Designing Data Models for Asset Metadata
Daniel Hurtubise
SGI

Abstract

The Media Asset Management System (MAMS) stores digital data and metadata used to support the mission of a company. In essence, the MAMS is similar to a traditional database with its entities and attributes but with added features such as media streaming, media transfers, and workflow dynamics. In fact, some vendors’ MAMS solutions use commercial databases to store the metadata and provide the ability to optimize the database to increase the MAMS search performance. They may also provide basic data models for assets that can be used immediately in the design or enhanced to meet specific needs. However, in almost all cases there may be a need for the creation of new data models that will require skills in some form of data modeling. This paper presents an overview of how data models can be used to define asset metadata and ultimately support a company’s mission. This paper is not intended to present the complex theories of data modeling, but rather uses basic data modeling techniques to show how data models can be designed for asset metadata.

Goals of Data Modeling

Once the data model is defined and illustrated, it becomes the tool that will guarantee cohesion and harmony during the development cycle. It encourages both the developer and the client to explore and revisit the essence or purpose of the application as the system scales and content types are added to the digital archive.

As the assets evolve in the digital media asset management system, layers are added to the application that can sometimes alter its core essence. The data model should always be referenced when enhancing the system and adding attributes to asset data model definitions. This will reduce iterative development efforts by the programmers when the data model is not enhanced in parallel with development, and problems can be avoided in the perceived functionality of the system by the user or in the established workflow of the assets.

The ultimate goal is to build a system that makes the asset more useable and effective, which translates into an effective media asset management system. The maintenance of the system’s data models will go a long way toward meeting the goals of the company, such as increased revenue, cost reduction, better workflow, and reliable asset content.
Data Modeling of Assets

Before purchasing any software or storing a single byte of media data in the digital asset management system, analysis of the client’s requirements is paramount. You cannot spend too much time doing this analysis, and in fact, it can take several iterations to clearly and completely define and come to an agreement on all the requirements for a major project.

For this reason, it is important to maintain focus during the modeling process by identifying and respecting the scope of the model or the scope of integration. The modeler, the management, and the eventual end user of the media asset management system must agree upon the scope. The worst thing a modeler can do is exclude a key person in the requirements analysis. The data model should follow the life cycle of the content from creation of the content to completion of the product.

In the end, the data model should yield an accurate representation of the client’s information needs and the company’s business process. In turn, this model will serve as the framework for the development of the new or enhanced system. A number of books have been written that describe the considerations and practices of data modeling, for example, Building the Data Warehouse by Inmon (1996) and The Data Modeling Handbook by Reingruber and Gregory (1994).

Assets in a Relational Model

A digital media asset management system is made up of assets consisting of attributes (metadata) and physical data (content). In a MAMS, the asset is an instance of an entity or a class of objects of significance to the business that uses metadata to support the playout, cataloging, browsing, and transferring of assets.

We see the word entity commonly used in the relational model of database design. Finkelstein (1989) defines an entity by stating, “A data entity represents some ‘thing’ that is to be stored for later reference. The term entity refers to the logical representation of data.” Consequently, when designing a MAMS, it may be possible to use traditional database-design methods.

Most people involved in database application development have heard of data modeling when it comes to designing common relational database applications. Data modeling practices are commonplace in database design with the use of modeling techniques such as entity-relation diagrams (ERD), data item sets (DIS), and physical data models.
Relational Model

Here is an example of a simplistic relational model of part of an asset management system. We can demonstrate that there is a many-to-many relationship between users and assets and show the attributes we intend to associate with these entities. However, it is not obvious how to demonstrate a MAMS check in, check out, and transfer workflow. Also, it would be more efficient for us to be able to show subtypes of assets and users that could give the workflow more granularity.

Extended Relational Entities

The media asset management system is often perceived as an archival system for digital content (assets). In fact, assets are more than collections of data. Assets in MAMS are more complex because they may represent content such as documents, images, video, and sound, or they may represent behavior such as rights management, business policies, and business strategies.

Therefore, it becomes apparent that the asset model inherits characteristics from the extended relational modeling schemes and the object management modeling schemes. It becomes practically impossible to represent the asset data model with basic entity-relationship diagrams. Nonetheless, this model could be used as a building block toward a more representative but more complex extended relational model.
The Asset Model

Using an extended-relational model, we can clear the cloud from our entity-relationship model and identify the check-in and check-out relationships between employees and assets. In addition, we can identify subtypes to the asset entity using the is-part-of (ISA) relationship. In a segmented example of a full data model that will be given below, the rights and approvals entity is added as the subtype. The appropriate employee can add rights and approvals to an asset by including the Rights and Approval data model with the asset metadata when he or she checks in the asset.

We could also use this type of modeling for assets that contain references to other assets, which would allow us to build composite assets. This is demonstrated later in our complete asset management system data model.

Asset Metadata
After you identify an asset, you then define it in real terms or through its attributes. An attribute is a property of the asset. It serves to identify, classify, qualify, quantify, or otherwise express the state of the asset. The attributes of interest for an asset called video clip could be creator, media kind, compression, sample rate, clip ID, reuse, creation date, formats, tracks, and purpose.

Attributes are specific pieces of information that need to be known or be held. The attributes reflect the need for the information they provide. In the analysis meeting, the participants should list as many attributes as possible. Later they can weed out those attributes that are not applicable to the application or those that the client is not prepared to spend resources to collect or
maintain. The participants come to an agreement on which attributes belong with an asset, as well as which attributes are optional.

