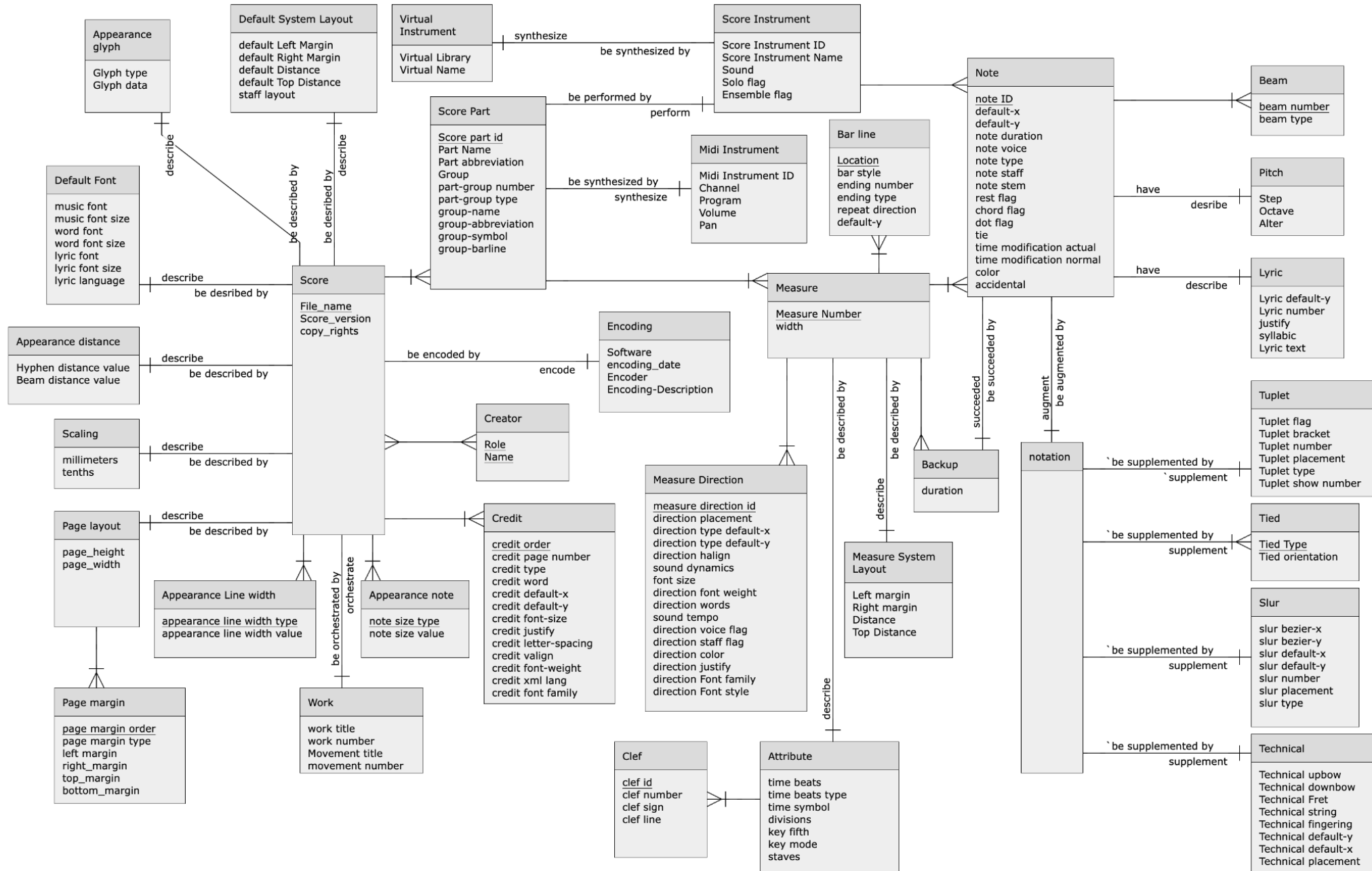


LDS



## Screenshots

```
<!--=====
<measure number="5" width="266">
  <note default-x="27">
    <pitch>
      <step>D</step>
      <alter>-1</alter>
      <octave>5</octave>
    </pitch>
    <duration>24</duration>
    <voice>1</voice>
    <type>eighth</type>
    <accidental>flat</accidental>
    <stem default-y="20">up</stem>
    <beam number="1">begin</beam>
    <lyric default-y="-80" number="1">
      <syllabic>single</syllabic>
      <text>Je</text>
    </lyric>
  </note>
  <note default-x="66">
    <pitch>
      <step>F</step>
      <octave>4</octave>
    </pitch>
    <duration>24</duration>
    <voice>1</voice>
    <type>eighth</type>
    <stem default-y="10">up</stem>
    <beam number="1">end</beam>
    <lyric default-y="-80" number="1">
      <syllabic>begin</syllabic>
      <text>rê</text>
    </lyric>
  </note>
```

```

<score-part id="P1">
  <part-name>Voice</part-name>
  <score-instrument id="P1-I14">
    <instrument-name>Choir Aahs</instrument-name>
    <instrument-sound>voice.vocals</instrument-sound>
    <solo/>
  </score-instrument>
  <midi-instrument id="P1-I14">
    <midi-channel>1</midi-channel>
    <midi-program>53</midi-program>
    <volume>80</volume>
    <pan>0</pan>
  </midi-instrument>
</score-part>
<score-part id="P2">
  <part-name>Piano</part-name>
  <score-instrument id="P2-I1">
    <instrument-name>Grand Piano</instrument-name>
    <instrument-sound>keyboard.piano</instrument-sound>
  </score-instrument>
  <midi-instrument id="P2-I1">
    <midi-channel>2</midi-channel>
    <midi-program>1</midi-program>
    <volume>80</volume>
    <pan>0</pan>
  </midi-instrument>
</score-part>

