

Version: 1.0
Date: September 30th 2003

The Monet Mathematical Service Response Ontology

Yannis Chicha and David Roberts

Deliverable D18A (Public)

Contents

1	Introduction	2
2	Anatomy of a Response	2
3	Extract from Monet Schema Describing a Service Response	3
4	Examples of an Execution Response	3
4.1	Example 1	3
4.2	Example 2	4
4.3	Example 3	5
4.4	Example 4	6

1 Introduction

When a fully-resolved plan is executed, the execution response is described by the Service Response ontology and expressed in MEL(Mathematical Explanation Language). The schema and ontology described here is an implementation of the explanation ontology described in deliverable D07. The response document has a recursive structure which allows the result of a plan to contain many service responses. An individual response contains the result (if any) of the service's execution and optional elements describing the choices made by the service, errors returned and any additional information.

2 Anatomy of a Response

The result returned by a service or plan should be wrapped in an 'executionResponse' tag. This tag has the following attributes and elements:

href a required attribute that acts as a unique identifier identifying the plan or service that this is a response from. If a response to a service, this href is a concatenation of the 'targetNamespace' in the 'definitions' tag of the MSDL file for the service and the 'name' attribute of the 'service' tag in the MSDL file. If this is the response to a plan the href should be a unique URI for the plan.

explanation an element describing the choices made by the service to solve a problem. This element is Required.

result an element encapsulating the result e.g. expressed in openmath.

resultAdditionalInfo an element containing information characterizing the result

problem an element describing the input parameters to the service. This component is redundant for all problems whose inputs were stated in the plan submitted for execution. However, the 'problem' tag should be present when the input parameter of one service call has been supplied by the output of another service call. In this case, the value was not known in advance and the 'problem' element might provide useful information to a client.

The explanation, result and resultAdditionalInfo elements can be expressed using any formalism (e.g. the result may be an OpenMath object). Any formalism used should also be referenced in the 'monet' ontology so that clients can understand the results returned.

As mentioned in the Introduction, the response ontology has a recursive structure. Thus the response to a plan containing many service responses would also be a document such as the one described above. In this case the 'explanation' tag would contain the service responses from each service executed by the plan (see 4.4 on page 6 below). Providing such information allows the client to obtain all possible details about the resolution of the submitted problem: what plan was chosen, how it executed, and what were the characteristics of the intermediate results. The result component would contain the final result taken from the computation of the individual results of each service.

The 'explanation' tag is required. Within the explanation tag the 'isProvided' attribute is required, this attribute is either true or false. If the Web Service is able to provide explanation information this value is true, otherwise false; if the service provides explanations but chooses not to return any in this case isProvided="true" should still be returned. The 'errorcode' element is required within the explanation tag, this errorcode can take values -1, 0, 1. A returned value

of 0 indicates a result which exactly matches the client's query is returned, -1 indicates that an error has occurred and no result can be returned, 1 indicates a result is returned but that this result does not meet all the criteria of the query - for example, some of the constraints of the query may not have been met. Thus the minimum required return from a service is an explanation tag with an errorcode (see 4.1 on page 3 and other examples).

The schema used for expressing results the is shown in the next section.

3 Extract from Monet Schema Describing a Service Response

The schema describes the result of executing a service or a plan:

```
<xsd:element name="executionResponse" type="monet:tExecutionResponse"/>

<xsd:complexType name="tExecutionResponse">
  <xsd:complexContent>
    <xsd:extension base="monet:tExtensibleDocumented">
      <xsd:sequence>
        <xsd:element ref="monet:problem" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="explanation" type="monet:tExplanation" maxOccurs="1"/>
        <xsd:element name="result" type="monet:tResult" minOccurs="0"
          maxOccurs="1"/>
        <xsd:element name="resultAdditionalInfo" type="monet:tResultAdditionalInfo"
          minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:ID" use="optional"/>
      <xsd:attribute name="href" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

If the results of more than one plan are required, many 'executionResponse' tags may be wrapped in a 'query-response' element (see 4.4 on page 6 below) in a similar way to 'query' elements (see deliverable D13).

4 Examples of an Execution Response

4.1 Example 1

The service failed to return a result, no further explanation of errors is able to be provided by the service. The errorcode = -1 indicates that no result was returned.

NB the attribute 'isProvided="false"' indicates that the service is incapable of providing a further explanation, if the service provides explanations but chooses not to return any in this case a tag with isProvided="true" should be returned.

```
<executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
  href="http://anyuri.org/services/aMathService1">

  <explanation>
    <errorCode>-1</errorCode>
    <explanationFormat isProvided="false"/>
  </explanation>

</executionResponse>
```

4.2 Example 2

A Result is obtained from the service, no further explanation is able to be provided by the service. The errorCode=0 indicates that a result is returned that meets exactly the query's requirements. The result is returned in OpenMath format which is indicated to the client by the 'resultFormat' element. The href of the resultFormat element should be a recognized Monet ontology class.

NB the attribute 'isProvided="false"' indicates that the service is incapable of providing a further explanation, if the service provides explanations but chooses not to return any in this case a tag with isProvided="true" should be returned.

