

Proposal: A Flat XSD Rendering with Global elements, global <Entities> element bag and <abstract> elements/substitution groups (Modified after the Teragrid GLUE XSD)

The main modifications include:

- 1) Pro: All main elements are made global so that 3rd party XSD can import this schema and re-use those elements standalone. Con: Multiple global elements so the intended document root element needs to be clearly documented (this is common, is this as a real con?).
- 2) The modified XSD includes abstract elements with corresponding concrete element implementations. For example:
 - abstract <Domain> and concrete <AdminDomain> and <UserDomain>,
 - abstract <AbstractService> and concrete <Service>, <ComputingService>, <StorageService>.
- 3) The <Entities> element references abstract elements.
 - Pro: In doing this, new element specialisations that define the appropriate substitution group can be nested within the 'Entities' element in any future/extending profile (requires no future modification of the glue XSD).
 - Pro: Proposal can be applied to both the Teragrid XSD and the current XSD: next few slides show modified Teragrid XSD simply because this schema was already 'flat' (a few more modifications needed for the other XSD to flatten, but not hard to do).

Full schema and sample doc:

http://tools.ngs.ac.uk/ngstools/glue2proposal/modifiedTeraGridXSD_Sample.zip

XSD Fragment (modified from Teragrid)

```
<!-- Entities is still the DOCUMENT ROOT ELEMENT -->
<element name="Entities" type="glue:ExtensibleEntities_t"/>
<complexType name="ExtensibleEntities_t">
  <sequence>
    <!-- Abstract element references:
    Abstract elements allow sub-type specialisations. New specialisations that
    define the appropriate substitution group can be nested within this
    'Entities' element (requires no modification to this XSD). -->
    <element ref="glue:Domain" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:AbstractService" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:AbstractEndpoint" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Share" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Manager" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Resource" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:AbstractActivity" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Policy" minOccurs="0" maxOccurs="unbounded"/>
    <!-- Concrete element references:
    TODO: These elements do not have a parent abstract type. Therefore, we could
    reference them directly in the schema if we want to specify an order,
    or maybe define a new 'OtherEntities' element that can nest any element in
    the target namespace in any order (see 8 lines below) -->
    <element ref="glue:Location" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Contact" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:Benchmark" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:ApplicationEnvironment" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:ApplicationHandle" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:StorageServiceCapacity" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="glue:StorageShareCapacity" minOccurs="0" maxOccurs="unbounded"/>
    <!-- List concrete elements as above Or define 'OtherEntities' -->
    <element name="OtherEntities">
      <complexType>
        <sequence>
          <any namespace="##targetNamespace" processContents="strict" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>

<element name="Location" type="glue:Location_t" />
<element name="Contact" type="glue:Contact_t" />
<element name="Domain" type="glue:Domain_t" abstract="true"/>
<element name="AdminDomain" type="glue:AdminDomain_t" substitutionGroup="glue:Domain"/>
<element name="UserDomain" type="glue:UserDomain_t" substitutionGroup="glue:Domain"/>
```

<Entities> is Document Root element
(modified from Teragrid example)

<Entities> is an element
bag that references other
global elements (both
abstract and concrete).

The main entities are also global so that they can be
referenced from within <Entities>, and also be
imported/used standalone in 3rd party XSD

Sample XML Instance Doc (most elements are collapsed)

```
<?xml version="1.0" encoding="UTF-8"?>
<glue:Entities
  xmlns:xsi='http://www.w3.org/2001/XMLSchema
  xmlns:glue='http://info.teragrid.org/glue/
  xsi:schemaLocation='http://info.teragrid.o

  <glue:AdminDomain BaseType="Domain">
    <ID>urn://some.uniqueID</ID>
    <Description>hello world</Description>
    <WWW>http://some.url</WWW>
  </glue:AdminDomain>
  <glue:UserDomain>
  <glue:Service>
  <glue:ComputingService>
  <glue:StorageService>
  <glue:Endpoint>
  <glue:ComputingEndpoint>
  <glue:StorageEndpoint>
  <glue:ComputingShare>
  <glue:StorageShare>
  <glue:ComputingManager>
  <glue:StorageManager>
  <glue:DataStore>
  <glue:ExecutionEnvironment>
  <glue:Activity>
  <glue:ComputingActivity>
  <glue:AccessPolicy>
  <glue:MappingPolicy>
  <OtherEntities>
    <glue:MappingPolicy>
    <glue:Benchmark>
  </OtherEntities>
</glue:Entities>
```

<Entities> is Document Root element
(an ordered element bag)

Abstract <Domain> implementations

<AbstractService> impls

<AbstractEndpoint> impls

<Share> impls

<Manager> impls

<Resource> impls

<Activity> impls

<Activity> impls

List other entities that do not have abstract parents