experience and knowledge about the topic, at the time of the topic's study, without benefit from hindsight.

Type definition in XML Schema:

```
<xs:complexType name="PrimarySource">
  <xs:complexContent>
    <xs:extension base="prov:Derivation">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:hadPrimarySource` is used to reference a prov:PrimarySource from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="hadPrimarySource" type="prov:Prin
```

EXAMPLE 17: prov:hadPrimarySource

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:entity prov:id="ex:la-campagne-de-Russie-1812-1813">
    <prov:type xsi:type="xsd:QName">map</prov:type>
  </prov:entity>

  <prov:entity prov:id="ex:revue-d-Histoire-de-la-Pharmacie-t-XVIII">
    <prov:type xsi:type="xsd:QName">journal</prov:type>
  </prov:entity>

  <prov:hadPrimarySource>
    <prov:generatedEntity prov:ref="ex:la-campagne-de-Russie-1812-1813"/>
    <prov:usedEntity prov:ref="ex:revue-d-Histoire-de-la-Pharmacie-t-XVIII"/>
  </prov:hadPrimarySource>

</prov:document>
```

no prefix / name space

3.3 Component 3: Agents, Responsibility, and Influence

The third component of PROV-DM is concerned with agents and the relations WasAttributedTo (Attribution), WasAssociatedWith (Association), ActedOnBehalfOf (Delegation), relating agents to entities, activities, and agents, respectively.

3.3.1 Agent

An agent is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

Type definition in XML Schema:

```
<xs:complexType name="Agent">
  <xs:sequence>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
```

```
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>
```

The element prov:agent is used to reference a prov:Agent from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="agent" type="prov:Agent"/>
```

EXAMPLE 18: prov:agent

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:agent prov:id="e1">
    <prov:type xsi:type="xsd:QName">prov:Person</prov:type>
    <ex:name>Alice</ex:name>
    <ex:employee>1234</ex:employee>
  </prov:agent>

</prov:document>
```

3.3.1.1 Person

Agents of type Person are people.

Type definition in XML Schema:

```
<xs:complexType name="Person">
  <xs:complexContent>
    <xs:extension base="prov:Agent">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element prov:person is used to reference a prov:Person from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="person" type="prov:Person"/>
```

EXAMPLE 19: prov:person

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:person prov:id="ex:Paolo" />

</prov:document>
```

3.3.1.2 Organization

Agents of type Organization are social or legal institutions such as companies, societies, etc.

Type definition in XML Schema:

```
<xs:complexType name="Organization">
  <xs:complexContent>
    <xs:extension base="prov:Agent">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:organization` is used to reference a prov:Organization from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="organization" type="prov:Organiza
```

EXAMPLE 20: prov:organization

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:organization prov:id="w3:Consortium" />

</prov:document>
```

prefix not defined.

3.3.1.3 Software Agent

A SoftwareAgent is running software.

Type definition in XML Schema:

```
<xs:complexType name="SoftwareAgent">
  <xs:complexContent>
    <xs:extension base="prov:Agent">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:softwareAgent` is used to reference a prov:SoftwareAgent from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="softwareAgent" type="prov:Softwar
```

EXAMPLE 21: prov:softwareAgent

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">
```

```
    <prov:softwareAgent prov:id="ag" />
  </prov:document>
```

3.3.2 Attribution

Attribution is the ascribing of an entity to an agent.

Type definition in XML Schema:

```
<xs:complexType name="Attribution">
  <xs:sequence>
    <xs:element name="entity" type="prov:IDRef"/>
    <xs:element name="agent" type="prov:IDRef"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>
```

The element `prov:wasAttributedTo` is used to reference a prov:Attribution from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="wasAttributedTo" type="prov:Attri
```

EXAMPLE 22: prov:wasAttributedTo

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:tr="http://example.com/ns/tr#"
    xmlns:rec54="http://example.com/ns/rec54#">

  <prov:agent prov:id="ex:Paolo">
    <prov:typexsi:type="xsd:QName">prov:Person</prov:type>
  </prov:agent>

  <prov:agent prov:id="ex:Simon">
    <prov:type xsi:type="xsd:QName">prov:Person</prov:type>
  </prov:agent>

  <prov:entity prov:id="tr:WD-prov-dm-20111215">
    <prov:type xsi:type="xsd:QName">rec54:WD</prov:type>
  </prov:entity>

  <prov:wasAttributedTo>
    <prov:entity prov:ref="rec54:WD"/>
    <prov:agent prov:ref="ex:Paolo"/>
    <prov:type xsi:type="xsd:QName">editorship</prov:type>
  </prov:wasAttributedTo>

  <prov:wasAttributedTo>
    <prov:entity prov:ref="rec54:WD"/>
    <prov:agent prov:ref="ex:Simon"/>
    <prov:type xsi:type="xsd:QName">authorship</prov:type>
  </prov:wasAttributedTo>

</prov:document>
```

3.3.3 Association

An activity association is an assignment of responsibility to an agent for an activity, indicating that the agent had a role in the activity. It further allows for a plan to be specified, which is the plan intended by the agent to achieve some goals in the context of this activity.

