

Web Services Standards for Data Mining

Robert Chu

Abstract

Most, if not all, data mining and scoring tool providers require users to use provider-specific ways to invoke their services. The provider-specific approach could be a major factor affecting why data mining tools and applications are not currently as widespread as one might hope. Web services standards can address these proprietary issues. This article discusses what web services are, in general, as well as in the context of data mining and scoring. The intended readers are data mining practitioners who are new to web services.

1. Web Services

One not-so-rigorous description of web services is as follows: A web service client passes a request in text while the service provider acts on the request and returns text to the client, all via the Web. Plain old web browsing is a form of web services: a user sends "<http://cnn.com>", for example, from a web browser to the CNN main web server which sends its home page back to the requesting browser in text. Web services are identical in concept to this process. However, complicated web services often involve richer content as input than simple web page browsing with web services. XML[8] is most often used to format the input. As to the output, the contrast between web browsing and web services is not about whether or not the content is complicated, but rather whether the format is HTML or not. Even though it is not entirely technically correct, one can view an HTML document as an instance of an XML document. However, HTML is particularly designed for web browser consumption, while an XML document is designed for a specific business need. It wouldn't be complete to describe web services without mentioning the SOAP[5] protocol. Keep the following notes in mind if you are new to SOAP: SOAP is not really a simple protocol and "object" has nothing to do with the protocol. Fortunately, you most likely will have no need to understand SOAP as it should be transparent to you unless you deal with related low-level programming.

The Worldwide Web is based on the HTTP protocol. Currently the SOAP protocol fits nicely on top of the HTTP protocol. The ubiquities of HTTP and low-cost HTTP-based web servers are catalysts to the quick and widespread adoption of web services. The data mining industry can take full advantage of the cost factor.

2. Web Services for Data Mining and Scoring

Let us use examples to illustrate web services for data mining.

Example 1.

John has 5 (x, y) data points: (1, 12.1), (2, 14.2), (3, 16.1), (4, 18.2), and (5, 20.1) and would like to fit the following regression model: $y = a + b x$.

John sends a simple web service via email like the following:

Hi Fred,

I have 5 (x, y) data points: (1, 12.1), (2, 14.2), (3, 16.1), (4, 18.2), and (5, 20.1). Could you help me fit the regression model $y = a + bx$? Thanks for your time.

Your best friend,

John

Thirty minutes later, John gets the following response from Fred:

Hi John,

a is 0, b is 2. Let me know if you need more help.

Fred

Example 2.

Data is the same as in example 1.

John sends Fred email with the request in XML format:

Fred,

Please help me fit the regression model:

```
<BuildModel>
  <RegressionModel>
    <Target>y</Target>
    <Intercept/>
    <Predictor>x</Predictor>
  </RegressionModel>
  <InlineTable>
    <row><x>1</x><y>12.1</y></row>
    <row><x>2</x><y>14.2</y></row>
    <row><x>3</x><y>16.1</y></row>
    <row><x>4</x><y>18.2</y></row>
    <row><x>5</x><y>20.1</y></row>
  </InlineTable>
</BuildModel>
```

If you will, could you describe the result in XML as well? Much obliged. John

Being tired of reading the XML document John sent, Fred responded three days later with the following:

John,

Here is the result:

```
<RegressionTable>
  <Intercept>10</Intercept>
  <Parameter name="x">2</Parameter>
</RegressionTable>
```

Fred.

Example 3.

The request text is the same as in example 2, but this time John does not send email, instead John copies and pastes the request text in a text box on a window of a data mining tool and clicks on the submit button. The request is sent to a remote data mining server. This time, instead of three days, John gets the modeling results back in one second.

Example 4.

The data for this example is the same as in example 1, but is stored in a Microsoft Excel worksheet. Someone wrote an Excel add-in for John. John just launches the add-in GUI and specifies the data source and modeling settings by point and click. John then submits the model build request to a remote data mining server. The result is returned in one second. John doesn't see any XML string flowing back and forth between Excel and the remote data mining server. The web service details are simply transparent to any user.

Please note that all the examples above use an embedded data source and skip the connection parameters for easy illustration purposes. In the real world, the data source can be in a database and connection parameters are typically supplied in the request XML string.

