XSD Design Guidelines

The Ed-Fi Core XML Schema is the embodiment of the Ed-Fi Unifying Data Model (UDM) in an XML schema format that is designed to meet the requirements for XML data interchange, and has the following attributes:

- A core set of domain, association, and attribute types that directly map to the Ed-Fi UDM
- A method and examples of composing interchange schemas, reusing the types defined in the core schema
- A method and examples for extending the core schema and interchange schema to account for implementation-specific, or even interchange-specific, data (discussed in the Ed-Fi Extension Framework section of this documentation)

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

Ed-Fi Core XML Schema Expresses the Ed-Fi UDM

The Ed-Fi Core XML Schema expresses the Ed-Fi UDM's entities, associations, and attributes for education data. The Ed-Fi UDM is organized into 16 domains. These domains are organized into 64 different entities and 23 associations, and supported by 58 Descriptors — all of which have representative XML schema complex types.

Principles

XML schemas enforce rules around the content of XML-instance documents. However, there are many different ways and styles for XML schemas to accomplish the same goal.[1]

The Ed-Fi Core XML Schema is designed with the following principles:

- **Consistency.**The XML schema should have a consistent organization and design pattern.
- **Extensibility.**The XML schema should be designed to easily allow for customizations to meet specific interchange requirements.
- **Flexibility.**The XML schema should be flexible to support the interchange of different subsets and collections of education data.
- **Reuse.**The XML schema should be designed to facilitate the reuse of types of elements in constructing the interchange schemas used for data transfer.
- **Composition Using XML Mechanisms.**The creation of interchange schemas from core and extended schemas should be accomplished using native XML mechanisms, not through a cut-and-paste operation.

Schema Design Pattern

The Ed-Fi Core XML Schema uses the Venetian Blind design pattern to facilitate reuse while also hiding namespace complexities. For each interchange schema, the style defines a single global element that nests local elements that use types (simple or complex) that are defined within the global namespace.

For example, consider the data exchange schema for parents and their student relationship depicted below:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.
org/0210" targetNamespace="http://ed-fi.org/0210" elementFormDefault="
qualified" attributeFormDefault="unqualified">
        <xs:include schemaLocation="Ed-Fi-Core.xsd"/>
        <xs:annotation>
                <xs:documentation>===== Parent Interchange Model =====</xs:
documentation>
        </xs:annotation>
        <xs:element name="InterchangeParent">
                <xs:complexType>
                        <xs:choice maxOccurs="unbounded">
                                <xs:element name="Parent" type="Parent"/>
                                <xs:element name="
StudentParentAssociation" type="StudentParentAssociation"/>
                        </xs:choice>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

A single element InterchangeParent defines the interchange format. The two elements of the interchange, Parent and StudentParentAssociation, are nested within the single element, referencing types in Ed-Fi-Core.xsd (see the `include` statement above).

To provide maximum flexibility for the interchange, we chose to encapsulate the elements of the interchange in an unbounded `choice` statement rather than in an XML `sequence`. As a result, the various elements of the interchange are optional and can be provided in any order.

Version Identification

Versions of the Ed-Fi Core XML Schema are identified by major and minor version number, such as "3.0". A version may be appended with RFC ("Request for Comment"), RC ("Release Candidate"), or DRAFT. Versions are identified by the following:

- A unique default namespace
- A unique location for the file on the web

Namespace

The Ed-Fi Core XML Schema identifies the target namespace `http://ed-fi.org/nnnn` (with an appending version) as the default namespace. This is done so references to complex types do not have to qualify each reference, hence improving readability.

The implication is that any interchange (or customized core schema) that references the Ed-Fi-Core.xsd would need to use the same namespace, as shown here:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.
org/0210" targetNamespace="http://ed-fi.org/0210" elementFormDefault="
qualified" attributeFormDefault="unqualified">
     <xs:include schemaLocation="Ed-Fi-Core.xsd"/>
     . . .
</xs:schema>
```

Ed-Fi Core XML Schema Organization

The primary Ed-Fi Core XML Schema types listed below are the building blocks of interchange schemas. The domain entities and associations match exactly those in the Ed-Fi UDM.

Table 1.Ed-Fi Core Schema Primary Types

Name	Definition
Domain entities	XML complex types representing the major entities in the education domain, as modeled in the Ed-Fi UDM.
Descriptors	XML complex types representing enumerations that cannot be standardized, and are determined by the content of their application and require loading in an interchange.
Associations	XML complex types representing those associations between the domain and Descriptor entities that require attributes.

Domain entities, Descriptors, and association types are composed from a second tier of XML simple and complex types and are organized as follows:

Table 2.Ed-Fi Supporting Types

Name	Definition
Base types	Template structures for entities, Descriptors, reference types, and Descriptor reference types.
Extended reference types	Supporting associations via XML `IDREF`s or via a lookup. Extended reference types provide a sort of "off page connector" for interchange schemas, pointing to entities that are already known by the receiving system.
Extended Descriptor reference types	Referencing the Descriptor values for an attribute.
Common types	Complex types composed of cohesive records of attributes, such as Name, Address, or Telephone.

Enumeration type	Standard code lists or controlled vocabulary that are used for attributes or are used to map Descriptor values.
String simple types and numeric simple types	Strings, integers, and doubles with specific constraints, such as length or value range, for the various simple typed attributes. While strings can be restricted in-line, this approach creates an anonymous type that can cause trouble when extending the parent complex type. Therefore, all restricted types are named.

Domain Entities

All domain entities are XML extensions of the type ComplexObjectType. This type defines the common ID attribute that holds the assignment of an XML `IDREF` to instances of the domain type used in interchange schemas, as shown below.