Type definition in XML Schema:

```
<xs:complexType name="Association">
  <xs:sequence>
    <xs:element name="activity" type="prov:IDRef"/>
    <xs:element name="agent" type="prov:IDRef" minOccurs="0"/>
    <xs:element name="plan" type="prov:IDRef" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>
```

The element `prov:wasAssociatedWith` is used to reference a prov:Association from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="wasAssociatedWith" type="prov:Ass
```

EXAMPLE 23: prov:wasAssociatedWith

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:activity prov:id="a">
    <prov:type xsi:type="xsd:string">workflow execution</prov:type>
  </prov:activity>

  <prov:agent prov:id="ag1">
    <prov:type xsi:type="xsd:QName">operator</prov:type>
  </prov:agent>

  <prov:agent prov:id="ag2">
    <prov:type xsi:type="xsd:QName">designator</prov:type>
  </prov:agent>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag1"/>
    <prov:role xsi:type="xsd:QName">loggedInUser</prov:role>
    <ex:how>webapp</ex:how>
  </prov:wasAssociatedWith>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag2"/>
    <prov:plan prov:ref="ex:wf"/>
    <prov:role xsi:type="xsd:QName">designer</prov:role>
    <ex:content>project1</ex:content>
  </prov:wasAssociatedWith>

  <prov:plan prov:id="ex:wf">
    <ex:label>Workflow 1</ex:label>
    <prov:location xsi:type="xsd:anyURI">http://example.org/workflow1.bpel</prov:location>
  </prov:plan>

</prov:document>
```

do you want to make it a QName like the other types?

prefix/namespace

3.3.3.1 Plan

A plan is an entity that represents a set of actions or steps intended by one or more agents to achieve some goals.

Type definition in XML Schema:

```
<xs:complexType name="Plan">
  <xs:complexContent>
    <xs:extension base="prov:Entity">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:plan` is used to reference a prov:Plan from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="plan" type="prov:Plan"/>
```

EXAMPLE 24: prov:plan

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:activity prov:id="a">
    <prov:type xsi:type="xsd:string">workflow execution</prov:type>
  </prov:activity>

  <prov:agent prov:id="ag1">
    <prov:type xsi:type="xsd:QName">operator</prov:type>
  </prov:agent>

  <prov:agent prov:id="ag2">
    <prov:type xsi:type="xsd:QName">designator</prov:type>
  </prov:agent>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag1"/>
    <prov:role xsi:type="xsd:QName">loggedInUser</prov:role>
    <ex:how>webapp</ex:how>
  </prov:wasAssociatedWith>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag2"/>
    <prov:plan prov:ref="ex:wf"/>
    <prov:role xsi:type="xsd:QName">designer</prov:role>
    <ex:content>project1</ex:content>
  </prov:wasAssociatedWith>

  <prov:plan prov:id="ex:wf">
    <ex:label>Workflow 1</ex:label>
    <prov:location xsi:type="xsd:anyURI">http://example.org/workflow1.bpel</prov:location>
  </prov:plan>

</prov:document>
```

make it a QName maybe?

3.3.4 Delegation

Delegation is the assignment of authority and responsibility to an agent (by itself or by another agent) to carry out a specific activity as a delegate or representative, while the agent it acts on behalf of retains some responsibility for the outcome of the delegated work.

Type definition in XML Schema:

```
<xs:complexType name="Delegation">
  <xs:sequence>
    <xs:element name="delegate" type="prov:IDRef"/>
    <xs:element name="responsible" type="prov:IDRef"/>
    <xs:element name="activity" type="prov:IDRef" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>
```

The element `prov:actedOnBehalfOf` is used to reference a prov:Delegation from within a

prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="actedOnBehalfOf" type="prov:Deleg
```

EXAMPLE 25: prov:actedOnBehalfOf

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#">

  <prov:activity prov:id="a">
    <prov:type xsi:type="xsd:QName">workflow</prov:type>
  </prov:activity>

  <prov:agent prov:id="ag1">
    <prov:type xsi:type="xsd:QName">programmer</prov:type>
  </prov:agent>

  <prov:agent prov:id="ag2">
    <prov:type xsi:type="xsd:QName">researcher</prov:type>
  </prov:agent>

  <prov:agent prov:id="ag3">
    <prov:type xsi:type="xsd:QName">funder</prov:type>
  </prov:agent>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag1"/>
    <prov:role xsi:type="xsd:QName">loggedInUser</prov:role>
  </prov:wasAssociatedWith>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag2"/>
  </prov:wasAssociatedWith>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag3"/>
  </prov:wasAssociatedWith>