Another method of determining attributes is to analyze existing content files to extract key words that may be useful in describing that content. Take for example, a Web page application form for a magazine subscription. The HTML source could be read to extract the fields that will need to be stored to identify the subscriber and the magazine requested. Some vendors’ APIs can also be used to read proprietary formats and extract specific information required for identifying the file-type information and thumbnail image generation.

Metadata for an asset may need to change during the course of its life cycle or workflow navigation. In other words, the asset could have different metadata needs as it goes from initial creation and editing to the release stages. This is known as metadata evolution. There can be a constant basic data model that remains with the digital asset, and it can be repeated for each asset version. For example, each asset could have an asset ID to uniquely identify it as well as other basic attributes such as creator, creation date, asset name, and version number. The version number is updated whenever an employee in a different role edits the asset’s metadata or content. Using a version number facilitates tracking the asset through its development. Finally, another attribute could be added to identify the source repository for the transferring of assets. The source repository ID can provide decision support for assets arriving at their destinations. This attribute identifies where the asset came from when its value is different than the repository ID, which may be stored as part of the asset ID as shown in the earlier asset model diagram. If it is the same, then it could mean either the asset was created in the repository where it is currently stored or it was transferred and the asset was modified in the new repository.
Distributed Streaming Media

This workflow diagram shows a media asset management system that deploys streaming media to edge servers.

The diagram depicts eight major subject areas.

1. Ingestion of original content—The content can arrive from a variety of locations in a variety of formats. This process of storing the original high-quality content is important to the company because if there is any problem with derived lower-bit-rate content in the future, having the original content provides a security net.

2. Content conversion—The content conversion group uses a client interface (Web- or platform-specific) to log in and retrieve the original content from the MAMS. The content is converted into various formats and bit rates that are relevant to the business requirements placed on the asset. The new formats are checked back into the MAMS to be reedited for later transfer to the edge repositories for broadcast.

3. Rights management and approvals—The process of deciding what content can be made available falls under the rights management and approvals process. The person responsible
for clearing the asset for distribution views the content and updates a metadata attribute to indicate that the asset is ready for use.

4. Content publisher—Once the rights have been cleared, the content publisher can assign how the assets are to be distributed to the edge repositories. The assets can be assigned to specific edge servers and published manually or a batch scheduling system can be timed to transfer the asset in time for the broadcast from the edge servers.

5. Assets publish—Once all the assets are stored in the proper format and all the appropriate metadata is updated, the asset is ready for publishing.

6. Assets transferred—Once the assets are marked as published, they are ready for the transfer process to the edge servers for broadcasting. This process either is automated via the MAMS transfer service or can be done manually via an in-house batch process depending on the requirements of the edge servers. For example, a weekly scheduled program can rely on the MAMS transfer service and a special news item can be transferred manually.

7. Assets streamed—The assets are streamed from the edge servers according to the requirements for that remote location. The broadcaster, according to the scheduled agreement, uses the content, and when that agreement expires the content is deleted. Once the asset is situated on the edge server, it is streamed directly from the remote MAMS via a streaming media plug-in.

8. Play statistics returned—The edge servers report back statistics about the assets that were played. This information is transferred in terms of metadata and stored on the main MAMS for further decision-support considerations and business analysis. Information that we are interested in is the maximum number of streams that were used, average bit rate pipe, system loads, and efficiency percentage. The goal in this process is to inform the creators of the performance of their streaming media so they can make improvements for future broadcasting. After the play statistics have been collected, another transfer process occurs that deletes the asset from the edge server.
Streaming Asset Data Model

This diagram demonstrates different data modeling concepts that can be helpful in designing the data model. The entities Asset and Employee illustrate is-part-of (ISA) relationships where Rights and Approvals, VideoNaudio, and Statistics are subtypes of Assets. Rights and Approval Editor, Content Publisher, and Technician are subtypes of Employee. The check-in and check-out relationships associate the Employee and the Asset. These relationships are actually implemented as client applications. The remote cloud represents an external event because the asset is transferred to a remote repository that has its own data model and processes. The remote repositories will have their own data model diagrams dependent on their roles. How the assets in the destination repositories are updated depends on the value in the Repoid attribute.

When the employee checks in an asset, the Creator attribute of the asset data model will be assigned the value of the EmployeeID attribute, thereby showing the relational association between the entities Employee and Asset.

When the initial ingestion of the asset occurs, the technician is the creator and decides whether the asset needs to be versioned according to business policies and type of media. To update an asset with rights and approvals, the rights and approval editor checks out the asset and uses it to determine the appropriate rights. Upon check in, the rights and approval editor will add the rights and approvals data model to the asset, thereby assigning the corresponding attributes to the asset for the publishers to use. The publishers essentially go through the same process of check out and check in, and afterwards the asset can be queued for transfer to the broadcast-server repositories which completes the representation of the workflow using this data model.

Summary
Modeling data in the MAMS is arguably the most important task to undertake in the design process. The purpose of data modeling is to develop an accurate model of the client's information needs and business processes. The data model acts as a framework for the development of the new assets and provides the ability to extend assets during its life cycle. This paper has described the data modeling scheme for digital assets by discussing some types of attributes that are needed to describe the assets and give them functional meaning in the business workflow and end-user environment. Metadata provides an effective means for searching the digital asset repositories so that users can easily find stored assets. It also provides an effective way to manage workflow by enabling such mechanisms as versioning and asset transfer between business unit repositories.

References