```
<xs:complexType name="ComplexObjectType" abstract="true">
        <xs:annotation>
        . . .
        <xs:attribute name="id" type="xs:ID">
        . . .
</xs:complexType>
```

The complex type for the domain entity corresponds to the class defined in the Ed-Fi UDM. The elements of the complex type correspond to the attributes of the UDM class, as shown in the following example for the CourseOffering entity:

```
<xs:complexType name="CourseOffering">
        <xs:complexContent>
                <xs:extension base="ComplexObjectType">
                    <xs:sequence>
                        <xs:element name="LocalCourseCode" type="
LocalCourseCode">
                                        . . .
                        <xs:element name="LocalCourseTitle" type="
CourseTitle" minOccurs="0">
                                        . . .
                        <xs:element name="InstructionalTimePlanned" type="
Duration" minOccurs="0">
                                        . . .
                        <xs:element name="CurriculumUsed" type="
CurriculumUsedType" minOccurs="0" maxOccurs="unbounded">
                                        . . .
                        <xs:element name="SchoolReference" type="
SchoolReferenceType">
                                        . . .
                        <xs:element name="SessionReference" type="
SessionReferenceType">
                                        . . .
                        <xs:element name="CourseReference" type="
CourseReferenceType">
                                        . . .
                    </xs:sequence>
                </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

XML Attributes

Following the tenet that XML attributes contain data that is only used in the processing of the XML, the ComplexObjectType and ReferenceType base types provide the XML-internal Identifiers, as shown below.

```
<xs:complexType name="ComplexObjectType" abstract="true">
     . . .
     <xs:attribute name="id" type="xs:ID">
     . . .
</xs:complexType>

<xs:complexType name="ReferenceType">
     . . .
     <xs:attribute name="id" type="xs:ID">
     . . .
     <xs:attribute name="ref" type="xs:IDREF">
     . . .
</xs:complexType>
```

Generalizations

Generalizations are specified using the XML schema extension. For example, EducationOrganization is a
generalization of School, LocalEducationAgency, EducationServiceCenter,
EducationOrganizationNetwork, and StateEducationAgency as shown below. Conversely, it can also
be said that School is a specialization of EducationOrganization.

The complex type EducationOrganization is identified as an abstract type denoting that it is only defined
as a generalization and may not be used on its own.

```
<xs:complexType name="EducationOrganization" abstract="true">
     <xs:complexContent>
         <xs:extension base="ComplexObjectType">
             <xs:sequence>
                                        <xs:element name="
StateOrganizationId" type="IdentificationCode">
                                . . .
                                        <xs:element name="
EducationOrganizationIdentificationCode" type="
EducationOrganizationIdentificationCode" minOccurs="0" maxOccurs="
unbounded">
                                . . .
                             <xs:element name="NameOfInstitution" type="
NameOfInstitution">
                                . . .
                                        <xs:element name="
ShortNameOfInstitution" type="NameOfInstitution" minOccurs="0">
                                . . .
                             <xs:element name="EducationOrganizationCategory"
type="EducationOrganizationCategoryType" maxOccurs="unbounded">
                                . . .
                             <xs:element name="Address" type="Address"
maxOccurs="unbounded">
                                . . .
                                        <xs:element name="
InternationalAddress" type="InternationalAddress" minOccurs="0" maxOccurs="
unbounded">
                                . . .
                                        <xs:element name="
InstitutionTelephone" type="InstitutionTelephone" minOccurs="0" maxOccurs="
unbounded">
                                . . .
                                        <xs:element name="WebSite" type="
URI" minOccurs="0">
                                . . .
                                        <xs:element name="
OperationalStatus" type="OperationalStatusType" minOccurs="0">
                                . . .
             </xs:sequence>
         </xs:extension>
     </xs:complexContent>
</xs:complexType>
```

Specializations are reflected as extensions of the base type, specifying the additional elements particular to School, as shown below. Note that School inherits all of the attributes of EducationOrganization.

```
<xs:complexType name="School">
     <xs:complexContent>
         <xs:extension base="EducationOrganization">
              <xs:sequence>
                    <xs:element name="SchoolId" type="xs:int">
                          . . .
                    <xs:element name="GradeLevel" type="
GradeLevelDescriptorReferenceType" maxOccurs="unbounded">
                          . . .
                    <xs:element name="SchoolCategory" type="
SchoolCategoryType" minOccurs="0" maxOccurs="unbounded">
                          . . .
                    <xs:element name="SchoolType" type="SchoolType"
minOccurs="0">
                          . . .
                    <xs:element name="CharterStatus" type="
CharterStatusType" minOccurs="0">
                          . . .
                    <xs:element name="TitleIPartASchoolDesignation" type="
TitleIPartASchoolDesignationType" minOccurs="0">
                          . . .
                    <xs:element name="MagnetSpecialProgramEmphasisSchool"
type="MagnetSpecialProgramEmphasisSchoolType" minOccurs="0">
                          . . .
                    <xs:element name="AdministrativeFundingControl" type="
AdministrativeFundingControlDescriptorReferenceType" minOccurs="0">
                          . . .
                    <xs:element name="InternetAccess" type="
InternetAccessType" minOccurs="0">
                          . . .
                    <xs:element name="LocalEducationAgencyReference" type="
LocalEducationAgencyReferenceType" minOccurs="0">
                          . . .
                    <xs:element name="CharterApprovalAgencyType" type="
CharterApprovalAgencyType" minOccurs="0">
                          . . .
                    <xs:element name="CharterApprovalYear" type="
SchoolYearType" minOccurs="0">
                          . . .
              </xs:sequence>
         </xs:extension>
     </xs:complexContent>
</xs:complexType>
```

[1] Adapted from "PESC Guidelines for XML Architecture and Data Modeling", Version 3.0, Postsecondary Electronic Standards Council (PESC), April 29, 2005.