  <prov:actedOnBehalfOf>
    <prov:delegate prov:ref="ag1"/>
    <prov:responsible prov:ref="ag2"/>
    <prov:activity prov:ref="a"/>
    <prov:type xsi:type="xsd:QName">line-management</prov:type>
  </prov:actedOnBehalfOf>

  <prov:actedOnBehalfOf>
    <prov:delegate prov:ref="ag2"/>
    <prov:responsible prov:ref="ag3"/>
    <prov:activity prov:ref="a"/>
    <prov:type xsi:type="xsd:QName">contract</prov:type>
  </prov:actedOnBehalfOf>

</prov:document>
```

3.3.5 Influence

Influence is the capacity of an entity, activity, or agent to have an effect on the character, development, or behavior of another by means of usage, start, end,

generation, invalidation, communication, derivation, attribution, association, or delegation.

Type definition in XML Schema:

```
<xs:complexType name="Influence">
  <xs:sequence>
    <xs:element name="influencee" type="prov:IDRef"/>
    <xs:element name="influencer" type="prov:IDRef"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>
```

The element `prov:wasInfluencedBy` is used to reference a prov:Influence from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="wasInfluencedBy" type="prov:Influ
```

EXAMPLE 26: prov:wasInfluencedBy

```
<prov:document
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:tr="http://example.com/ns/tr#"
    xmlns:w3="http://w3.org/">

  <prov:entity prov:id="tr:WD-prov-dm-20111215"/>

  <prov:agent prov:id="w3:Consortium"/>

  <prov:wasInfluencedBy>
    <prov:influencee prov:ref="tr:WD-prov-dm-20111215"/>
    <prov:influencer prov:ref="w3:Consortium"/>
  </prov:wasInfluencedBy>

</prov:document>
```

the correct URC is http://www.w3.org/Consortium.

3.4 Component 4: Bundles

The fourth component is concerned with bundles, a mechanism to support provenance of provenance.

3.4.1 Bundle

A bundle is a named set of provenance descriptions, and is itself an entity, so allowing provenance of provenance to be expressed.

do you mean type or element?

A `prov:Bundle` identifies a set of provenance descriptions, and is an extension of `prov:Entity`, so allowing provenance of provenance to be expressed by referencing the associated entity. The content of a bundle, i.e. its provenance records, can be represented by the `prov:BundleConstructor` complexType and can be specified with the `prov:bundleContent` element, its `prov:id` corresponds to the bundle entity.

Type definition in XML Schema:

```
<xs:complexType name="Bundle">
```

```
    <xs:complexContent>
      <xs:extension base="prov:Entity">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

The element `prov:bundle` is used to ~~reference~~ denote a prov:Bundle from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="bundle" type="prov:Bundle"/>
```

EXAMPLE 27: prov:bundle

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:bob="http://example.com/ns/bob#"
    xmlns:alice="http://example.com/ns/alice#"
    xmlns:agg="http://example.com/ns/agg#">

  <prov:bundle prov:id="ex:bundle1">
    <ex:version>1</ex:version>
  </prov:bundle>

  <prov:bundleContent prov:id="ex:bundle1">
    <prov:entity prov:id="ex:report1"/>

    <prov:entity prov:id="ex:report2">
      <prov:type xsi:type="xsd:QName">report</prov:type>
      <ex:version>2</ex:version>
    </prov:entity>

    <prov:wasGeneratedBy>
      <prov:entity prov:ref="ex:report2"/>
      <prov:time>2012-05-25T11:00:01</prov:time>
    </prov:wasGeneratedBy>

    <prov:wasDerivedFrom>
      <prov:generatedEntity prov:ref="ex:report2"/>
      <prov:usedEntity prov:ref="ex:report1"/>
    </prov:wasDerivedFrom>
  </prov:bundleContent>

</prov:document>
```

<ex: description> This is the bundle describing the provenance on ex:report2 </ex: description>.

bundles are not really versioned! Better to choose other "attribute"

3.4.2 Bundle Constructor

The `prov:BundleConstructor` complexType is used to define a named set of provenance statements. The Bundle Constructor type supports the `prov:id` attribute.

The bundle entity associated with a bundle constructor set must have the same `prov:id` as the bundle constructor set.

Type definition in XML Schema:

```
<xs:complexType name="BundleConstructor">
  <xs:sequence maxOccurs="unbounded">
    <-- references to standard non-PROV attribute PROV elements -->
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
```

```
</xs:complexType>
```

The element prov:bundleContent is used to reference a set of nested provenance statements from within a prov:Document.

Although prov:bundleContent can only be expressed at the prov:document level, the corresponding bundle entities may be specified at from either the prov:document or any prov:bundleContent, if at all.