```
<executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
  href="http://anyuri.org/services/aMathService2">

  <explanation>
    <errorCode>0</errorCode>
    <explanationFormat isProvided="false"/>
  </explanation>

  <result>
    <resultFormat href="http://monet.nag.co.uk/monet/result#openmath">
      <OMA xmlns="http://www.openmath.org/OpenMath">
        <OMS cd="nums1" name="rational"/>
        <OMI>1</OMI>
        <OMI>3</OMI>
      </OMA>
    </resultFormat>
  </result>

</executionResponse>
```

4.3 Example 3

A result is returned by the service. The service also provides an explanation of the result. The errorcode=1 indicates that a result is returned but it does not exactly meet the requirements of the query. The format of the explanation is indicated by the href attribute of the 'explanation-Format' element. This href should be a recognized Monet ontology class. The formats contained in the result element and the resultAdditionalInfo element are indicated in a similar way.

```

<executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
  href="http://anyuri.org/services/aMathService3">

  <explanation>
    <errorcode>1</errorcode>
    <explanationFormat isProvided="true"
      href="http://monet.nag.co.uk/monet/explanation#expl1">
      <myExplanation xmlns="http://anyuri.com/explain">
        <Algorithmn>
          MySpecialAlgo
        </Algorithmn>
      </myExplanation>
    </explanationFormat>
  </explanation>

  <result>
    <resultFormat href="http://monet.nag.co.uk/monet/result#openmath">
      <OMA xmlns="http://www.openmath.org/OpenMath">
        <OMS cd="nums1" name="rational"/>
        <OMI>1</OMI>
        <OMI>3</OMI>
      </OMA>
    </resultFormat>
  </result>

  <resultAdditionalInfo>
    <addInfoFormat isProvided="true">
      <myAdditionalInfo xmlns="http://anyuri.com/addInfo">
        <accuracy>
          5dp
        </accuracy>
      </myAdditionalInfo>
    </addInfoFormat>
  </resultAdditionalInfo>

</executionResponse>

```

4.4 Example 4

A response to a Plan containing the results of its component services as an explanation.

```

<query-response
  xmlns="http://monet.nag.co.uk/monet/ns">
  <!--
    N.B. query-response tag can contain responses
    to many plans. Here we have one top-level
    executionResponse tag, which is the reponse
    to one plan: '...#plan6'. This response
    contains the responses from the 3 services
    that are executed by the plan.
  -->
  <executionResponse
    href="http://monet.nag.co.uk/monet/plans#plan6">
    <explanation>
      <errorCode>-1</errorCode>
      <explanationFormat isProvided="true"/>
  <!-- Service 1 response -->
    <executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
      href="http://anyuri.org/services/aMathService1">
      <explanation>
        <errorCode>-1</errorCode>
        <explanationFormat isProvided="false"/>
      </explanation>
    </executionResponse>
  <!-- Service 2 response -->
    <executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
      href="http://anyuri.org/services/aMathService2">
      <explanation>
        <errorCode>0</errorCode>
        <explanationFormat isProvided="false"/>
      </explanation>
      <result>
        <resultFormat href="http://monet.nag.co.uk/monet/result#openmath">
          <OMA xmlns="http://www.openmath.org/OpenMath">
            <OMS cd="nums1" name="rational"/>
            <OMI>1</OMI>
            <OMI>3</OMI>
          </OMA>
        </resultFormat>
      </result>
    </executionResponse>
  <!-- Service 3 response -->
    <executionResponse xmlns="http://monet.nag.co.uk/monet/ns"
      href="http://anyuri.org/services/aMathService3">
      <explanation>
        <errorCode>1</errorCode>

```

```

    <explanationFormat isProvided="true"
      href="http://monet.nag.co.uk/monet/explanation#expl1">
      <myExplanation xmlns="http://anyuri.com/explain">
        <Algorithmn>
          MySpecialAlgo
        </Algorithmn>
        <ExecutionTime>
          12ms
        </ExecutionTime>
      </myExplanation>
    </explanationFormat>
  </explanation>

  <result>
    <resultFormat href="http://monet.nag.co.uk/monet/result#openmath">
      <OMA xmlns="http://www.openmath.org/OpenMath">
        <OMS cd="nums1" name="rational"/>
        <OMI>1</OMI>
        <OMI>3</OMI>
      </OMA>
    </resultFormat>
  </result>

  <resultAdditionalInfo>
    <addInfoFormat isProvided="true">
      <myAdditionalInfo xmlns="http://anyuri.com/addInfo">
        <accuracy>
          5dp
        </accuracy>
      </myAdditionalInfo>
    </addInfoFormat>
  </resultAdditionalInfo>

  </executionResponse>
</explanation>

</executionResponse>
</query-response>

```