Extensions and MetaEd

Sayee Srinivasan
Extensibility

• Ed-Fi standards are extensible, meaning that individual agencies can add new entities, attributes and associations*

• Extensibility is the opposite of standardization, so why support it?
  • K12 is full of diverse data mandates and no single standard can capture this diversity
  • Extensibility allows the community to try new models and exchanges, which can grow the standards
    • Teacher Prep
    • Finance Domain
    • Assessment Roster
Great!

- My API is asking for data not in the Ed-Fi model
- My API is asking for granular data
- I just want the database field to be in the ODS, but my API specs will not require systems to submit this data

Question Yourself…

- My API is asking for an aggregate value not in source systems
- My API is asking for an aggregate value to avoid asking for a whole lot of other data
- My APIs will force business logic downstream onto SIS systems (or others)

Evil!

- My API is asking for aggregates of data I already have
- I am replicating my local data domains in Ed-Fi to simplify other local processes (e.g. keeping local names, etc.)
- I have not consulted the extensions of others in the community
Why “evil”?

- More API differences = more custom code for providers = more bugs for everyone
- More code for providers = higher costs for school districts
- More data model differences = more expense for SEAs to collaborate
history of extensions
challenging
duplication
synchronization
SQL
CREATE TABLE [extension].[StaffExtension](
[StaffUSI] [INT] NOT NULL,
[ProbationCompleteDate] [DATE] NULL,
[Tenured] [BIT] NULL,
CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED ( [StaffUSI] ASC 
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, I

CREATE TABLE [extension].[StaffExtension](
    [StaffUSI] [INT] NOT NULL,
    [ProbationCompleteDate] [DATE] NULL,
    [Tenured] [BIT] NULL,
    CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED ( [StaffUSI] ASC )
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, INFOPRINT = OFF, STATISTICS = ON)
CREATE TABLE [extension].[StaffExtension](
  [StaffUSI] [INT] NOT NULL,
  [ProbationCompleteDate] [DATE] NULL,
  [Tenured] [BIT] NULL,
  CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED (  
    [StaffUSI] ASC  
  ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, I...
CREATE TABLE [extension].[StaffExtension](
  [StaffUSI] [INT] NOT NULL,
  [ProbationCompleteDate] [DATE] NULL,
  [Tenured] [BIT] NULL,
  CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED ( [StaffUSI] ASC )
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, I...
CREATE TABLE [extension].[StaffExtension](
    [StaffUSI] [INT] NOT NULL,
    [ProbationCompleteDate] [DATE] NULL,
    [Tenured] [BIT] NULL,
    CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED ( 
        [StaffUSI] ASC 
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, I

ALTER TABLE [extension].[StaffExtension] WITH CHECK ADD CONSTRAINT [FK_StaffExtension_Staff] FOREIGN KEY ([StaffUSI]) REFERENCES [edfi].[Staff] ([StaffUSI])
ON DELETE CASCADE
CREATE TABLE [extension].[StaffExtension](
    [StaffUSI] [INT] NOT NULL,
    [ProbationCompleteDate] [DATE] NULL,
    [Tenured] [BIT] NULL,
    CONSTRAINT [StaffExtension_PK] PRIMARY KEY CLUSTERED ( 
        [StaffUSI] ASC 
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNOR 
)

ALTER TABLE [extension].[StaffExtension] WITH CHECK ADD CONSTRAINT 
[FK_StaffExtension_Staff] FOREIGN KEY ([StaffUSI]) REFERENCES [edfi].[Staff] ([StaffUSI]) 
ON DELETE CASCADE
CREATE TABLE [extension].[StaffSeniority](
    [CredentialFieldDescriptorId] [INT] NOT NULL,
    [NameOfInstitution] [NVARCHAR](75) NOT NULL,
    [StaffUSI] [INT] NOT NULL,
    [YearsExperience] [DECIMAL](5, 2) NOT NULL,
    [CreateDate] [DATETIME] NOT NULL,
    CONSTRAINT [StaffSeniority_PK] PRIMARY KEY CLUSTERED (    
        [CredentialFieldDescriptorId] ASC,
        [NameOfInstitution] ASC,
        [StaffUSI] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_ 
ALTER TABLE [extension].[StaffSeniority] WITH CHECK ADD CONSTRAINT 
[FK_StaffSeniority_Staff] FOREIGN KEY ([StaffUSI]) REFERENCES [edfi].[Staff] ([StaffUSI]) 
ON DELETE CASCADE
other files
<xs:element name="ProbationCompleteDate" type="xs:date" minOccurs="0">  
  <xs:annotation>  
    <xs:documentation>The date the probation period ended or is scheduled to end.</xs:documentation>  
  </xs:annotation>  
</xs:element>  