```
<xs:complexType name="Document">
  <xs:sequence maxOccurs="unbounded">
    <!-- references to standard non-PROV attribute PROV elements -->
    <xs:element name="bundleContent" type="prov:BundleConstructor" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

EXAMPLE 28: prov:bundleContent

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:bob="http://example.com/ns/bob#"
    xmlns:alice="http://example.com/ns/alice#"
    xmlns:agg="http://example.com/ns/agg#">

  <prov:bundle prov:id="ex:bundle1">
    <ex:version>1</ex:version>
  </prov:bundle>

  <prov:bundleContent prov:id="ex:bundle1">
    <prov:entity prov:id="ex:report1"/>

    <prov:entity prov:id="ex:report2">
      <prov:type xsi:type="xsd:QName">report</prov:type>
      <ex:version>2</ex:version>
    </prov:entity>

    <prov:wasGeneratedBy>
      <prov:entity prov:ref="ex:report2"/>
      <prov:time>2012-05-25T11:00:01</prov:time>
    </prov:wasGeneratedBy>

    <prov:wasDerivedFrom>
      <prov:generatedEntity prov:ref="ex:report2"/>
      <prov:usedEntity prov:ref="ex:report1"/>
    </prov:wasDerivedFrom>
  </prov:bundleContent>

</prov:document>
```

3.5 Component 5: Alternate Entities

The fifth component of PROV-DM is concerned with relations SpecializationOf (Specialization) and AlternateOf (Alternate) between entities.

3.5.1 Specialization

An entity that is a specialization of another shares all aspects of the latter, and additionally presents more specific aspects of the same thing as the latter. In particular, the lifetime of the entity being specialized contains that of any

specialization.

Type definition in XML Schema:

```
<xs:complexType name="Specialization">
  <xs:sequence>
    <xs:element name="specificEntity" type="prov:IDRef"/>
    <xs:element name="generalEntity" type="prov:IDRef"/>
  </xs:sequence>
</xs:complexType>
```

denote.

The element `prov:specializationOf` is used to reference a prov:Specialization from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="specializationOf" type="prov:Spec
```

EXAMPLE 29: prov:specializationOf

```
<prov:document
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:bbc="http://www.bbc.co.uk/">

  <prov:entity prov:id="ex:bbcNews2012-03-23"/>

  <prov:entity prov:id="bbc:news"/>

  <prov:specializationOf>
    <prov:specificEntity prov:ref="ex:bbcNews2012-03-23"/>
    <prov:generalEntity prov:ref="bbc:news"/>
  </prov:specializationOf>

</prov:document>
```

3.5.2 Alternate

Two alternate entities present aspects of the same thing. These aspects may be the same or different, and the alternate entities may or may not overlap in time.

Type definition in XML Schema:

```
<xs:complexType name="Alternate">
  <xs:sequence>
    <xs:element name="alternate1" type="prov:IDRef"/>
    <xs:element name="alternate2" type="prov:IDRef"/>
  </xs:sequence>
</xs:complexType>
```

The element `prov:alternateOf` is used to reference a prov:Alternate from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="alternateOf" type="prov:Alternate
```

EXAMPLE 30: prov:alternateOf

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:bbc="http://www.bbc.co.uk/news"
    xmlns:bbcmobile="http://www.bbc.co.uk/news/mobile">

  <prov:entity prov:id="bbc:science-environment-17526723">
    <prov:type xsi:type="xsd:string">a news item for desktop</prov:type>
  </prov:entity>

  <prov:entity prov:id="bbcmobile:science-environment-17526723">
    <prov:type xsi:type="xsd:string">a news item for mobile devices</prov:type>
  </prov:entity>

  <prov:alternateOf>
    <prov:alternate1 prov:ref="bbcmobile:science-environment-17526723"/>
    <prov:alternate2 prov:ref="bbc:science-environment-17526723"/>
  </prov:alternateOf>

</prov:document>
```

3.6 Component 6: Collections

The sixth component of PROV-DM is concerned with the notion of collections. A collection is an entity that has some members. The members are themselves entities, and therefore their provenance can be expressed. Some applications need to be able to express the provenance of the collection itself: e.g. who maintains the collection (attribution), which members it contains as it evolves, and how it was assembled. The purpose of Component 6 is to define the types and relations that are useful to express the provenance of collections.

3.6.1 Collection

A collection is an entity that provides a structure to some constituents that must themselves be entities. These constituents are said to be member of the collections.

Type definition in XML Schema:

```
<xs:complexType name="Collection">
  <xs:complexContent>
    <xs:extension base="prov:Entity">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:collection` is used to reference a prov:Collection from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="collection" type="prov:Collectior
```

EXAMPLE 31: prov:collection

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:collection prov:id="c1" />

</prov:document>
```

prefix (namespace.

3.6.1.1 Empty Collection

An empty collection is a collection without members.

Type definition in XML Schema:

```
<xs:complexType name="EmptyCollection">
  <xs:complexContent>
    <xs:extension base="prov:Collection">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The element `prov:emptyCollection` is used to reference a prov:EmptyCollection from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="emptyCollection" type="prov:Empty
```

EXAMPLE 32: prov:emptyCollection

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:bbc="http://www.bbc.co.uk/news/"
    xmlns:bbcmobile="http://www.bbc.co.uk/news/mobile">

    <prov:emptyCollection prov:id="c0"/>

</prov:document>
```

3.6.2 Membership

Membership is the belonging of an entity to a collection.

