

BrainML Protocols

1. Overview

As described in [specification](#) BrainML describes a format and set of conventions for defining data models and document structure for neuroscience data exchange. (See [diagram](#).) *BrainML-X* is a developing specification that builds on BrainML, defining protocols for query, submission, and transfer of BrainML documents between applications and across networks. This document describes describes the query + response formats, and the submission + acknowledgement formats defined in BrainML-X.

These formats are defined in [XML schemas](#); queries and responses are valid XML documents conforming to these schemas. BrainML-X aware software will accept such queries delivered over a certain transfer protocol (e.g., by HTTP POST), and return responses according to this protocol. Transfer protocols are a developing portion of the specification and not addressed here.

There are two primary categories of query-response exchange: *data* queries and *catalog* queries. These are described in turn below.

2. Data Query and Response

Data queries (also called *selector* queries) are sent to an individual BrainML capable data repository to request data specified by particular criteria. The response is a list of repository data submissions matching the request conditions. In addition, additional data may be returned for each submission as specified by the query.

A data query is made up of two major components, one specifying the conditions and one specifying the return contents. The conditions specification uses boolean operators plus `<field>` tags to specify metadata fields and the (ranges of) values they should have. Here is an example illustrating the possible term specification types:

```
<data_query xmlns="urn:bml/brainml.org:internal/Protocols/3">
  <conditions>
    <and>
      <or>
        <field
          namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
          name="cytoarchitectural_area"
          value="5"/>
```

```

    <field
      namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
      name="cytoarchitectural_area"
      value="7"/>
  </or>
  <field
    namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
    name="electrode_tip_diameter"
    valueMin="0.05" valueMax="0.10"/>
  <and>
    <field
      namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
      name="cell_class"
      values="pyramidal"/>
    <field
      namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
      name="cortical_layer"
      values="surface,intracortical.supragranular/I-III"/>
  </and>
</and>
</conditions>
<return>
  ...
</return>
</data_query>

```

Here, the first two `<field>` tags require that the submissions returned possess the 'cytoarchitectural_area' attribute, with value "5" or "7" (Brodmann areas 5 or 7). The 'value' attribute requires an exact match. The third `<field>` tag specifies a range for numeric attributes. The fifth specifies a list of permissible values for a controlled vocabulary field.

Here is an excerpt illustrating specification of data fields to be returned:

```

<data_query xmlns="urn:bml/brainml.org:internal/Protocols/3">
  <conditions>
    ...
  </conditions>
  <return xml:base="urn:bml/brainml.org:internal/BrainML/2">
    <entity name="experiment" contents="specified">
      <entity name="view">
        <entity name="trace"/>
      </entity>
    </entity>
    <entity name="method" subclasses="none"/>
    <entity name="recording_site" contents="specified"/>
    <entity name="citation"/>
  </return>
</data_query>

```

The specification of fields to be returned works within the hierarchy defined by the data model. If a single `<entity>` element is given, the default action is to return that element for the

submission plus all of its fields and contents. This can be prevented by adding 'fields="specified"' and/or 'contents="specified"' attributes to <entity>. In these cases, only those fields or entities specified by <field> or <entity> child elements will be returned (or none if there are no such child elements). Finally, by default the naming of an entity in the return specification will also cause subclasses of that entity to be returned. Stop this by putting 'subclasses="none"' for that entity.

Notice that the namespace attribute is omitted for all <entity> tags. When this is done, the contextual value of `xml:base` will be used.

Here is an example data query response:

```
<data_response xmlns="urn:bml/brainml.org:internal/Protocols/3">
  <experiment xmlns="urn:bml/brainml.org:internal/BrainML/2">
    ...
  </experiment>
</data_response>
```

Here, <experiment> is the top-level element for a submission in this data model, and the contents under it (not shown) would be the standard contents of the submission as were specified in the return specification.

3. Data Submission and Acknowledgement

The data submission and acknowledgement protocol is used when a client submits data to a BrainML-aware repository. The client sends a submission document, an XML document consisting of a single root element containing identifying information together with one or more top-level data elements whose content is specified by a particular BrainML content model. The server processes the submission and informs the client as to the result.

If the submission is accepted, the server sends back an acknowledgment containing unique IDs assigned to the submission and its components by the server. If there are errors processing the submission or other reasons it cannot be accepted, the acknowledgement is still sent, but with content describing the errors or reason for rejection.

Here is an example submission:

```
<data_submission xmlns="urn:bml/brainml.org:internal/Protocols/3"
  userID="arobert" authKey="3h42kh23k4jh2334k"
  identifier="sub-1">
  <bml:experiment xmlns:bml="urn:bml/brainml.org:internal/BrainML/2"
    xmlns:bmtl="urn:bml/brainml.org:internal/BrainMetaL/1"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns="urn:bml/brainml.org:med.cornell.edu/Cortex/2">
    ... (conformant to Cortex BrainML model XML schema)
```

```
</bml:experiment>
</data_submission>
```

Here, `userID` is the account name of the submitter. BrainML-X does not currently specify authentication methods, however it is assumed that SSL or a similar layer will be used. If said layer requires additional verification of the content itself, such as a digital signature or hash value, this may be provided in the `authKey` attribute. The `identifier` is an arbitrary string (containing only letters, hyphen, and numbers, and starting with a letter) supplied by the client that is used to identify the submission in the subsequent acknowledgement. Finally, the `<experiment>` element is the top-level element defined by the "Cortex" BrainML model specified by the namespace, and conforms to its structure. This element will vary depending on the data model being submitted for.

