USING NEW ADI 2.0 STRUCTURES TO EVOLVE VIDEO-ON-DEMAND MANAGEMENT & SERVICES

Yasser F. Syed, Ph.D.
Cable Television Laboratories

Abstract

The new ADI 2.0 specifications allows for the separate handling and combining of the offer & title metadata, content, and its metadata. This becomes a valuable set of managing tools as catalogues get larger, new VOD marketing techniques are envisioned, and content becomes more long lasting and less traditional. This paper will first describe what are the new ADI 2.0 structures and compare this against older ADI 1.1 structures. It will also describe how VOD services are changing and what does it imply for new management and service needs and how this will be easier to handle using the new structure & backend distribution. Then it will talk about possible ways to transition from a 1.1 ADI structure to a 2.0 structure. Finally the paper will describe how the new ADI 2.0 structures can help meet these new types of demands on cable systems.

INTRODUCTION

Traditional VoD evolved as a replacement of a Pay-Per-View (PPV) model comprising of a selection of top 20 movies that are replaced every 2-3 weeks. Initial trial deployments as far back as 1993 were created to see if it could replace the PPV model. But really it was the establishment of digital cable networks that gave VoD a platform to grow on. Since then, VOD deployments among cable operators have moved beyond the trial stages towards becoming a viable digital cable service product. It is one of the differentiators between cable services from other media service types.

Bandwidth used for Pay-per-View (PPV) services in time will be recaptured by VoD services while still retaining the value offered by PPV – current top twenty movies. But the model for VOD service can extend beyond this to carry TV shows, news, music videos, and other events found on broadcast while including possible commercial support and channel identification. This can be further extended to offer content with a continuity not offered on broadcast because of niche demands and limited bandwidth: viewing past show episodes, taking educational classes, seeing telescoping ads, touring local homes for sale, etc. The initial success of VoD bring challenges of its own that need to handle these new possibilities while scaling to increased customer usage demands.

The Cablelabs 1.1 ADI/VOD metadata documents is the current defacto specification used for delivering content with metadata to cable systems via satellite data transport or by tape delivery. Its intended purpose is to deliver a movie, its preview, and its boxcover along with metadata for a single video title offering in a one-way delivery system (see figure 1). This is the main delivery mechanism from a content provider or distributor to a cable headend or regional cable distribution center. With constant requests for new ways to offer Video content, this one logical structure has been refitted constantly to meet new VOD service, technology, and process demands; but at a considerable operational cost. This approach becomes harder to maintain as there is more content, more ways to offer content, and shorter lead times for distributing this material.

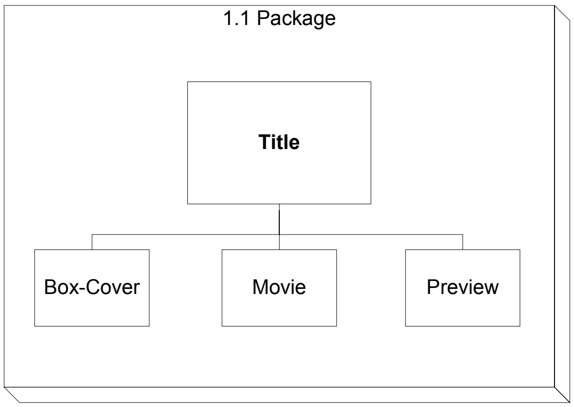

Figure 1: VOD 1.1 Structure

The new ADI 2.0 specifications create an approach to building a VoD structure more flexible to these new demands. The ADI 2.0 specifications provide a way to create more than one type of logical structure while creating a common approach to managing the delivery of these structures. The delivery mechanism has expanded to include multi-distribution approaches, more automation, and increased validation. The additional improvements in ADI 2.0 specification allow for more handling of large-sized content files and the ability to utilize this content in more than one VOD offering.

ADI 2.0 SPECIFICATIONS

The ADI 2.0 structures specification defines the basic building blocks and connectors that can be used to build logical structures for any type of VOD offering. It also describes the message document envelope needed for delivery of this information.

Assets

The basic building blocks to represent logical VOD structures are called assets. Each asset at a minimum has metadata to provide a universally unique identification (Provider ID/ Asset ID), versioning within the asset (updateNum), and management (Asset Lifetime window -- which determines how long the asset can remain with that cable operator). From an asset management perspective, the metadata is necessary to be able to track, manage, and ultimately purge assets from the cable system. There are three general types of assets: content assets, group assets, and metadata assets.

