

Project: InterNeuro, eeg.pl	Version: 0.6
Status: draft	Created: 23/06/2003
Author: G.Blinowski	Last rev: 01/10/2008
Title: Preliminary standard for inter-portal SOAP/RDF searching	

1. Introduction & background information

This draft is a result of cooperation between: [Laboratory of Medical Physics, Warsaw University](#), [The Nencki Institute of experimental Biology](#) and [CC Open Computer Systems Ltd](#). This work was sponsored by [Polish State Committee for Scientific Research](#) grant # 4438/IA/115/2003.

Please refer to [Appendix A,B] for general info, standards, methodologies, tools, etc.

2. Distributed query standard for NeuroInformatic sites

The idea is simple - to provide a standard for inter-site searches. We choose a SOAP-over-HTTP approach for the following reasons:

- SOAP/XML is portable and both platform and system independent.
- SOAP/XML and SOAP over http are a de-facto standards for building distributed applications.
- SOAP is simple - there is no heavyweight software required to generate and parse it, the overhead of generating and processing SOAP queries is minimal.
- There is a multitude of XML parsers and tools available (both commercial and open-source) so building software compatible with our format should not be a technical problem.

3. The API

3.1 The Query

This is a proposal for a simple query consisting of: datatype, fulltext and attribute search together with date limiting conditions. I propose a somewhat conservative approach: each RDF DC attribute is passed in parameter-name - value block, an alternative would be to construct a query more like an RDF embedded into SOAP (e.g. refer to: <http://www-106.ibm.com/developerworks/webservices/library/ws-soaprdf/>)... The attribute names are based on the **Dublin Core** standard see: [Appendix B]

The query is formulated as follows (an example):

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:dc="http://purl.org/dc/elements/1.1#" xmlns:neuro="http://www.eeg.pl/soap/
  XMLSchema"
  >
  <SOAP-ENV:Body>
    <NeuroQuery>
      <!-- The first component indicates the object type: Software | Dataset | Type -->
      <QueryType xsi:type="xsd:string">Software</QueryType>
      <!-- The next component is for "full-text" search -->
```

```

<FullTextQuery xsi:type="xsd:string">some pattern here</FullTextQuery>

    <!-- The third component is universal "by-DC-attribute" search -->
    <!-- perhaps we should not permit the following attributes: Date, Type, Identifier -->
<search xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="ns1:searchcrit[3]">
  <item>
    <pname>DC:creator</pname>
    <pvalue>regexp</pvalue>
  </item>
  <item>
    <pname>DC:title</pname>
    <pvalue>regexp</pvalue>
  </item>
  <item>
    <!-- ... and so on for other DC attributes ... -->
  </item>
</search>

    <!-- either AND or OR -->
<searchlogic xsi:type="xsd:string">AND</searchlogic>

    <!-- Date conditions -->
<datebeg xsi:type="xsd:string">2002-01-01</datebeg>
<dateend xsi:type="xsd:string">2003-01-01</dateend>
</NeuroQuery>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

3.2 Returning results

The result is an array of RDF records serialized in a SOAP response.

```

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:dc="http://purl.org/dc/elements/1.1#" xmlns:neuro="http://www.eeg.pl/soap/
  XMLSchema"
  >
  <SOAP-ENV:Body>
<NeuroQueryResponse>

    <!-- general status, e.g. 0 - OK, <0 - error -->
<status xsi:type="xsd:int">0</status>

    <!-- since more than one record may be returned an SOAP array is used -->
<results xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="ns1:res[3]">

    <!-- First result tuple -->
<item rdf:about="http://www.eeg.pl/paper/somepaper">
  <title>A fine paper on EEG</title>
  <dc:date>2003-06-23</dc:date>
  <dc:title>Analysis of EEG signals</dc:title>
  <dc:description>some info here</dc:description>
  <!-- more attributes here -->
  ...
</item>
    <!-- next result tuples come here -->
    ...

```

```
</results>
</NeuroQueryResponse>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2 Fault handling

We may as well not use SOAP faults and transmit all info about error conditions in the status code. If we would like to use faults it could be done like this:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:dc="http://purl.org/dc/elements/1.1#" xmlns:neuro="http://www.eeg.pl/soap/
  XMLSchema"
  >
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultactor>asdasd</faultactor>
    <faultcode>SOAP-ENV:Client</faultcode>
    <faultstring>ERR_SOMETHING_WRONG</faultstring>
    <detail>
      <what xsi:type="xsd:string">aaaaa</what>
    </detail>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Appendix A - Resources

- <http://www.eeg.pl/> - The EEG.pl main site
- <http://www.w3.org/RDF/> - W3C RDF Home
- <http://www.w3.org/XML/> - W3C XML home
- <http://www.dublincore.org/tools/> - Dublin Core Metadata Initiative (DCMI) Tools and Software
- <http://www.soapware.org/bdg> - SoapWare.Org : A Busy Developer's Guide to SOAP 1.1
- <http://www.saxproject.org/> - "The" XML parser

Appendix B - DCMI Type Vocabulary

Taken from: <http://dublincore.org/documents/dcmi-type-vocabulary/>

Collection

A collection is an aggregation of items. The term collection means that the resource is described as a group; its parts may be separately described and navigated

Dataset

A dataset is information encoded in a defined structure (for example, lists, tables, and databases), intended to be useful for direct machine processing.

Event

An event is a non-persistent, time-based occurrence. Metadata for an event provides descriptive information that is the basis for discovery of the purpose, location, duration, responsible agents, and links to related events and resources. The resource of type event may not be retrievable if the described instantiation has expired or is yet to occur. Examples - exhibition, web-cast, conference, workshop, open-day, performance, battle, trial, wedding, tea-party, conflagration.

Image

An image is a primarily symbolic visual representation other than text. For example - images and photographs of physical objects, paintings, prints, drawings, other images and graphics, animations and moving pictures, film, diagrams, maps, musical notation. Note that image may include both electronic and physical representations.

InteractiveResource

An interactive resource is a resource which requires interaction from the user to be understood, executed, or experienced. For example forms on web pages, applets, multimedia learning objects, chat services, virtual reality.

Service

A service is a system that provides one or more functions of value to the end-user. Examples include: a photocopying service, a banking service, an authentication service, interlibrary loans, a Z39.50 or Web server.

Software

Software is a computer program in source or compiled form which may be available for installation non-transiently on another machine. For software which exists only to create an interactive environment, use interactive instead.

Sound

A sound is a resource whose content is primarily intended to be rendered as audio. For example - a music playback file format, an audio compact disc, and recorded speech or sounds.

Text

A text is a resource whose content is primarily words for reading. For example - books, letters, dissertations, poems, newspapers, articles, archives of mailing lists. Note that facsimiles or images of texts are still of the genre text.

PhysicalObject

An inanimate, three-dimensional object or substance. For example -- a computer, the great pyramid, a sculpture. Note that digital representations of, or surrogates for, these things should use Image, Text or one of the other types.