3. Web Services Standards for Data Mining

As you can imagine from example 2 in the previous section, just using XML can lead to multiple flavors of XML formats to describe input and output for data mining. Without a XML data mining standard, if you switched from one data mining provider to another, you would most likely need to rewrite your code.

Currently, there are two publicly available data mining related web services specification standards: JDM API web services extensions[1] (JDMWS) and XML for Analysis Specification[2]. JDMWS is based on the object models used in the JDM API specification, while XML for Analysis reuses OLE DB for Data Mining Schema Rowsets. The next two sections show simple examples for each specification. It is not the intention of this article to rigorously compare these two specifications. Our intention is only to promote the idea of web services standards for data mining in general.

4. JDM Web Service Examples

Java Specification Request 73: Java Data Mining (JDM) Version 1.0 is a pure Java API (Applications Programming Interface) to facilitate the development of data mining and scoring-enabled applications. It includes web services extensions (JDMWS). The following three example fragments are based on the specification to show readers what JDMWS strings look like. It is not an intention of this article to give an overview or tutorial of JDMWS, so the explanation is brief.

Example 1.

```
<SOAP-ENV:Body>
<saveObject xmlns="http://www.jsr73.org/2004/webservices/"
  xmlns:jdm="http://www.jsr73.org/2004/JDMSchema"
  name="CampaignSettings-101" overwrite="true" verify="true">
  <object xsi:type="ClassificationSettings" miningFunction="classification">
    <algorithmSettings algorithm="naiveBayes" pairwiseThreshold="0.1"
      singletonThreshold="0.1"/>
    <buildAttribute attributeName="Job" usage="active" outlierTreatment="asMissing"/>
    <buildAttribute attributeName="Gender" usage="active" outlierTreatment="asIs"/>
    <buildAttribute attributeName="Education" usage="active" outlierTreatment="asIs"/>
    <buildAttribute attributeName="customerID" usage="inactive"/>
  </object>
</saveObject>
</SOAP-ENV:Body>
```

Each object that can be persisted in a JDM-based Data Mining Engine has a type and a unique name. This example shows that an object of type ClassificationSettings is saved. Later this object can be retrieved by the type and the name. Named objects are to promote object reuse.

Example 2.

```
<SOAP-ENV:Body>
<executeTask xmlns="http://www.jsr73.org/2004/webservices/">
  <task xsi:type="BuildTask" name="CampaignBuildTask-26">
    <objectName>CampaignBuildTask_106</objectName>
    <modelName>Campaign_106</modelName>
    <buildDataName>Campaign20040115</buildDataName>
    <buildSettingsName>CampaignClassificationSettings</buildSettingName>
  </task>
</executeTask>
</SOAP-ENV:Body>
```

A JDM-based task can be defined and persisted in a JDM-based Data Mining Engine. To execute a JDM task is to send a web service request that is defined by specifying a pre-defined task and then associating a few related resource objects.

Example 3.

```
<SOAP-ENV:Body>
<executeTask xmlns="http://www.jsr73.org/2004/webservices/">
  <task xsi:type="RecordApplyTask" modelName="CampaignClassification106">
    <recordValue name="Job" value="Sales Management"/>
    <recordValue name="Gender" value="F"/>
    <recordValue name="Education" value="College"/>
    <recordValue name="CustID" value="20040214-5673"/>
    <applySettingsName xsi:type="ClassificationApplySettings">
      <sourceDestinationMap sourceAttrName="CustID" destinationAttrName="CustomerID"/>
      <applyMap content="predictedCategory" destPhysAttrName="churn" rank="1"/>
      <applyMap content="probability" destPhysAttrName="churnProb" rank="1"/>
    </applySettingsName>
  </task>
</executeTask>
</SOAP-ENV:Body>
```

This example shows a single-record-scoring web service.

5. XML for Analysis for DM Examples

XML for Analysis (XMLA) is a Simple Object Access Protocol (SOAP)-based XML API designed specifically for standardizing the data access interaction between a client application and a data provider working over the Web. XMLA addresses both OLAP (OnLine Analytical Processing) and data mining. The following three example fragments are based on the specification and the accompanying OLE DB for Data Mining Specification Version 1.0[4] to show readers what XMLA strings look like. It is not an intention of this article to give an overview or tutorial of XMLA for Data Mining, so the explanation is brief.