Type definition in XML Schema:

```
<xs:complexType name="Membership">
  <xs:sequence>
    <xs:element name="collection" type="prov:IDRef"/>
    <xs:element name="entity" type="prov:IDRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The element `prov:hadMember` is used to reference a prov:Membership from within a prov:Document or prov:BundleConstructor.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="hadMember" type="prov:Membership'
```

EXAMPLE 33: prov:hadMember

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#">

  <prov:entity prov:id="e0"/>
  <prov:entity prov:id="e1"/>
  <prov:entity prov:id="e2"/>

  <prov:collection prov:id="c" />

  <prov:hadMember>
    <prov:collection prov:ref="c"/>
    <prov:entity prov:ref="e0"/>
    <prov:entity prov:ref="e1"/>
    <prov:entity prov:ref="e2"/>
  </prov:hadMember>

</prov:document>
```

may be use other word here.

3.7 Further Elements of PROV

This section introduces further elements of PROV.

3.7.1 Identifier

The identifier attribute is used to identify instances of PROV types or relations.

```
<xs:attribute xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" type="xs:QName"/>
```

EXAMPLE 34: prov:id

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:tr="http://example.com/ns/tr#">

  <prov:entity prov:id="tr:WD-prov-dm-20111215">
    <prov:type xsi:type="xsd:Qname">document</prov:type>
    <ex:version>2</ex:version>
  </prov:entity>

</prov:document>
```

→ identifiers are QNames.

→ QNames map to URIs (as per DM)

→ Note restriction

3.7.2 Reference

A reference-by-id to a PROV entity, activity, agent, or relation.

```
<xs:attribute ref="prov:ref" use="required" />
```

EXAMPLE 35: prov:ref

```
<prov:document
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:entity prov:id="e1"/>

  <prov:activity prov:id="a1"/>

  <prov:wasGeneratedBy>
    <prov:entity prov:ref="e1"/>
    <prov:activity prov:ref="a1"/>
    <prov:time>2001-10-26T21:32:52</prov:time>
    <ex:port>p1</ex:port>
  </prov:wasGeneratedBy>

</prov:document>
```

- ref are QNames.
- show schema of prov:ref.
- Note restriction.

3.7.3 Attributes

The PROV-DM defined PROV attributes are represented in XML as elements.

3.7.3.1 Label

The attribute prov:label provides a human-readable representation of an instance of a PROV-DM type or relation.

The element `prov:label` is used to represent a PROV label attribute and has type prov:InternationalizedString.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="label" type="prov:Internationaliz
```

show schema.

EXAMPLE 36: prov:label

```
<prov:document
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:entity prov:id="ex:e1">
    <prov:label>This is a human-readable label</prov:label>
  </prov:entity>

  <prov:entity prov:id="ex:car01">
    <prov:label xml:lang="fr">Voiture 01</prov:label>
    <prov:label xml:lang="en">Car 01</prov:label>
  </prov:entity>

</prov:document>
```

3.7.3.2 Location

A location can be an identifiable geographic place (ISO 19112), but it can also be a non-geographic place such as a directory, row, or column.

The element `prov:location` is used to represent a PROV location attribute and has type xs:anySimpleType.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="location" type="xs:anySimpleType'
```

EXAMPLE 37: prov:location

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:entity prov:id="ex:MonaLisa">
    <prov:type xsi:type="xsd:QName">StillImage</prov:type>
    <prov:location xsi:type="xsd:string">Le Louvre, Paris</prov:location>
  </prov:entity>

  <prov:entity prov:id="ex:cell">
    <prov:location xsi:type="xsd:string">(5,5)</prov:location>
    <prov:value xsi:type="xsd:integer">10</prov:value>
  </prov:entity>

</prov:document>
```

3.7.3.3 Role

A role is the function of an entity or agent with respect to an activity, in the context of a usage, generation, invalidation, association, start, and end.

The element `prov:role` is used to represent a PROV Role attribute and has type xs:anySimpleType.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="role" type="xs:anySimpleType"/>
```

EXAMPLE 38: prov:role

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag1"/>
    <prov:role xsi:type="xsd:QName">loggedInUser</prov:role>
    <ex:how>webapp</ex:how>
  </prov:wasAssociatedWith>

