

The AAF File Format

A Nuts and Bolts, Bottom Up,
Look at AAF Files
And the AAF Open Source SDK

143rd SMPTE Technical Conference
New York, 5 Nov 2001

Jim Trainor, AAF Association

The AAF File Format

“The AAF Object Specification defines a structured container for storing essence data and metadata using an object-oriented model”

- AAF Developers' Guide

The AAF File Format

“The *AAF Low-Level Container Specification* describes how each object is stored on disk. The *AAF Low Level Container Specification* uses Structured Storage...”

- AAF Developers' Guide

Structured Storage

“Structured Storage provides a solution by defining how to treat a single file entity as a structured collection of two types of objects—storages and streams”

- Microsoft MSDN Documentation

Structured Storage

- Structured Storage files contain a tree of storage objects. (COM interface: IStorage)
- The tree reflects object containment (i.e. “encapsulation”, or “ownership”).
- Storage object contain other storage objects *and* contain data streams (COM interface: IStream).
- Analogous to directories and files.

Structured Storage

- Structured Storage objects store data only.
- How are these data objects associated with code that knows how to process them?

Structured Storage

- Structured Storage requires that each storage object has a *CLSID*
- CLSID = Class ID (unique)
- The CLSID is used to identify the COM interface that understands the data stored in the object.