<xs:element name="Tenured" type="xs:boolean" minOccurs="0">  
  <xs:annotation>  
    <xs:documentation>Indicator or whether the staff member is tenured.</xs:documentation>  
  </xs:annotation>  
</xs:element>  

<xs:element name="MembershipFTE" type="EXTENSION-MembershipFTE" minOccurs="0">  
  <xs:annotation>  
    <xs:documentation>Student membership as a percentage</xs:documentation>  
  </xs:annotation>  
</xs:element>  

<xs:element name="ResidenceZoneSchoolReference" type="SchoolReferenceType" minOccurs="0">  
  <xs:annotation>  
    <xs:documentation>The school whose residence zone the student lives in.</xs:documentation>  
  </xs:annotation>  
</xs:element>
complex
simplify
MetaEd
what is MetaEd?
language
language
generator technology
language

generator technology

integrated editor
Staff Extension

Domain Entity Staff additions

date ProbationCompleteDate
  documentation "The date the probation period ended or is scheduled to end."
  is optional

common Seniority
  documentation "Entries of job experience contributing to computations of seniority."
  is optional collection

bool Tenured
  documentation "Indicator of whether the staff member is tenured."
  is optional
lab 1
Your First Extension – Student School Association

StudentSchoolAssociation is the basic enrollment record in the Ed-Fi data model. It is often extended in SEA use cases. We will walk through these examples together.

Use case

1. SEA needs to capture the percentage of time the student is enrolled in school (e.g. for student counts, etc.)

2. SEA needs to capture the student’s school of residency (i.e. the student is attending a school for which they are not in the geographical residency area)

Answers: see slides below – answers inline
Open Atom and Configure Settings

1. Open the Atom editor
2. Go to MetaEd > Settings
3. Under section “Settings” set
   • Target Ed-Fi Data Standard Version to “3.1”
   • Target Ed-Fi ODS/API Version to “3.1.1”
4. Close the Settings tab
Create an Extension Project and File

1. Create an extension project
   a. **MetaEd > Create New Extension Project**
   b. Create a filesystem folder. You can put the folder wherever you want. Recommend: no spaces in folder name; name after your state abbreviations or state DOE (e.g. “ME”, “HI”, “FLDOE”, etc.)
   c. Highlight folder and click **Select Folder**

2. Name your project
   a. Double-click on **package.json** to open it
   b. Change **projectName** to the folder name you chose above (folder name and project name are not required to match, but it is a good idea)
   c. **File > Save (or Ctrl + S)**

This project name will appear in many places in the ODS API (database schema name and API extension namespace, among them)
```json
{
  "metaEdProject": {
    "projectName": "TEAExt",
    "projectVersion": "1.0.0"
  }
}
```
Continued...

3. Add an extension file
   Right-click your new folder/project and select Add MetaEd File > Association

4. Rename the extension file
   Right-click on the file Association.metaed and select Rename
   Name the file: StudentSchoolAssociation.metaed
Author Your Extension

Modify the contents of StudentSchoolAssociation.metaed

Before

Association ExampleName
documentation "This is documentation."
domain entity FirstEntityName
documentation "This is documentation."
domain entity SecondEntityName
documentation "This is documentation."
bool PropertyName
documentation "This is documentation."
is part of identity

After

Association ExampleName EdFi.StudentSchoolAssociation
additions
    documentation "This is documentation."
    domain entity FirstEntityName
        documentation "This is documentation."
    domain entity SecondEntityName
        documentation "This is documentation."
    bool PropertyName
        documentation "This is documentation."
        is part of identity
    decimal MembershipFTE
        documentation "Student membership as a percentage"
        is optional
        total digits 3
        total places 2
What does it all mean?