  <prov:wasAssociatedWith>
    <prov:activity prov:ref="a"/>
    <prov:agent prov:ref="ag2"/>
    <prov:plan prov:ref="ex:wf"/>
    <prov:role xsi:type="xsd:QName">designer</prov:role>
    <ex:content>project1</ex:content>
  </prov:wasAssociatedWith>

</prov:document>
```

3.7.3.4 Type

The attribute prov:type provides further typing information for any construct with an optional set of attribute-value pairs.

The XML element `prov:type` is used to represent a PROV Type attribute and has XML type xs:anySimpleType.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="type" type="xs:anySimpleType"/>
```

EXAMPLE 39: prov:type

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#"
    xmlns:tr="http://example.com/ns/tr#">

  <prov:entity prov:id="tr:WD-prov-dm-20111215">
    <prov:type xsi:type="xsd:QName">document</prov:type>
    <ex:version>2</ex:version>
  </prov:entity>

  <prov:agent prov:id="e1">
    <prov:type xsi:type="xsd:QName">prov:Person</prov:type>
    <ex:name>Alice</ex:name>
    <ex:employee>1234</ex:employee>
  </prov:agent>

  <prov:activity prov:id="a1">
    <prov:startTime>2011-11-16T16:05:00</prov:startTime>
    <prov:endTime>2011-11-16T16:06:00</prov:endTime>
    <prov:type xsi:type="xsd:QName">ex:edit</prov:type>
    <ex:host>server.example.org</ex:host>
  </prov:activity>

</prov:document>
```

prefix / namespace.

3.7.3.5 Value

The attribute prov:value provides a value that is a direct representation of an entity as a PROV-DM Value.

The XML element `prov:value` is used to represent a PROV Value attribute and has XML type xs:anySimpleType.

Element definition in XML Schema:

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="value" type="xs:anySimpleType"/>
```

EXAMPLE 40: prov:value

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">
```

```
    <prov:entity prov:id="ex:in">
      <prov:value xsi:type="xsd:string">abcd</prov:value>
    </prov:entity>

    <prov:entity prov:id="ex:out">
      <prov:value xsi:type="xsd:integer">4</prov:value>
    </prov:entity>

  </prov:document>
```

3.7.4 Value

A value is a constant such as a string, number, time, qualified name, IRI, and encoded binary data, whose interpretation is outside the scope of PROV.

Relations defined by the PROV-DM to have type Value have type `xs:anySimpleType` in PROV-XML unless otherwise specified.

3.8 Structural Elements of PROV-XML

3.8.1 Document

The root element of all PROV-XML documents is `prov:document` which has type `prov:Document`.

Similar to a `prov:BundleConstructor`, the `prov:Document` complexType is used to define a set of provenance statements.

Unlike the `prov:BundleConstructor`, a `prov:Document`

- may contain `prov:bundle` elements (but not other `prov:document` elements)
- does not support the prov:id attribute.

Type definition in XML Schema:

```
<xs:complexType name="Document">
  <xs:sequence maxOccurs="unbounded">
    <!-- references to standard non-PROV attribute PROV elements -->
    <xs:element name="bundleContent" type="prov:BundleConstructor" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The element `prov:document` may only be used as the root element of a a PROV-XML document.

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="document" type="prov:Document"/>
```

EXAMPLE 41: prov:document

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <!-- prov statements go here -->

</prov:document>
```

3.8.2 Other

The `prov:other` element provides a place to include non-PROV XML elements inside a `prov:document` or `prov:bundleContent`.

Type definition in XML Schema:

```
<xs:complexType name="Other">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The element `prov:other` may be used in a `prov:document` or a `prov:bundleContent` but may not be used inside a relation, entity, or activity element.

```
<xs:element name="other" type="prov:Other"/>
```

EXAMPLE 42: prov:other

```
<prov:document
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:ex="http://example.com/ns/ex#">

  <!-- prov statements go here -->

  <prov:other>
    <ex:foo>
      <ex:content>bar</ex:content>
    </ex:foo>
  </prov:other>