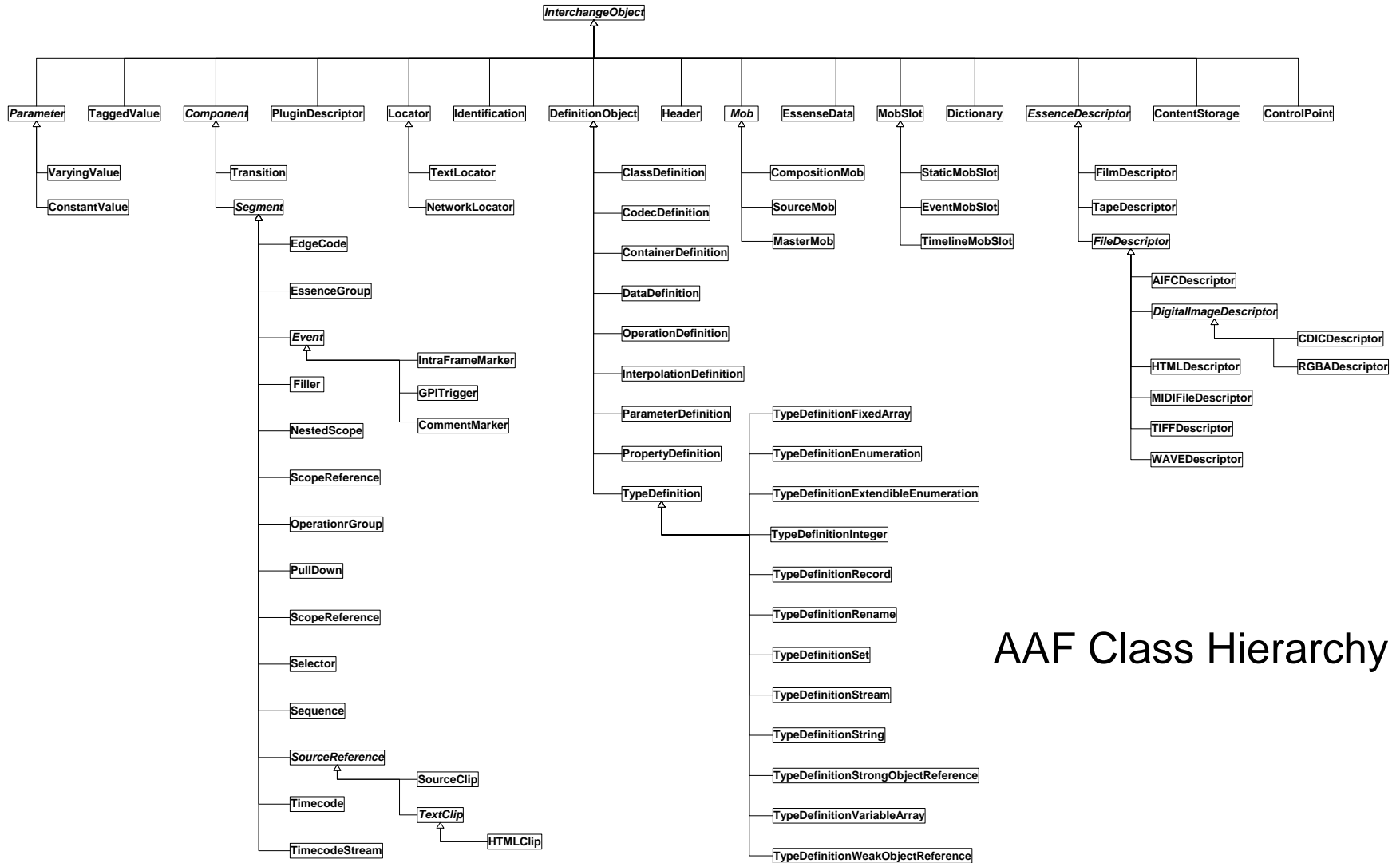
SMPTE Unique Labels

- Finally, something to do with video!
- SMPTE 298M - Universal Labels for Unique Identification of Digital Data
- A subset of the SMPTE 298M namespace is devoted to the AAF.
- SMPTE 298 AAF Subset
= Structured Storage Class ID's

AAF Object Model

- Stored objects specifications are found in the “AAF Specification Version 1.1”.
- The specification defines a class inheritance hierarchy for AAF object classes, and,
- It specifies individual classes, and ...
- It specifies object containment relationships.

AAF Object Model



AAF Class Hierarchy

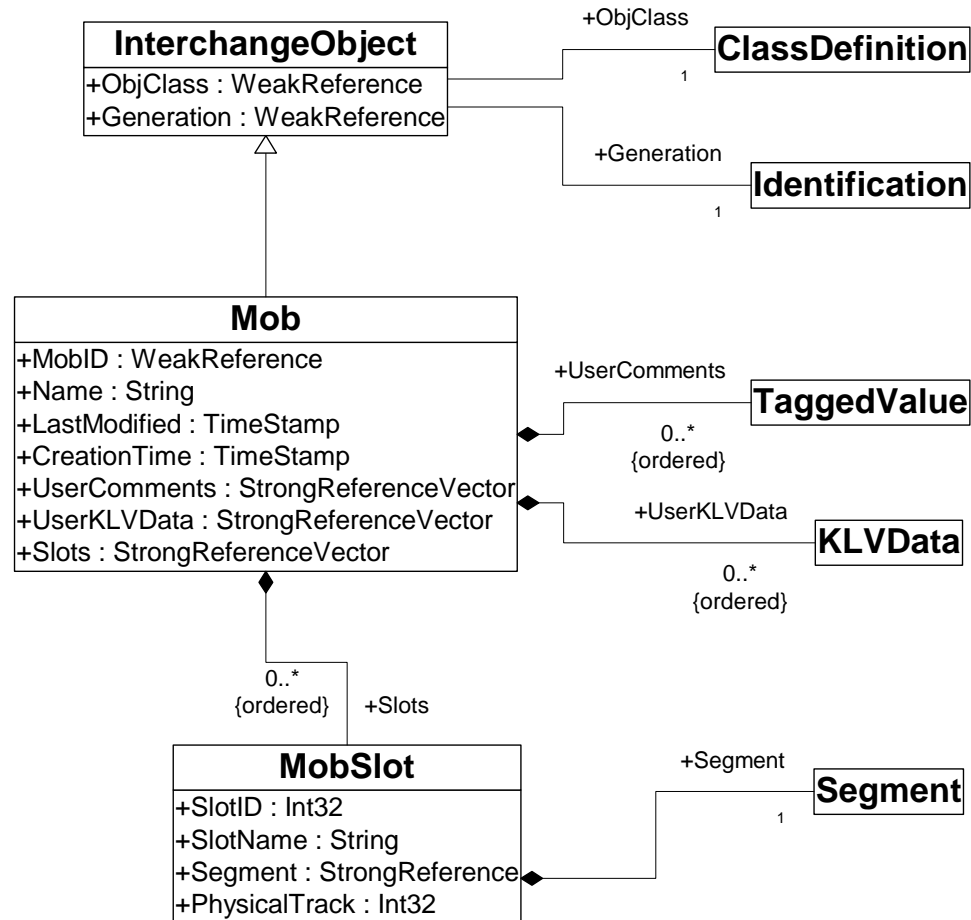
AAF Object Model

- Objects are simply a set of properties.
- A property is a value with a defined name and type.
- Example:

Property Name	Property Type	Value
LastModified	TimeStamp	Fri Nov 2 14:47:35 2001

AAF Object Model

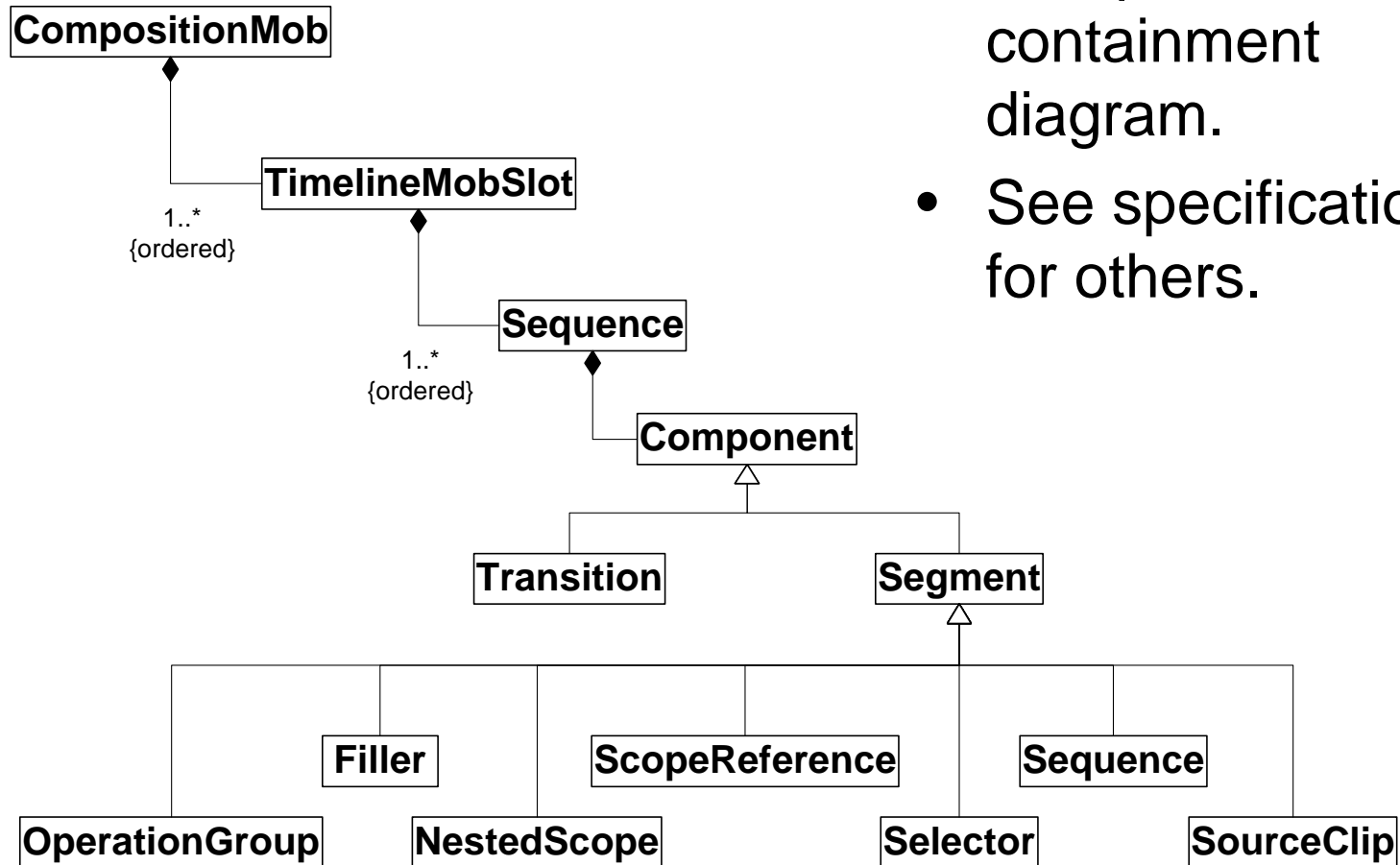
- Example class specifications.
- These are stored objects.
 - Data Only!



AAF Object Model

- Object Containment versus Association:
 - Type “StrongReference” indicates ownership.
(A StrongReference refers to a Structured Storage IStorage object encapsulated by the owner’s IStorage object.)
 - Type “WeakReference” is an association.
- Class instances have UUID’s that are used for both strong and weak references.

AAF Object Model



- CompositionMob containment diagram.
- See specification for others.

AAF Object Model

- There are 80 objects defined in the “AAF Specification Version 1.1”.
- It doesn't stop there.
- The object model is extensible.
- You can define new classes, new properties, new property types, and new essence types.

AAF Object Model

- How many ways are there to combine these objects?
- Many!
- Maybe infinite!
- Implementation Guidelines such as Operation Patterns required to constrain the model.

AAF File Format - Summary

- Structured Storage is the persistent object store.
- SMPTE UL's for Class ID's
- AAF Specification 1.1 defines the object model.
- It is extensible.
- Implementation guidelines are required to constrain the object model.