The acknowledgement takes the following form:

```
<data_acknowledgement xmlns="urn:bml/brainml.org:internal/Protocols/3"
  userID="arobert" identifier="sub-1"
  status="SUCCESS">
  <submittedEntity
    type="urn:bml/brainml.org:internal/BrainML/2,experiment"
    id="arobert-ndb-2111"/>
</data_acknowledgement>
```

Here, `userID` and `identifier` repeat the values given by the submitter and server to identify the submission to the client. The `status` attribute may take on values `SUCCESS`, `FAILURE`, or `PARTIAL`, and indicates whether all, none, or some of the submission components were accepted without errors.

The acknowledgement will contain one child element for each submission component. For those that were accepted, this will be `<submittedEntity>`. Its `id` attribute indicates the ID assigned by the server that uniquely identifies the submission. For those that were NOT accepted, this will be `<error>`. These elements look like the following:

```
<error code="134">
  This is some text describing the error.
</error>
```

The error codes are specific to a server and are documented by it. (Codes below 100 are reserved for BrainML use but are not defined yet.) The text description beneath the error element should describe the error type in general as well as any specific information particular to the occurrence.

4. Login and Response

BrainML-X does not mandate an authentication protocol, however it provides a default one that

can be used if desired. The main requirement for any protocol is that it provide a limited-lifetime authorization key back to the client that can be used for subsequent authenticated interactions.

In the BrainML-X default authorization protocol, *login* requests are sent to a data server as a preliminary to submission, and, in some cases, queries. The login request contains a user ID and password in plain text in the XML, and the response contains an opaque authorization key. Subsequent requests should include both the key and the user ID.

Example

```
LOGIN
<login xmlns="urn:bml/brainml.org:internal/Protocols/3"
      userID="foo" password="fooPass" />

RESPONSE
<login_response xmlns="urn:bml/brainml.org:internal/Protocols/3"
               authKey="sdlfjh2342kb13fsfdw334k" />
```

Here, the response can contain arbitrary application-specific content in addition to the authorization key. A value of "ERROR" will be returned for the key if login was unsuccessful. Optionally, if the server desires to provide the information, either "ERROR-USER", "ERROR-PASSWORD", or "ERROR-EXPIRED" may be sent. (Providing such clarifying information is sometimes considered a security flaw, however we feel the usefulness outweighs this.) Valid keys will not contain spaces or the string "ERROR".

The login sends a username and password in plain-text, which may be considered a security risk. Additional security can be provided by running the login protocol over a transport layer, such as SSL, that provides for encryption, or by implementing another authentication protocol entirely. This is left for applications to decide.

5. Catalog Query and Response

Catalog queries (also called *detector* queries) are sent to a special "catalog server" which maintains a list of internet-accessible BrainML-capable data repositories indexed by the BrainML models they contain data for. The queries specify a list or boolean combination of BrainML models, and the server responds with a list (possibly empty) of URLs for providing servers.

Example

```
QUERY
<catalog_query xmlns="urn:bml/brainml.org:internal/Protocols/3">
  <and>
    <or>
```

```

    <entity namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
      name="cortical_recording_location"/>
    <entity namespace="urn:bml/brainml.org:med.cornell.edu/Cortex/2"
      name="cortical_recording_source"/>
  </or>
  <field entity="urn:bml/brainml.org:internal/BrainML/2"
    name="data_class" value="membrane potential.spike times"/>
</and>
</catalog_query>

```

```

RESPONSE
<catalog_response xmlns="urn:brainml.org:internal/Protocols/3">
  <repository url="http://neurodatabase.org"/>
  <repository url="http://footech.edu"/>
</catalog_response>

```

Here, the query can contain an arbitrary nesting of `<and>` and `<or>` tags, together with `<model>`, `<entity>`, and `<field>` tags specifying the contents queried for. A `<model>` tag requests a repository serving data for a particular model, whereas an entity requests a repository serving that particular entity, whether used as part of the model it is defined in or a model derived from it.

The `<field>` tag requests a repository serving that field with particular a particular value or values.

The response simply lists the servers found. The recipient may contact them individually as desired.

6. Catalog Submission and Acknowledgement

Finally, a protocol for submission to a catalog server is also provided. This is essentially identical to the protocol for data submission (see above) with the exception that the word 'catalog' is substituted for the word 'data' in all tags. Thus, submissions are sent with `<catalog_submission>` and acknowledged by `<catalog_acknowledgement>`.

7. Data and Catalog Updates

In some cases a server may support the ability to update existing submissions. The tags for update are similar to those for data and catalog submission with the word 'update' substituted for the word 'submission' in all tags. The entry to update is determined by examining the ID attribute of each top-level submission component, so this should be set to the ID of an existing submission to modify.

Permission to update submissions is usually restricted to administrators and/or the original submitter user account.

8. Administrative Functions

BrainML-X also provides for messages implementing administrative functionality. For these, the tags `<admin>`, `<admin_result>`, and `<admin_error>` are used. Please refer to the schema documentation and documentation at the particular implementing server for more details.