  <!-- more prov statements can go here -->

</prov:document>
```

4. Media Type

Contact:
Ivan Herman
See also:
How to Register a Media Type for a W3C Specification
Internet Media Type registration, consistency of use
TAG Finding 3 June 2002 (Revised 4 September 2002)

The Internet Media Type / MIME type for PROV-XML is "`application/provenance+xml`".

It is recommended that PROV-XML files have the extension "`.provx`" (all lowercase) on all platforms.

It is recommended that PROV-XML files stored on Macintosh HFS file systems be given a file type of `TEXT`.

The information that follows has been submitted to the IESG for review, approval, and registration with IANA.

Type name:
application

Subtype name:
provenance+xml
Required parameters:
none
Optional parameters:
Same as charset parameter of application/xml as specified in RFC3023 (Section 3.2).
Encoding considerations:
Same as encoding considerations of application/xml as specified in RFC 3023 (Section 3.2).
Security considerations:
PROV-XML is an XML language for describing the provenance of things; applications may evaluate given data to dereference URIs, invoking the security considerations of the scheme for that URI. Note in particular, the privacy issues in [RFC3023] section 10 for HTTP URIs. Data obtained from an inaccurate or malicious data source may lead to inaccurate or misleading conclusions, as well as the dereferencing of unintended URIs. Care must be taken to align the trust in consulted resources with the sensitivity of the intended use of the data.
PROV-XML can express data which is presented to the user, for example, by means of label attributes. Application rendering strings retrieved from untrusted PROV-N documents must ensure that malignant strings may not be used to mislead the reader. The security considerations in the media type registration for XML ([RFC3023] section 10) provide additional guidance around the expression of arbitrary data and markup.
PROV-XML is a language for describing the provenance of things, and therefore a PROV-XML document is metadata for other resources. Untrusted PROV-XML documents may mislead its consumers by indicating that a third-party resource has a reputable lineage, when it has not. Provenance of PROV-XML document should be sought.
PROV-XML uses QNames mappable to IRIs as term identifiers. Applications interpreting data expressed in PROV-XML should address the security issues of Internationalized Resource Identifiers (IRIs) [RFC3987] Section 8, as well as Uniform Resource Identifier (URI): Generic Syntax [RFC3986] Section 7.
Multiple IRIs may have the same appearance. Characters in different scripts may look similar (a Cyrillic "o" may appear similar to a Latin "o"). A character followed by combining characters may have the same visual representation as another character (LATIN SMALL LETTER E followed by COMBINING ACUTE ACCENT has the same visual representation as LATIN SMALL LETTER E WITH ACUTE). Any person or application that is writing or interpreting data in PROV-N must take care to use the IRI that matches the intended semantics, and avoid IRIs that make look similar. Further information about matching of similar characters can be found in Unicode Security Considerations [UNISEC] and Internationalized Resource Identifiers (IRIs) [RFC3987] Section 8.
Interoperability considerations:
There are no known interoperability issues.
Published specification:
PROV-XML: The PROV XML Schema, Hua, Tilmes, Zednik (eds), Moreau http://www.w3.org/TR/prov-xml/, 2013.
Applications which use this media type:
It may be used by any application for publishing provenance information. This format is designed to be an XML form of provenance.
Fragment identifier considerations:
N/A
Additional Information:
Magic number(s):

PROV-XML documents are XML documents and thus may have initial strings similar to any XML document.

File extension(s):

.provx

Macintosh file type code(s):

"TEXT"

Person & email address to contact for further information:

Ivan Herman, ivan@w3.org

Intended usage:

COMMON

Restrictions on usage:

none

Author:

The PROV-XML specification is the product of the World Wide Web Consortium's Provenance Working Group.

Change controller:

The W3C, and the W3C Provenance Working Group, have change control over this specification.

A. XML Schema

A.1 prov.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.w3.org/ns/prov#"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:prov="http://www.w3.org/ns/prov#"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">

  <xs:include schemaLocation="prov-core.xsd"/>
  <xs:include schemaLocation="prov-dictionary.xsd"/>
  <xs:include schemaLocation="prov-links.xsd"/>

</xs:schema>
```