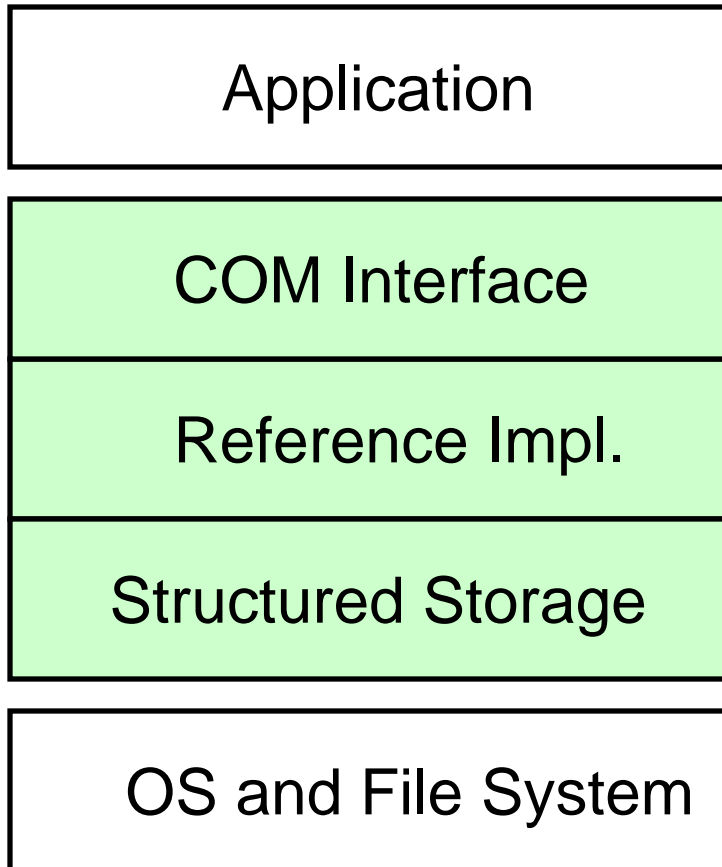
Low Level Object Spec.

- For all the excruciating details visit:

<http://aaf.sourceforge.net/docs>

- See:
 - OM Stored Format
 - OM Design Specification
 - AAF Specification 1.1

The AAF Open Source SDK



- COM interface used to access and manipulate stored objects.
- Reference implementation implements the AAF object model.
- Structured Storage provides the persistent object store.

The AAF Open Source SDK

- Platforms:
 - Win2K
 - Mac
 - Linux
 - Irix
- Structured Storage is ported to non Microsoft platforms.
- A minimal COM implementation is used to support the COM interface on non Microsoft platforms.

The AAF Open Source SDK

- There are 145 COM interfaces.
- The COM objects implement interfaces that reflect the structure of the stored AAF objects.
- There are additional COM interfaces that implement transient objects such as iterators (called enumerators in SDK).
- <http://aaf.sourceforge.net/docs/com-api/contentsf.html>

The AAF Open Source SDK

- The SDK also has a plugin codec model.
- Essence is accessed using plugin codecs.
- Raw access to essence is also possible.
- Examples of codecs:
 - WAV audio codec
 - AIFC audio codec
 - CDIC video codec
 - JPEG video codec
 - BBC MPEG2 codec
 - Roll your own.

AAF SDK - First Program

- Short program to create a new AAF file that contains a single named MasterMob.
- Mobs are Metadata Objects that describe essence.
- Master Mobs store metadata required to locate essence data. i.e. a tape or file location, or embedded essence.

AFF SDK - First Program

- Load the AAF Library with `AAFLoad()`

```
#include <AAF.h>
#include <AAFStoredObjectIDs.h>

int main( int argc, char** argv )
{
    AAFLoad( "./libcom-api.so" );
}
```


AAF SDK - First Program

- Create a new file.

```
aafProductIdentification_t ident =  
    { L"AAF Association", // Company Name  
      L"First File",      // Product Name,  
      L"2.71828182818",  // Product Version  
      {0x89aa595e, 0x51ec, 0x8ce8,  
        {0x8c, 0xe8, 0x37, 0x81, 0x8d, 0xef, 0x78, 0xf3}} // Product UID  
    };  
  
IAAFFile* file;  
AAFFileOpenNewModify( L"./my_first_aaf_file.aaf", 0, &ident, &file );
```

AAF SDK - First Program

- Get the file header, and dictionary.

```
IAAFHeader* header;  
file->GetHeader( &header );
```

```
IAAFDictionary* dictionary;  
header->GetDictionary( &dictionary );
```