Association ExampleName EdFi.StudentSchoolAssociation additions
  documentation "This is documentation."
  domain entity FirstEntityName
    documentation "This is documentation."
  domain entity SecondEntityName
    documentation "This is documentation."
  bool PropertyName
    documentation "This is documentation."
    is part of identity
    is optional
    total digits 2
    decimal places 2
Build the Extension

1. Save
   *File > Save All*

2. Build
   *MetaEd > Build*

3. Look for build errors in the console, fix, and rebuild if necessary!

4. Inspect the results
   In the filesystem, open these files:
   
   - [extension folder]/MetaEdOutput/extension/Database/SQLServer/ODS/Structure/0020-EXTENSION-extension-Tables.sql
   - [extension folder]/MetaEdOutput/extension/Database/SQLServer/ODS/Structure/0030-EXTENSION-extension-ForeignKeys.sql
An Aside on Extending

Is this the best modeling option?

• Calculate using Student.Address?
• Extend Student?
• Use Student School Responsibility Association?

Association EdFi.StudentSchoolAssociation additions
  decimal MembershipFTE
  documentation "Student membership as a percentage as a percentage is optional"
  total digits 2
  decimal places 2
  domain entity EdFi.School
  documentation "The school whose residence zone is optional with context ResidenceZone"
An Aside on Extending

In extending, you are asking others (mostly technology providers) to make exceptions for you (i.e. to do more work than normal)

• When is it right to do that?
• When is it not ok?

Association EdFi.StudentSchoolAssociation additions
  decimal MembershipFTE
  documentation "Student membership as a percentage "
  is optional
  total digits 2
  decimal places 2
  domain entity EdFi.School
  documentation "The school whose residence zone the student li"
The JSON representation of the "studentSchoolAssociation" resource to be created or updated.

Example Value | Model
---|---

```json
{
  "studentSchoolAssociation": {
    "required": true,
    "body": {
      "employedWhileEnrolled": true,
      "entryGradeLevelDescriptor": "string",
      "entryGradeLevelReasonDescriptor": "string",
      "entryTypeDescriptor": "string",
      "exitWithdrawDate": "2019-11-19",
      "exitWithdrawTypeDescriptor": "string",
      "primarySchool": true,
      "repeatGradeIndicator": true,
      "residencyStatusDescriptor": "string",
      "schoolChoiceTransfer": true,
      "etag": "string",
      "_ext": {
        "Sample": {
          "membershipTypeDescriptor": "string"
        },
        "IEAExt": {
          "membershipFTE": 0,
          "residenceZoneSchoolReference": {
            "schoolId": 0,
            "link": {"rel": "string", "href": "string"
          }
        }
      }
    }
  }
}
```
documentation and troubleshooting
Key Documentation

• MetaEd space (all documentation)
  • https://techdocs.ed-fi.org/display/METAED

• Language Specification (formal language description)
  • https://techdocs.ed-fi.org/display/METAED/Language+Specification

• Cookbook (modeling HOWTOs)
  • https://techdocs.ed-fi.org/display/METAED/Cookbook

• Extensions Framework (allowed and disallowed patterns)
  • https://techdocs.ed-fi.org/display/EFDS31/Ed-Fi+Extension+Framework
MetaEd build failures

1. Read the error message
2. Check the Extensions Framework
3. Update the MetaEd IDE
4. Raise a Tracker Ticket in MetaEd project
   • Attach entire extension project as a .zip

MetaEd vs ODS API Build Issues

It is possible that MetaEd will build, but the ODS API will not. In these cases, raise an ODS API ticket. The ODS API team will route the ticket to MetaEd if necessary.

