Unifying Data Model - Design Principles

The building blocks of the Ed-Fi Unifying Data Model (UDM) are entities (classes), elements (attributes), and relationships (associations). Even with such simple building blocks there are a multitude of design choices. The following design principles are used in creating the Ed-Fi UDM.

Core Principles

The Ed-Fi UDM embodies the following core principles:

- Is easily understood
- Is independent of any specific application or physical implementation
- Implements a common core that provides consistency across applications
- Easily extends and accommodates change and evolution
- Logically separates the data from the handler mechanisms and implementation-specific information

Additional Principles

The Ed-Fi UDM was developed in accordance with the following general principles.

Stay Within the Defined Scope

The following questions are used to select what is included in the Ed-Fi UDM:

- Does the entity, attribute, or relationship exist within the defined K–12 education scope?
- Is the entity, attribute, or relationship important for a specific educational purpose?
- Is the entity, attribute, or relationship available to be collected?
- Are there scenarios where the entity, attribute, or relationship is shared or exchanged?

When an education data element is considered, it is not included in the Ed-Fi UDM without some evidence that:

- The data element is being electronically captured in an application or information system used by an education organization.
- The data element has a use that requires sharing or interchange, such as for analytics, decision-making, or state and federal reporting.

Consider the “Natural State and Structure” of the Data

The Ed-Fi UDM is constructed with easily understood semantics:

- Entities are naturally the most important “things” in the domain that require representation in data.
- Attributes naturally identify, describe, characterize, or classify entities.
- Relationships (associations) between entities are not transient and typically persist over time.

Include Generalizations Sparingly

The Ed-Fi UDM includes generalizations that contribute critical inheritances and represent important generalization concepts in the domain. The following questions are used to determine whether a generalization should be included:

- Is the generalization a commonly used term within the domain?
- Does the generalized entity have common important attributes that should be inherited?
- Does the generalized entity have associations that should be inherited?
- Are all of the specialized entities (subclasses) of common purpose and structure?
- Do source systems commonly identify the lower-level abstractions or store and display them in the same structure and context?

The generalization EducationOrganization has both important attributes and associations that are meaningfully inherited by school, district, regional educational service center, and other sub-classes of EducationOrganization. Note that all are of a similar common purpose and structure.

By contrast, the concept of a "person" is not reflected in the UDM, as it fails the final two questions in the above list. While it is tempting to use an inheritance structure to create efficiency around storage and exchange of records for students and teachers, the roles of student and teacher are quite distinct, and those records are most often stored in separate K–12 systems-of-record. Inheritance is generally most useful when an entity is of one and only one subclass (e.g., an EducationOrganization is a school, LEA, RESA, etc., but not multiple of these), and the Ed-Fi UDM adopts this pattern.

Create Classes to Abstract Cohesive Groups of Attributes

UDM Documentation Contents

Explore the Ed-Fi Unified Data Model:

UDM Reference:

- Ed-Fi Unifying Data Model Handbook
- Ed-Fi Unifying Data Model UML Diagrams (Visio format, on GitHub)

To improve understandability, the Ed-Fi UDM abstracts cohesive collections of attributes into classes. These generally become JSON objects within API resources in the REST API bindings (in a few cases these classes are "inlined" to simplify the API model), and complex types in Ed-Fi XML schemas. For example, a student’s name could be represented as a flat structure with attributes FirstName, MiddleName, LastSurname, and NameSuffix. A new class (in XML, a new complex type) is created for Name that includes the attributes above. This reduces the number of attributes directly shown for Student without loss of understandability.

Examples where the Ed-Fi UDM applies this technique are as follows:

- Address
- BirthData
- Telephone

Create Classes to Group Attributes that are Multi-Valued Records

Some attributes are expressed as multi-valued records. Rather than create a separate domain class, the model defines the record attributes as a separate class, and then references a multi-valued attribute. These become collections in JSON, complex types in XML Schema, and so forth. For example, students may have multiple disabilities with each specified by a disability type, a description, and an order of severity (primary or secondary).

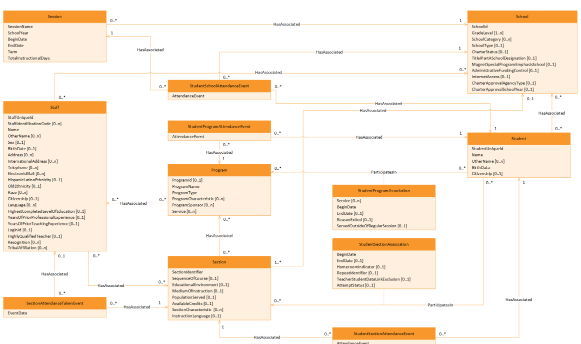
Examples where the Ed-Fi UDM applies this technique are as follows:

- Disability
- Language

Consider the Direction of Associations in the Context of Possible Interchanges

When associations do not have their own attributes, the reference embodying the association is contained in the source entity. It is important to consider which entity is the source for the association, based on its likely use in the interchange. Field usage and needs are commonly the guide for determining the direction of an association.

Consider, for example, the associations in the Student Attendance domain shown in the diagram below.



As shown, the association relating a student to the attendance event is contained in the source AttendanceEvent. It is reasonable to assume that there could be an interchange loading only attendance events. Note that if the direction of this association were reversed, then the loading of attendance would always need to be accompanied by a load of students. This same rationale applies to the section and program associations.

Use Real Data to Validate the Practicality of the Ed-Fi Unifying Data Model

Data profiling is a recognized technique for designing database schemas and data warehouse structures. Similarly, analyzing real data as it is housed in student information and other education systems provides significant validation of the Ed-Fi UDM.

In addition, the process and rigor of representing the Ed-Fi UDM as both API Resources in JSON and as an XML schema significantly adds to the depth and quality of the model. The Alliance also produces database schema (generally non-normative, but still useful) to further test the model.