A.2 prov-core.xsd

```
<?xml version="1.0" encoding="utf-8"?>

<!--
  In PROV-DM, all ids are qualified names, specified as prov:QualifiedName in PROV-N.
  In this schema, all ids are instead defined as xsd:QNames.
  -->


<xs:schema targetNamespace="http://www.w3.org/ns/prov#"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:prov="http://www.w3.org/ns/prov#"
           xmlns:cu="http://www.w3.org/1999/xhtml/datatypes/"
           xmlns:xml="http://www.w3.org/XML/1998/namespace"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">



  <xs:import namespace="http://www.w3.org/1999/xhtml/datatypes/" />
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
             schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!-- Component 1 -->
```

```
<xs:complexType name="Entity">
  <xs:sequence>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:value" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Activity">
  <xs:sequence>
    <xs:element name="startTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="endTime" type="xs:dateTime" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Generation">
  <xs:sequence>
    <xs:element name="entity" type="prov:IDRef"/>
    <xs:element name="activity" type="prov:IDRef" minOccurs="0"/>
    <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Usage">
  <xs:sequence>
    <xs:element name="activity" type="prov:IDRef"/>
    <xs:element name="entity" type="prov:IDRef" minOccurs="0"/>
    <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Communication">
  <xs:sequence>
    <xs:element name="informed" type="prov:IDRef"/>
    <xs:element name="informant" type="prov:IDRef"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Start">
  <xs:sequence>
    <xs:element name="activity" type="prov:IDRef"/>
```

```
      <xs:element name="trigger" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="starter" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <xs:complexType name="End">
    <xs:sequence>
      <xs:element name="activity" type="prov:IDRef"/>
      <xs:element name="trigger" type="prov:IDRef"  minOccurs="0"/>
      <xs:element name="ender" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <xs:complexType name="Invalidation">
    <xs:sequence>
      <xs:element name="entity" type="prov:IDRef"/>
      <xs:element name="activity" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <!-- Component 2 -->

  <xs:complexType name="Derivation">
    <xs:sequence>
      <xs:element name="generatedEntity" type="prov:IDRef"/>
      <xs:element name="usedEntity" type="prov:IDRef"/>
      <xs:element name="activity" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="generation" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="usage" type="prov:IDRef" minOccurs="0"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <xs:complexType name="Revision">
    <xs:complexContent>
      <xs:extension base="prov:Derivation">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="Quotation">
    <xs:complexContent>
```

```
      <xs:extension base="prov:Derivation">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrimarySource">
    <xs:complexContent>
      <xs:extension base="prov:Derivation">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <!-- Component 3 -->

  <xs:complexType name="Agent">
    <xs:sequence>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:location" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <xs:complexType name="Person">
    <xs:complexContent>
      <xs:extension base="prov:Agent">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="Organization">
    <xs:complexContent>
      <xs:extension base="prov:Agent">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SoftwareAgent">
    <xs:complexContent>
      <xs:extension base="prov:Agent">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="Attribution">
    <xs:sequence>
      <xs:element name="entity" type="prov:IDRef"/>
      <xs:element name="agent" type="prov:IDRef"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
  </xs:complexType>

  <xs:complexType name="Association">
    <xs:sequence>
      <xs:element name="activity" type="prov:IDRef"/>
      <xs:element name="agent" type="prov:IDRef" minOccurs="0"/>
      <xs:element name="plan" type="prov:IDRef" minOccurs="0"/>
      <!-- prov attributes -->
      <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:role" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="prov:id"/>
```

```
</xs:complexType>

<xs:complexType name="Delegation">
  <xs:sequence>
    <xs:element name="delegate" type="prov:IDRef"/>
    <xs:element name="responsible" type="prov:IDRef"/>
    <xs:element name="activity" type="prov:IDRef" minOccurs="0"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<xs:complexType name="Influence">
  <xs:sequence>
    <xs:element name="influencee" type="prov:IDRef"/>
    <xs:element name="influencer" type="prov:IDRef"/>
    <!-- prov attributes -->
    <xs:element ref="prov:label" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="prov:type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="prov:id"/>
</xs:complexType>

<!-- Component 4 -->

<xs:complexType name="Bundle">
  <xs:complexContent>
    <xs:extension base="prov:Entity">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- bundle container and allowable PROV elements -->

<xs:complexType name="BundleConstructor">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="prov:entity" minOccurs="0"/>
    <xs:element ref="prov:activity" minOccurs="0"/>
    <xs:element ref="prov:wasGeneratedBy" minOccurs="0"/>
    <xs:element ref="prov:used" minOccurs="0"/>
    <xs:element ref="prov:wasInformedBy" minOccurs="0"/>
    <xs:element ref="prov:wasStartedBy" minOccurs="0"/>
    <xs:element ref="prov:wasEndedBy" minOccurs="0"/>
    <xs:element ref="prov:wasInvalidatedBy" minOccurs="0"/>
    <xs:element ref="prov:wasDerivedFrom" minOccurs="0"/>
    <xs:element ref="prov:wasRevisionOf" minOccurs="0"/>
    <xs:element ref="prov:wasQuotedFrom" minOccurs="0"/>
    <xs:element ref="prov:hadPrimarySource" minOccurs="0"/>
    <xs:element ref="prov:agent" minOccurs="0"/>
    <xs:element ref="prov:person" minOccurs="0"/>
    <xs:element ref="prov:organization" minOccurs="0"/>
    <xs:element ref="prov:softwareAgent" minOccurs="0"/>
    <xs:element ref="prov:wasAttributedTo" minOccurs="0"/>
    <xs:element ref="prov:wasAssociatedWith" minOccurs="0"/>
    <xs:element ref="prov:actedOnBehalfOf" minOccurs="0"/>
    <xs:element ref="prov:wasInfluencedBy" minOccurs="0"/>
    <xs:element ref="prov:bundle" minOccurs="0"/>
    <xs:element ref="prov:specializationOf" minOccurs="0"/>
    <xs:element ref="prov:alternateOf" minOccurs="0"/>
    <xs:element ref="prov:collection" minOccurs="0"/>
    <xs:element ref="prov:emptyCollection" minOccurs="0"/>
    <xs:element ref="prov:hadMember" minOccurs="0"/>
    <xs:element ref="prov:plan" minOccurs="0"/>
    <xs:element ref="prov:other" minOccurs="0"/>
    <xs:element ref="prov:internalElement" minOccurs="0"/>
  </xs:sequence>
```

why sequence? choice.