Example 1.

```
<SOAP-ENV:Body>
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <Command>
    <Statement>
      CREATE MINING MODEL [MemberCards]
      (
        [customer Id] LONG KEY ,
        [Yearly Income] TEXT DISCRETE ,
        [Member Card Type] TEXT DISCRETE PREDICT,
        [Marital Status] TEXT DISCRETE
      )
      USING VendorA_Decision_Trees
    </Statement>
  </Command>
  <Properties>
    ...
  </Properties>
</Execute>
</SOAP-ENV:Body>
```

This example illustrates an XML string and an OLE DB for Data Mining script for building a decision tree mining model skeleton. [Member Card Type] is the target column since the keyword “PREDICT” is specified for the column.

Example 2.

```
<SOAP-ENV:Body>
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <Command>
    <Statement>
      INSERT INTO [MyModel]
      // Define the list of columns to be populated
      (
        [Name], [Age], [Hair Color]
      )
      OPENROWSET
      (
        'SQLOLEDB', 'Initial Catalog=FoodMart 2000',
        'Select [Name], [Age], [Hair Color] FROM [Customers]'
      )
    </Statement>
  </Command>
  <Properties>
    ...
  </Properties>
</Execute>
</SOAP-ENV:Body>
```

This example illustrates a XML string plus a SQL-like script for building a data mining model.

Example 3.

```
<SOAP-ENV:Body>
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <Command>
    <Statement>
      SELECT t.[Customer ID], [Age Prediction].[Age]
      FROM [Age Prediction]
      PREDICTION JOIN
      (
        SHAPE
        {
          SELECT [Customer ID], [Gender],
          FROM Customers
          ORDER BY [Customer ID]
        }
        APPEND
        (
          {SELECT [CustID], [Product Name], [Quantity]
          FROM Sales
          ORDER BY [CustID]
          }
          RELATE [Customer ID] To [CustID]
        )
        AS [Product Purchases]
      ) as t
      ON [Age Prediction] .Gender = t.Gender and
      [Age Prediction] .[Product Purchases].[Product Name]
      = t.[Product Purchases].[Product Name] and
      [Age Prediction] .[Product Purchases].[Quantity]
      = t.[Product Purchases].[Quantity]
    </Statement>
  </Command>
  <Properties>
    ...
  </Properties>
</Execute>
</SOAP-ENV:Body>
```

This example illustrates how a XML string is used for scoring based on a trained mining model.

Note that both JDMWS and XMLA for Data Mining support generation of PMML-based[3] model output.

6. Issues in Web Services for Data Mining

Data Security – By default, a data mining web services client and server may communicate with each other in clear text. Data integrity and confidentiality could be compromised. WS-Security[6] is a good starting point for exploring the web service security issues.

Asynchronous Web Services – A data mining task could be long-running. Asynchronous operations should be considered, otherwise, a client application may be undesirably blocked until the service results are returned.

Data Mining Session State Management – Web services is a stateless operation. One can simulate a stateful session by including a session ID in web services strings.

WS-Resource Framework – Many computers (often many cheap PC's) can be grouped together to function as a virtual supercomputer for data mining tasks. WS-Resource Framework[7] is a standard that can facilitate such virtual supercomputers.

7. Conclusion

Data mining, web services, and software standards in general are all growing in popularity. Data mining web services standards in particular are a sure bet for those data mining practitioners who are looking for technologies to improve their organizations' competitive edges. This article briefly introduces web services in general, and web services in data mining in particular. Examples were given to simply illustrate the styles of current popular data mining web services standards – JDMWS[1] and XMLA for DM [2&4]. Related issues and suggested resolutions were also discussed.

References

- [1] Java Specification Request 73: Java Data Mining (JDM), Version 1.0, Final Review.
- [2] XML for Analysis Specification version 1.0.
- [3] Predictive Model Markup Language, Version 2.1.2, <http://www.dmg.org>.
- [4] OLE DB for Data Mining Specification, Version 1.0.
- [5] SOAP Version 1.2, <http://www.w3.org/TR/soap/>.
- [6] WS-Security, <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>.
- [7] WS-Resource Framework, <http://www.globus.org/wsrf/>.
- [8] XML Specification, <http://www.w3.org/TR/2000/REC-xml-20001006>.

Acknowledgement

Thanks to the XML for Analysis Council and Microsoft for the permission to use examples described in the XML for Analysis Specification and the OLE DB for Data Mining Specification, respectively.