The content asset (CA) contains the content file (or link to the file) and the metadata that belongs with the content. Examples of this could be a video (movie, preview), still-image (box-cover), or even audio. This content asset by itself has no context, meaning that it is not yet associated with a particular VoD offering. This has advantages because the content asset can then be possibly shared with several VOD offerings at the same time. For example a preview could come in with a set of other previews in a barker offering and still be reused as the main preview for its movie. Furthermore it allows the option of separate transport and ingestion of large Giga Byte (GB) content files into the cable system on a time schedule different than the rest of the offer metadata. The metadata that goes along with this asset can indicate the size of the content, if it is corrupted, and whether it is completely built. Other metadata that is contained within the content asset are shareable metadata to describe the content but non-specific to the offer. An example of this would be Screen Format as opposed to BillingID.

Actions are done on the content assets through operations. This replaces the verb structure described in the 1.1 documents. “Accepting” a content asset brings the asset into the cable system as an unattached asset. “Destroying” the asset removes the asset from the cable system by effectively ending the content asset’s lifetime. “Replacing” the content asset can update the content metadata or even swap the content file though this is not often done except to recorrect existing corrupted files.

The group asset (GA) defines the context to use a collection of assets which could be a VOD movie, a movie on SVOD, a barker loop for new action movies, or whatever people possibly define. It is the organizing asset that can group and identify everything including other assets collectively specific to that context. This type of group asset has a flag to indicate itself so asset management systems can identify this as an organizing asset. This group asset can also contain its own metadata that is universally applicable for other assets it organizes.

Since the group asset is the main organizing asset indicating context, it needs to be initially ‘Opened’. Other operations for the group asset is ‘Dropped’ – the GA and its member assets are removed from system, ‘Replaced’— information in the group asset is replaced, and ‘Closed’—a hint that no more metadata or assets could be added to the GA context.

The last building block is the metadata asset (MA). It needs to belong to only one specific instance of a group asset and cannot exist on its own. It basically suborganizes a set of metadata that may be specific to only a subset of the assets organized. An example of its possible use could be releasing a title in a high definition format a week ahead of its standard definition format release. In this case, a movie metadata asset could be created for each version containing the licensing window as well as other metadata specific for each version. The licensing window metadata cannot reside at the content asset level because it is metadata pertinent to that particular offer. An interesting aspect of suborganizing metadata into different assets is that a different provider can potentially create each asset. This can allow for 3rd party contracted contributors authorized by the provider of the owning group asset. Contrary to this, the 1.1 specifications restrict all metadata and content to originate from the same provider.

The metadata operations are in reference to the particular owner group asset. The metadata asset can be ‘Added’ to that group asset (once the group asset is ‘Opened’ of course). The metadata asset can be ‘removed’ from the group asset. Lastly the set of metadata in that asset can be ‘Replaced’ as a whole for updates to the metadata asset. The metadata asset is always subjugated to the existence of the owning group asset which means once a group asset is ‘dropped’ all its member metadata assets are removed as well.

<u>Connectors</u>

In the 1.1 specifications, an implicit structural definition is created that restricts the logical structure to be only one general type: namely a title with a movie, box-cover, and preview (plus add-ons). This is fine for delivering a top-twenty movie, but was a rough fit for delivering other types of content or title offers. The ADI 2.0 specifications create an explicit structural definition where the relationships between the assets under certain rules can be redefined to suit the particular offer type. (see figure 2)

The group membership connection was previously talked about in the paper when discussing metadata assets. This is the only type of membership relationship allowed. It limits the metadata asset to be a member of 1 group asset instance and prevents the MA to exist outside of that context. Reasons for this is to tightly connect all elements of the offer together such that it can be managed as a single entity. This allows for example a complete deletion of an offer by simply dropping the organizing group asset without needing to see how the changes affect other similar offers.

The second logical connection tool is the reference pointer. Because of the acceptance of the provider ID / asset ID as a universally unique identifier, assets can be tracked and be referenced by this identifier. The one-way pointer reference can be embedded in the metadata of the parent asset and can be visible to the asset manager for validation or be kept hidden until the particular application processes it. This is the main method to logically connect the shareable content asset to the offer with a reasonable amount of assurance due to the near static nature of a content asset. This same reference connector could also be used between any two assets including outside group assets though the integrity risks would need to be minimized by proper logical structure development for the specific application document. By this mechanism, it is possible to apply second level of offers, create playlists from multiple titles, or allow for other ways to innovate in the future.