AAF SDK - First Program

- Use the dictionary to create a MasterMob.

```
IAAFMasterMob* masterMob;
```

```
dictionary->CreateInstance(  
    AUID_AAFMasterMob, // The SMPTE object ID  
    IID_IAAFMasterMob, // The COM interface ID,  
    reinterpret_cast<IUnknown**>(&masterMob) );
```

```
masterMob->Initialize();
```

AAF SDK - First Program

- Name the Mob. Note use of QueryInterface.

```
IAAFMob* mob;
```

```
masterMob->QueryInterface(  
    IID_IAAFMob,  
    reinterpret_cast<void**>(&mob) );
```

```
mob->SetName( L"First Mob" );
```

AAF SDK - First Program

- To save the mob, it is added to the header, then the file is then saved and closed.

```
header->AddMob( mob );
```

```
file->Save();
```

```
file->Close();
```

AFF SDK - First Program

- Finally, unload the COM module.

```
AAFUnload();
```

```
return 0;
```

```
}
```

AAF SDK - First Program

```
#include <AAF.h>
#include <AAFStoredObjectIDs.h>
#include <unistd.h>

int main( int argc, char** argv )
{
    // Load the default COM implementation library.
    AAFLoad( "./libcom-api.so" );

    // We will need a product identification structure.
    // Need an UID? Try Microsoft's UUIDGEN tool.
    aafProductIdentification_t ident =
    { L"AAF Association", L"First Mob", L"2.71828182818",
      {0x89aa595e, 0x51ec, 0x8ce8,
       {0x8c, 0xe8, 0x37, 0x81, 0x8d, 0xef, 0x78, 0xf3}},
      0, 0};

    // Create a new file.
    unlink( "./my_first_aaf_file.aaf" );
    IAAFFile* file;
    AAFFileOpenNewModify( L"./my_first_aaf_file.aaf", 0, &ident, &file );

    // We will need the header.
    IAAFHeader* header;
    file->GetHeader( &header );

    // The dictionary is required to create a new object
    IAAFDictionary* dictionary;
    header->GetDictionary( &dictionary );
```

```
// Now, use the dictionary to create a MasterMob.
IAAFMasterMob* masterMob;
dictionary->CreateInstance(
    AUID_AAFMasterMob,
    IID_IAAFMasterMob,
    reinterpret_cast<IUnknown**>(&masterMob) );
masterMob->Initialize();

// Name this mob "My First Mob"
IAAFMob* mob;
masterMob->QueryInterface( IID_IAAFMob,
    reinterpret_cast<void**>(&mob) );
mob->SetName( L"First Mob" );

// Add mob to the file (via the header)
header->AddMob( mob );

// Save and close the file.
file->Save();
file->Close();

// Unload the COM module.
AAFUnload();

return 0;
}
```

```
g++ firstAAF.cpp -ldl -rdynamic -I ../AAF/AAFi686LinuxSDK/g++/include -Xlinker -rpath -Xlinker \
../AAF/AAFi686LinuxSDK/g++/Examples/debug\ -L../aaf/cvs/AAF/AAFi686LinuxSDK/g++/lib/debug -laaflib -laafiid -o firstAAF
```

AAF SDK - First Program

- This program does not:
 - Have proper error handling!
 - Correctly release the COM interfaces!
 - Do anything useful!

AAF SDK - First AAF File Dump

Level	Desc.	Detail
1	Object	Header
2	Prop	ByteOrder
3	Value	IAAFTypeDefInt: (Int16) 18761 0x4949
2	Prop	LastModified
3	Value	IAAFTypeDefRecord: 2 members
2	Prop	Content
3	Value	IAAFTypeDefStrongObjRef: to object of class ContentStorage
4	Object	ContentStorage
5	Prop	Mobs
6	Value	IAAFTypeDefSet:
7	Value	IAAFTypeDefStrongObjRef: to object of class MasterMob
8	Object	MasterMob
9	Prop	MobID
10	Value	IAAFTypeDefRecord: 6 members
9	Prop	Name
10	Value	IAAFTypeDefString: First Mob
9	Prop	Slots
10	Value	IAAFTypeDefVariableArray: 0 elements of type IAAFTypeDefStrongObjRef
9	Prop	LastModified
10	Value	IAAFTypeDefRecord: 2 members
9	Prop	CreationTime
10	Value	IAAFTypeDefRecord: 2 members
2	Prop	Dictionary
2	Prop	Version
2	Prop	IdentificationList

Here is our “First Mob”

