

```

<?xml version = "1.0" encoding = "UTF-8"?>
  <xsd:schema xmlns = "http://www.w3.org/Incubator/2010/03/30/decisions2"
    targetNamespace = "http://www.w3.org/Incubator/2010/03/30/decisions2"
    xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:georss="http://www.georss.org/georss"
    xmlns:gml="http://www.opengis.net/gml"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <xsd:element name="decisions" type="DecisionsType"/>

    <xsd:complexType name="DecisionsType">
      <xsd:sequence>
        <xsd:element name="comment" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="decision" minOccurs="0" maxOccurs="unbounded"
type="DecisionType"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="ValueListURIType">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The ValueListURI concept, borrowed from OASIS's
          ValueListURN, allows flexibility for organizations or communities to
          define their own lists of values
          and manage them externally. If enumerated lists were used (where the
          list of values is hard-coded
          into the schema), validation would be easier. The tradeoff in loss
          of easy validation is considered
          well worth the benefits of added flexibility (and often necessary
          flexibility) and tailoring allowed. Also
          validation is still possible, although more difficult, because the
          ValueListURI structure contains the unique name of the list,
          which could be a URL for easy lookup of list values to support a
          more sophisticated form of validation.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="listURI" type="xsd:anyURI"/>
        <xsd:element name="value" type="xsd:string" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TripleType">
      <xsd:sequence>
        <xsd:element name="subject" type="xsd:anyURI"/>
        <xsd:element name="verb" type="xsd:anyURI"/>
        <xsd:element name="object" type="xsd:anyURI"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="PedigreeType">
      <xsd:sequence>
        <xsd:element name="assertedBy" type="EventInfoType" minOccurs="0"

```

```

maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="BasicStatementType">
  <xsd:sequence>
    <xsd:element name="guid" type="xsd:anyURI" minOccurs="0" />
    <xsd:element name="title" type="xsd:string" minOccurs="0" />
    <xsd:element name="description" type="xsd:string" minOccurs="0" />
    <xsd:element name="state" type="DecisionStateType" minOccurs="0" />
    <xsd:element name="link" type="xsd:anyURI" minOccurs="0" />
    <xsd:element name="triple" type="TripleType" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="StatementType">
  <xsd:complexContent>
    <xsd:extension base="BasicStatementType">
      <xsd:sequence>
        <xsd:element name="pedigree" type="PedigreeType" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StateType">
  <xsd:complexContent>
    <xsd:extension base="ValueListURIType">
      <xsd:sequence>
        <xsd:element name="date" type="xsd:dateTime" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DecisionType">
  <xsd:sequence>
    <xsd:element name="guid" type="xsd:anyURI" />
    <xsd:element name="question" type="StatementType" />
    <xsd:element name="options" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="option" minOccurs="0" maxOccurs="unbounded"
type="OptionType" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="confidence" minOccurs="0" maxOccurs="1"
type="ConfidenceType" />
    <xsd:element name="subdecisions" minOccurs="0" maxOccurs="1"
type="DecisionsType" />
    <xsd:element name="notes" minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="note" minOccurs="0" maxOccurs="unbounded"
type="StatementType" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

    </xsd:element>
    <xsd:element name="references" minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="reference" minOccurs="0"
maxOccurs="unbounded" type="StatementType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="decisionState" type="DecisionStateType">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Default Decision State URI should be: http://www.w3.org/
Incubator/ontologies/2009/9/DecisionOntology_ver_01.owl#DecidingState
          Values from this list include: NotStartedState,
GatheringInfoState, AnalyzingInfoState, DecidingState,
ProducingDecisionProductState,
          CommunicatingDecisionState, ReceivingFeedbackState, DoneState
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="decisionEventInfo" type="EventInfoType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OptionType">
  <xsd:sequence>
    <xsd:element name="idea" type="StatementType"/>
    <xsd:element name="selected" type="xsd:boolean"/>
    <xsd:element name="pros" minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="pro" minOccurs="0" maxOccurs="unbounded"
type="ProConType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="cons" minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="con" minOccurs="0" maxOccurs="unbounded"
type="ProConType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DecisionStateType">
  <xsd:complexContent>
    <xsd:extension base="ValueListURIType">
      <xsd:sequence>
        <xsd:element name="date" type="xsd:dateTime" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

```

</xsd:complexType>

<xsd:complexType name="ProConType">
  <xsd:complexContent>
    <xsd:extension base="StatementType">
      <xsd:sequence>
        <xsd:element name="subdecisions" minOccurs="0"
type="DecisionsType"/>
        <xsd:element name="criterionEvaluated" minOccurs="0"
type="ValueListURIType">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              criterionEvaluated is intended to be a selected value
or values from a list of possible values
              for that particular criterion. For example, if the
criterion is cost, then the possible values might be
              Free, Inexpensive, Reasonable, Expensive,
VeryExpensive.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="ConfidenceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Very High"/>
    <xsd:enumeration value="High"/>
    <xsd:enumeration value="Medium"/>
    <xsd:enumeration value="Low"/>
    <xsd:enumeration value="Very Low"/>
    <xsd:enumeration value="Unknown"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="EventInfoType">
  <xsd:sequence>
    <xsd:element name="who" type="BasicStatementType"/>
    <xsd:element name="what" minOccurs="0" type="BasicStatementType"/>
    <xsd:element name="where" minOccurs="0" type="BasicStatementType"/>
    <!--xsd:element name="whereLatLon" type="georss:where"/-->
    <xsd:element name="when" minOccurs="0" type="xsd:dateTime"/>
    <xsd:element name="how" minOccurs="0" type="BasicStatementType"/>
    <xsd:element name="why" minOccurs="0" type="BasicStatementType"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```