```

```

<credit page="1">
  <credit-type>title</credit-type>
  <credit-words default-x="1447" default-y="3477" font-size="19.5" justify="center" valign="top">Prelude to a Tragedy
</credit-words>
</credit>
<credit page="2">
  <credit-type>composer</credit-type>
  <credit-words default-x="2718" default-y="3387" font-size="7.8" justify="right" valign="top">Lee Actor (2003)
</credit-words>
</credit>

```

```
<movement-title>Prelude to a Tragedy</movement-title>
<identification>
  <creator type="composer">Lee Actor</creator>
  <rights>© 2004 Polygames.    All Rights Reserved.</rights>
  <encoding>
    <software>Finale v27.0 for Mac</software>
    <encoding-date>2021-06-04</encoding-date>
  </encoding>
</identification>
```

## Prelude to a Tragedy

Moderato (♩ = 85) Lee Actor (2003)

The musical score is for a full orchestra and harp. The tempo is Moderato (♩ = 85). The key signature has one sharp (F#). The score is divided into two systems. The first system includes Piccolo, Flutes 1 & 2, Oboes 1 & 2, English Horn, Clarinets in Bb 1 & 2, Bass Clarinet in Bb, Bassoons 1 & 2, Horns in F 1 & 2, 3 & 4, Trumpets in C 1 & 2, Trombones 1 & 2, 3, Tuba, Timpani, Percussion 1 & 2, and Harp. The second system includes the same instruments. The score shows a series of rests for most instruments, with some woodwinds and strings playing a melodic line in the second system. The harp plays a series of chords marked 'mf'.

# Writeup:

## Purpose:

**1. Enables Compatibility:** The data model ensures that musical compositions can be accurately and consistently represented, making it easier to exchange music data between different software and platforms.

**2. Archival Support:** With a standard format for sharing documents, storing the MusicXML file standalone creates a means for easy (and potentially lightweight) archiving. Software itself does not need to be archived because there are over 260 scoring softwares that actively support the MusicXML standard.

**3. Western Musical Notation:** It serves as a reliable means of preserving the nuances of and representing Western musical notation.

**4. Statistical Analysis:** The MusicXML model provides an easily ingestible software format for data collection and analysis.

## Users:

**1. Music Notation Software Developers:** It can be used as a basis for creating software applications related to music. By providing a general structure for what elements of western sheet music are important, software developers can focus less on the structure of the sheet music.

**2. Musicians and Composers:** While composers may be writing new music, musicians playing it, or someone arranging another piece they all must exchange information about the music in some form. With so many different applications on the market MusicXML provides a standard so that information may still be exchanged regardless of what applications are involved on any end

**3. Music Teachers:** They can employ the MusicXML model for creating teaching materials. By providing a standard communication form for students, there can be faster turnaround for modeling and creation of sheet music versus pen and paper.



**4. Researchers:** Researchers and scholars can use the data model for in-depth analysis and study of music. An example is an analysis conducted in the link below for, “whether Bach’s pieces really have 90% of its notes in one of two durations.”

(<https://www.musicxml.com/publications/makemusic-recordare/notation-and-analysis/musicxml-analysis-examples/>)

### **Easy Parts Of Reverse Engineering:**

1. MusicXML is a standardized specification. With disagreements between individuals and data sets, there was a final piece that allowed dispute resolution.
2. Class hierarchy of XML helped to develop explicit relationships.
3. Because the data is in XML format, we had access to other tools to aid in reverse engineering.

### **Hard Parts Of Reverse Engineering:**

1. Validating the reverse engineered model with a wide range of MusicXML types of compositions is a complex task. No central repository for MusicXML files. Likely many behind a paywall.
2. With a wide range of software used to export/import MusicXML files, some softwares represented the same types of data differently. One file may instantiate something on a piece of music differently than another, even if they are the same visual representation.
  - a. This also applies to identifying descriptors for certain entities. Some softwares clearly defines certain attributes as meant to be used as identifiers. However when looking at other example files those identifying characteristics are lost. This created the need for a more abstract model.

3. Class hierarchy of XML obscured non-direct relationships. More complex relationships were extremely difficult to identify and subsequently model.
4. Understanding what was useful data to collect and model. There were multiple perspectives to try and be aware of when working with this data model. The perspective of a musician, software developer, or even composer. These perspectives not only impact what data might be important to store under the hood, but also what data should be represented visually.
5. Agreements on model shapes. There were a number of instances where team members disagreed on certain shapes for certain entities and relationships.
6. Team members staunchly believing one way or another. This often resulted from differing perspectives of the model. Whether we are creating a model to store multiple files, or are modeling the data within one file. A common disagreement point was whether we care about the visual representation of the data (sheet music) vs the raw XML data.

### **List Of Questions:**

1. Are you creating sheet music, ingesting sheet music, doing statistical analysis, or writing the MusicXML standard?
2. Are you more concerned with all the data stored, or the visual representation of the data in sheet music?
3. What software are you using to build your scores?
4. How important is the visual layout of the music (e.g., page size, margins, line spacing) for your work?
5. If some attributes have the same name (lots of 2d location attributes): should these be representative of all <list of entities>, or is it specific to each <entity>?





[illegible]