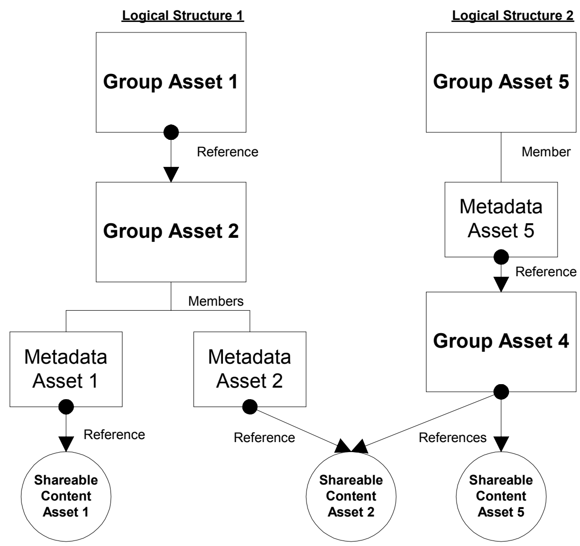

Figure 2: Explicit Structural Connectors

There also exists another connection tool (associate content signal) that is not required for creating the logical structure but does consider transportation timing issues for retrieving large sized content assets. As noted before, the content assets do not come at the same time a group asset is opened or when a metadata asset is added. Often they already exist in the system, but a copy needs to be retrieved as the offer is created. This requires preplanning to transport the file across sometimes very large regional networks. This connection tool does not create additional logical structures but ensures that the content assets are physically available to be connected with the offer.

Message Document

The message document is the delivery 'parcel' that arrives at the cable asset management system. The message document can contain operations and their assets or operations for existing assets in the cable system. Contrary to a 1.1 behavior, these operations do not have to be related to the same offer. This can allow an encoding house to pitch a batch of previews from one or even a set of content providers to cable systems. Also contrary to a 1.1 behavior, the offer has the option of being created from one message document or multiple message documents. (see figure 3)

Metadata for the message document are used for transport types of issues. It allows for "sender" to identify 3rd parties pitching the message document. It also provides metadata for a unique identifier for the message document (docNumber), creation time, relative ingest priority, and contact info for transport link issues. An exception to the metadata being used just for the message document is an acknowledgement field. The acknowledgement field provides an http connection to report back basic level

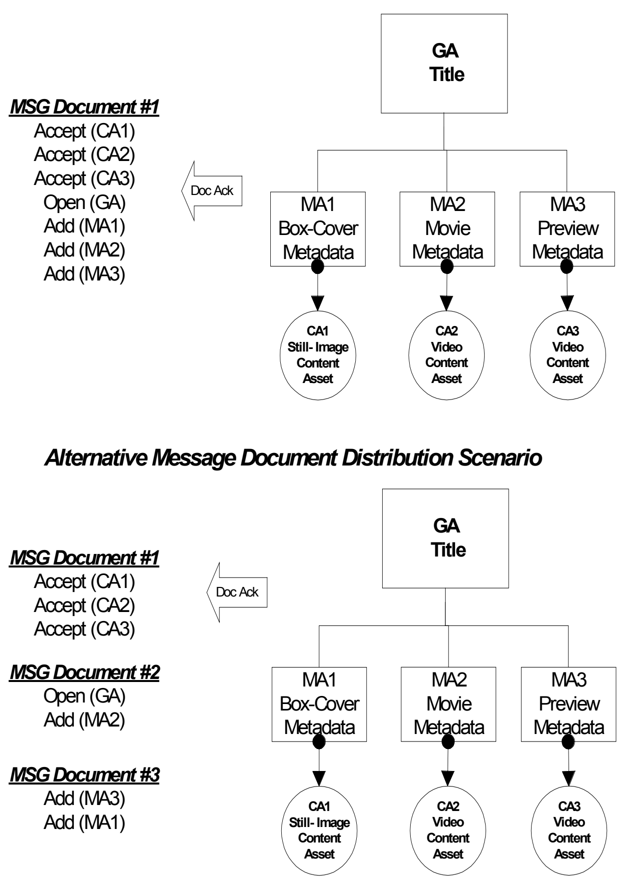

Figure 3: Single and Distributed Approaches to Create a Structure

validation on operations and assets enclosed within the document. This is an advantageous point to do this type of checking because it is the first point of ingest into a cable system. The basic level checking of the operations and assets are through message level xml parsing and validations mechanisms as well as content file integrity checks. Note this is not a basic validation of the organizing context but a basic validation of the components sent in the message documents. Validation of the offer happens at the application level where the logical structure is fully visible.