Attach entire extension project as a .zip!
Extension Framework – Dos and Don’ts
Supported Extensions

• Add a New Domain Entity
• Add a New Association Entity
• Add New Descriptor
• Add New Shared Simple Type
• Add New Common Type
Add a New Domain Entity

Domain Entity StaffEvaluation documentation "TBD"
string StaffEvaluationTitle documentation "The name or title of the staff evaluation."
is part of identity
min length 1
max length 50

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least one identity property</td>
<td>Identity property updates for new domain entities</td>
</tr>
<tr>
<td>Can contain more than one identity property</td>
<td></td>
</tr>
<tr>
<td>Properties can be required/optional/single/collection</td>
<td></td>
</tr>
</tbody>
</table>
Add a new Association Entity

Association
StudentMigrantEducationProgramAssociation

based on
GeneralStudentProgramAssociation

documentation “TBD”
bool PriorityForServices
documentation "priority for services"
is required

Inherited from base class
GeneralStudentProgramAssociation

Do’s

Unique Name

Name should have “Association” at the end

Additional attributes beyond reference to the associated entities

Required/Optional/Collection properties
Add a new Descriptor

• Descriptor AcademicSubject documentation "This descriptor holds the description of the content or subject area (e.g., arts, mathematics, reading, or a foreign language)."

<table>
<thead>
<tr>
<th>Dont’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map type</td>
</tr>
<tr>
<td>“Descriptor” in the name</td>
</tr>
<tr>
<td>Additional Properties</td>
</tr>
</tbody>
</table>
Add a new Shared Simple Type

• Shared String FirstName [49]
  • documentation "A name given to an individual at birth, baptism, or during another naming ceremony, or through legal change."
  • min length 1
  • max length 75

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique Name</td>
<td>Properties having other than simple type definition</td>
</tr>
<tr>
<td>Minimum appropriate details to define decimal, string, or integer</td>
<td></td>
</tr>
</tbody>
</table>
Add a new Common Type

Do’s

- May contain an identity property
- Must contain an identity property if used as a collection
- Must contain at least one property
- Required/optional/single/collection
Add new attributes to an existing Domain Entity

Domain Entity Staff additions

- date ProbationCompleteDate
documentation "The date the probation period ended"
is optional
- bool Tenured
documentation "Indicator of whether the staff member is tenured."
is optional

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contain at least one property</td>
<td>Identity</td>
</tr>
<tr>
<td>All properties must be optional</td>
<td>Entity-level documentation</td>
</tr>
<tr>
<td></td>
<td>Properties from the parent association unless additional context is provided</td>
</tr>
<tr>
<td></td>
<td>Required Properties</td>
</tr>
</tbody>
</table>
Add a new attribute to an existing Association Entity

Association
StudentMigrantEducationProgram
Association **additions**

descriptor EnrollmentType
documentation “Migrant Education Program Enrollment Type”
is optional

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contain at least one property</td>
<td>Identity</td>
</tr>
<tr>
<td>All properties must be optional</td>
<td>Entity-level documentation</td>
</tr>
<tr>
<td></td>
<td>Properties from the parent association unless additional context is provided</td>
</tr>
<tr>
<td></td>
<td>Required Properties</td>
</tr>
</tbody>
</table>
## Subclass Domain Entities/Associations

Subclassing of abstract entities such as EducationOrganization and GeneralStudentProgramAssociation is supported.

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contain at least one property</td>
<td>Identity</td>
</tr>
<tr>
<td>Follow the same naming convention as the parent</td>
<td>Properties from the parent</td>
</tr>
<tr>
<td>Subclass entities other than domain and association entities</td>
<td>Subclass descriptor/common types</td>
</tr>
</tbody>
</table>
Unsupported Extensions
Use Case – Extend SubClass Entity–Possible in v3.0 not in v2.0

• Any Domain Entity or Association subclass of StudentProgramAssociation or EducationOrganization can not be extended.

• Work Arounds:
  • Create a completely new domain entity or association that exists side-by-side with the core domain entity or association.
No, No and No

- Don’t remove a Core entity
- Don’t modify a Core element
- Don’t change Core data element definition
- Don’t remove a Required element
- Don’t change Optional element to Required element in Core
- Don’t change Optional collection to Required collection in Core
Additional Unsupported Modifications

- You cannot create new Inline Common and Enumeration types
  - Inline Common – No child table gets created.
  - Common – Child table gets created.
  - Use Common and Descriptor instead
Questions