Schema requirement in version 1.2).
- Remember OAI-PMH v1.1?
  - Intermediate version released because OAI-PMH v1.0 relied on a pre-final version of XML Schema.
- Implications of pre-standard support
  - Changes in tools.
  - Changes in namespaces.
  - Changes in semantics.

Do we wait indefinitely?

- Some standards are almost there – SOAP was almost there early last year.
- Maybe conduct experiments while standardisation continues?
  - Does a SOAP version present any new encoding problems?
  - Are there performance issues?
  - What about development effort?
  - How simple is it to write a new encoding into OAI-PMH?
  - Is it possible to create multi-interface components/systems?

Outline of Experiments

- 3 student developers with NO prior knowledge of OAI, XML or Web Services.
- SOAP changed version in the middle of the project!

- Procedure
  - Rewrite the OAI-PMH to create SOAP-PMH.
  - Create a data provider, service provider and testing tool.
  - Test for performance and compliance.

SOAP-PMH

- For experimental purposes, sections on requests and responses were rewritten to use SOAP.
- Large portions of the OAI-PMH did not need changes, but the latter half was HTTP-centric.
- Some minor issues (such as the meaning of request URL in SOAP) were ignored.

Experiments: Data Provider

- Custom-written database-driven metadata repository.
- DP support created from scratch.
- Supporting:
  - Multiple metadata formats, resumption tokens, sets.
- Both SOAP-PMH and OAI-PMH supported.

Experiments: Service Provider

- Simple search engine based on Lucene.
- Web-based user interface created with Java servlets to:
  - Submit queries.
  - Manage harvesting operation.
- Both SOAP-PMH and OAI-PMH supported.

Experiments: Testing Tool

- Perform protocol-level compliance tests.
- Standalone application, unlike existing tools.
  - More intuitive user interface.
  - Avoids network latency with remote testing tools.
  - Avoids firewall restrictions and allows testing of "closed" systems.
- Flexibility to choose subsets of tests or perform individual tests.
- Both SOAP-PMH and OAI-PMH supported.

Results

- Usability testing confirmed that the tools with UIs were reasonable.
- Performance testing indicated that switching to SOAP:
  - Did not increase the processing time by much.
  - Increased the response size significantly only if the XML was indented because of the envelope – otherwise, it didn't really matter.
  - Increased the request size if SOAP messages were used for requests, but not by a large amount.

Conclusions 1/2

- Even a group of relative novices can implement SOAP-PMH – so there is no increase in conceptual difficulty.
- Without much increase in complexity, a component/system can support both HTTP/SOAP and HTTP/XML versions of OAI-PMH.
- Care must be taken in encoding data – but, essentially, there is only a marginal increase in message size and processing time.

Conclusions 2/2

- The OAI-PMH mixes syntax and semantics:
  - "request" is HTTP-based
  - "baseURL" is HTTP-based
- Any new specifications should attempt to separate semantics from encoding.

Future Work

- As new standards emerge, a new OAI protocol can be developed to fully fit into the Web Services framework.

- More work is needed to:
  - Design protocols such that the semantics and syntax are not interwoven.
  - Experiment with WSDL and automatic tools.
  - Reconcile REST with SOAP, so that a move towards SOAP does not make the OAI-PMH less RESTful!

That's all Folks!

direct all comments to:

hussein@cs.uct.ac.za