This separation of the message level and application level metadata allows for the message layer to be versioned independently from the application level. The asset manager can manage and route assets intended for applications that it need not be fully cognizant of. By allowing for this separation, the same message document layer could be used to deliver components for a VOD application or for an advertising server. Furthermore the use of versioning on namespaces also allows xml support of a mix of older/newer versions of application and delivery systems. This allows for many on-demand applications to be developed and supported using the common platform described in this specification.

The message document format can also be used to deliver auxiliary information to the cable systems or back to the provider. An example of this is the return ACK message (call it an ACK document) sent back upon receipt of the message document containing operations and assets (call it the ADI2 document).

This concept is extensible and has created a new specification called the ADI2.0 AIM (asset inventory messages) specification. This specification contains two types of message documents related to VoD, but do not create any new logical structures.

The first message document described is the Notification to Deliver Content (NIDC) which is a message sent from the provider to the cable operator informing the operator on a periodic basis what content assets (movies) are being planned to be sent. The cable operator can then use this information for asset management and resource management of their servers & systems. This is especially useful as content catalogues increase and become more long lasting in cable systems.

The second message document described is the Provider Notification Message (PMN) which is basically another acknowledgement but this time on a content asset level instead of a message document level. This can be used as an event mechanism to indicate the condition of a content asset as it is further processed in the cable system.

ENABLING NEW MANAGEMENT & SERVICES

By creating a common platform to build and deliver logical structures, the ADI 2.0 specifications can assist addressing many of the concerns in transport of content and management as Video-on-Demand scales up in volume as well as new services.

Delivering content files in a large cable environment often takes time because of the transfer of large video files (in GB) across an increasingly distributed cable network. With the advent of High Definition Video, increasing the size of content catalogues, shorter delivery lead times, more demand for content storage & management, and reuse/repackaging of content; it is becoming increasing critical to minimize the number of content file deliveries and increase the robustness of the transfer process. In this new specification, file integrity information (filename, filesize, checksum, metadata flag), has been moved to ADI level metadata to allow non-application routers, servers, and ingestion points to check for content file integrity during transfers without need to understand the application context. A message-level ACK can indicate as soon as possible in the process if the content files needs to be resent. Another content-level ACK is used to monitor the progress of file ingestion as it propagates through the cable network. For more distributed systems, the associate content signal and the content URL is useful to assemble the content components with the logical structure in time for it to be operational. In terms of shelf space management, the NIDC function gives prior knowledge of upcoming content storage demands. Lastly the content files themselves are shareable to allow them to be repurposed for other offerings and services instead of requiring the content to be resent for each new purpose.

Asset management functions benefits from the ADI 2.0 specification as well. The management of the asset lifecycle process is improved by separating out an asset lifetime window from the licensing window. This allows for situations such as having an asset exist in the system but maybe not “in-service” at that time. The specification also creates a clearer demarcation between message document functions to deliver asset components and higher-level application responsibilities. This aids in the health and fault monitoring of the delivery to headend transport network separately from logical structural issues. Also reference links can be defined that need to be recognized by the AMS to do things like validate that the referenced asset pointed to exists in the system. Alternatively, a reference link can be defined that only is processed at the application level and in the context of that specific application processing it. Lastly, shifting to a schema-based format allows for more specific XML parsers to be used to validate elements and specific value formats of them. The acknowledgement function in the ADI 2.0 specification improves the initial 1-way pitch by indirectly providing a first level success/fail response for the pitched message document.

The new structural tools developed in the ADI 2.0 specifications can enhance at the application level the value in how assets can be offered. In 1.1 initially, the offer, title, and content were pitched and identified as a monolithic group. With the break-up of this group and new types of structural concepts introduced, it becomes easier to offer content in new ways. For instance dynamic offers can be created fairly quickly from existing content assets in the system. Examples of this are: a weekend choice of discounted films, an updated barker list for an action movie folder, a broadcasting TV show that can be offered on VoD just as it starts, coming attractions that can be updated on all VoD movies, or VoD that can be offered with or without commercials.

A new concept that can be utilized is the standardized use of a playlist which can determine the continuous play order of a list of assets. This can be used for things like sports highlights, newscasts, double features, barkers, and other things. With the creation of the metadata asset, there is an ability to have 3rd party contracted contributions as a part of the group while still retaining an overall owner for the group. For instance a music soundtrack for a movie supplied by the record company could be part of the VoD offering. Also encrypted and trick files could be added to the group asset as additional content assets that can vary depending on the VoD system that it is sent to. With these additional tools and flexibility, the variety of possible ways to offer and organize VOD content material enables more variety of VoD types of services.

TRANSITIONAL CONSIDERATIONS FOR 1.1 STRUCTURES

The ADI 2.0 structure does not create a Video-on-Demand application. It provides the asset structure tools, and delivery transport mechanism to create an organizing asset structure for this. To create an actual VoD application based on the ADI 2.0 platform, one or more group assets, metadata assets, and content assets and their connections need to be defined. The metadata elements residing in each of these assets also need to be defined.

In creating the VOD 2.0 application for real world purposes, the existence of 1.1 structure and compatibility needs to be considered at least for a while. The VoD 2.0 structure may initially be restricted in its instance to accommodate a 1.1 transitional mechanism, but the structure itself should be flexible to allow for more this. An example of additional features could be support of a playlist to automatically update coming attractions. Another example of this is to support alternate versions of the movie within the same offer. Since 1.1 VoD systems and 2.0 VOD systems would both need to be supported at least for some time, this restricted case needs to be considered in the structural design of the 2.0 application during the transitional phase.

It is important to know that VoD 2.0 systems could still at the same time receive VoD 2.0 group asset structures that do not consider 1.1 backwards compatibility. This would allow for more creative development of services to enhance the VoD experience. It would also give incentive for older systems to transition faster from a 1.1 environment.

An example of this could delivery of a collection of TV episodes that can be offered for one price for a set or an individual price for a la carte.

Some of the immediate new features demanded in VoD services could still be deployed in this transitional phase. With the mechanism of reference pointers, new logical structures can refer to other logical structures. Since this reference mechanism points to the unique identifier of the asset, this same reference could point to a 1.1 VOD title as well. This would allow for some new 2.0 features to incorporate existing 1.1 titles. For instance, dynamic category folders like an action folder can refer to a set of existing titles including 1.1 titles. Some of these titles can also belong in other folders simply through the reference mechanism. These types of strategies can ease the transitional pains of switching cable systems to the new ADI 2.0 platform.

CONCLUSION

This paper describes the ADI2.0 specifications. It defines general asset types and their relationships to create many different types of logical structures as opposed to a single logical structure defined for VoD 1.1. It also describes how a structure can be created through one or more message document deliveries into the cable system. These new mechanisms create flexibility to address some of the existing issues with delivery of large content files, management of VOD titles, and creations of new types of VOD offerings. It also addresses some of the scalability issues from a structural perspective as VOD increases in both volume and variety. Many different applications can use these same ADI2.0 specifications to create logical structures for there specific type of VoD offering. The most immediate application to be developed on the ADI2.0 platform will address the single title VoD offering that allows for a transitional phase with existing 1.1 VoD platforms.

REFERENCES

1. Cablelabs Asset Distribution Interface Specification Version 1.1, www.cablelabs.com
2. Cablelabs VoD Content Specification Version 1.1, www.cablelabs.com
3. Cablelabs ADI2.0 Structures Specification, www.cablelabs.com
4. Cablelabs ADI2.0 Asset Inventory Messages Specification, www.cablelabs.com
5. “VoD Victories: Be Careful What You Wish For”, Carl Weinschenk, Comm. Tecnology, Feb. 2004.
6. “Supporting Large Scale VoD Through Common Metadata”, Yasser F. Syed, Proceedings of NCTA 2004, New Orleans, May 2004.
7. “Game Planning For Bigger, Better VoD”, Karen Brown, CED, May 2004.
8. “Ingest & Metadata Partitioning: Requirements For Television on Demand”, Robert G. Scheffler, Proceeding of NCTA 2003, Chicago, May 2003.
9. “Video-on-Demand:Quality Shows”, The Hollywood Reporter, December 1st, 2003.