Grass Valley Group ContentShare Platform

Overview

Dave Slack, Program Manager

ContentShare is a registered trademark of Grass Valley Group. All other trade names referenced are the service marks, trademarks, or registered trademarks of their respective companies.

The Media Management Challenge

As facilities begin to transition from analog to digital and from tape to disk, many are realizing that the methods of managing material need to transition as well. The integration of videodisk servers is adding new operational efficiencies; methods of managing assets need to be upgraded to take full advantage of these advances.

In addition to the task of managing traditional video assets, many facilities want to incorporate other types of assets into a complete facility asset management system. These types of assets could include:

- high-resolution video and audio;
- low-resolution video and audio;
- textual information about the clip – such as date shot, editor, director, location, subject, etc.;
- edit decision lists;
- news scripts and rundowns;
- legal airing rights;
- traffic logs.

Most customers want a media management system that can incorporate all of their types of assets. The challenge lies in the fact that most applications already have management systems for their own media. These management systems are often highly tailored for managing types of media within a particular job function. You only need to look at modern automation systems or newsroom computer systems to see examples of this. In fact, many differentiate themselves by their ability to manage their own assets.

It would be unreasonable to assume that these could all be replaced by a single asset management system. The result would be an unwieldy and expensive system, or one that requires the applications give up abilities they currently have with their own management system. Further, few software vendors want to give up control over their own asset management to a third-party management system.

Since there is not a software package that can do everything, facilities usually use many different packages. However, integration of applications from different vendors is becoming more complex. Often, the lack of an integration solution dictates the creation of custom bridges between applications. This takes longer to develop, debug, and deliver. The cost has to be passed on to the customer.

For some, a system that works only within a single facility is not sufficient. Increasingly providers and organizers of content are being asked to deploy their systems on a national, and even global, scale. Many of these systems are not designed for use on this scale. These issues present challenges that must be faced by developers as well as customers.

ContentShare Platform Solutions

The approach Grass Valley Group has taken with the ContentShare platform is not to try to replace any of these applications or their ability to manage their own assets but to *allow*

applications to work together. Rather than create a large, monolithic database or asset management system, Grass Valley Group has created an asset management *platform.*

The platform is more that just a protocol or specification. It consists of active software components called agents and brokers. They allow application developers to create standardized interfaces to their products for the purposes of exchanging assets and information. Because these interfaces present a common layer to other applications, exchanging information between applications becomes standardized. Therefore, the applications can be easily integrated without requiring custom software. This approach allows applications to communicate with each other and share information and assets without requiring them to give up control over their own assets.

Many other asset management frameworks are based on the idea of a central database to which multiple clients are connected. This can work well in the case of a facility that is limited in its size and where all the client software on the system uses the central database as the primary storehouse of information.

Many organizations, however, want to bridge the systems of multiple facilities, or bridge departments within a single facility, such as the production and news departments. If each of these departments has a media management system that is based in a single server, connecting those departments requires some connection between the two servers. To maintain the client-server methodology one server would have to become subservient to the other. If possible, this approach of connecting servers in multiple levels of hierarchy becomes increasingly difficult. To avoid such a hierarchy, servers can be connected with a peer-to-peer architecture.

The ContentShare platform has, at its core, a distributed, peer-to-peer architecture. Similar in concept to the Internet, the platform allows applications to work together on a peer-to-peer basis rather than any one application having control over all the others. The departments using those applications can interact with other departments. Further, facilities containing those departments can interact with other facilities all on a peer-to-peer basis. This distributed approach makes it easier for systems to grow as the facility grows and allows the organization to expand without having to undergo a drastic change in architecture.

Open Communication for Asset Sharing

The ContentShare platform establishes a common language that all applications can use to find, browse, edit, and operate on information and assets. Each application also registers what kinds of assets or services it provides and how it describes those assets so that other applications know how to interpret the asset. Applications can access each other's assets without having to know how the other stores its data internally. This language is also extensible, which means an existing asset type can have additional metadata fields added without breaking existing applications. Even one step further, a new application can join a system that provides a previously unplanned type of asset without requiring the system to be redesigned.

Locating Assets

Applications can register assets with ContentShare as they create them or when they want them to be available. Registering assets makes them visible to other applications. Other applications can then search and reference these assets without having to know where they physically exist. This gives a location independent link to those assets.

Linking Assets

Applications can assign relationships between assets such as a low-res browse video that represents a hi-res clip that can then be seen by other applications. This allows the user to have a single, logical name for a group of assets that all belong together. When users are searching for material, these links allow applications to present material as belonging with others. For example, a video that is referenced in a news script or transcript could be presented when searching for either.

ContentShare Platform Architecture

The ContentShare architecture is based upon years of research at the Grass Valley Group Advanced Research Labs under a grant from the National Institute for Science and Technology. The focus of this research has been to apply a software technology known as the *agent architecture* to the challenge of asset management. The ContentShare platform consists of a number of components based on this work:

- the ContentShare messaging framework,
- System Agents,
- Brokers,
- and User Agents.

ContentShare Messaging Framework

The agents and brokers within the platform use a special language, or messaging protocol, to speak with other agents and brokers. This language is called the Knowledge Query Manipulation Language (KQML). It allows autonomous agents to share information and work toward solving problems in a cooperative manner. KQML provides the language to frame questions and sentences with expressions such as “tell”, “ask”, “recommend”, “broker”, and “forward”.

The actual sentence content is expressed in a language called the Extensible Markup Language (XML). The structure of the XML data is application specific – other applications can use this data through the support of the System Agents.

These standard languages are tailored for exchanging information in a defined manner and do not require knowledge of the type of information be built into the language itself. Without such an extensible language, the base framework would have to be modified every time an application added a new asset type to the system.

ContentShare System Agents

The collection of software known collectively as ContentShare System Agents is comprised of three components: the Advertise System Knowledge (ASK) agent, the Resolver, and the Agent Name Server (ANS). These work together for the purposes of application registration, asset registration, and security.

Application Registration

The System Agent known as ASK gives applications a place to register themselves. Registration includes stating the kind of assets they have to offer or what services they can perform on assets, and how they describe those assets. Maintaining knowledge regarding how each application describes its own assets is key to allowing them to exchange assets. This gives them the freedom to request an asset from other applications without requiring any knowledge of how that application stores its information. Without this, each application would have to write a custom interface to every other. Self-registration allows applications to join a ContentShare system at any time.

Asset Registration

The ContentShare platform allows applications to expose links between various types of media, within their own system, to other applications. A newsroom computer system, for instance, may have scripts that reference video clips. The links between the two can be exposed so that other applications can tell what video elements may be used by different scripts and to view a collection of video assets that are used by a particular script. This link exposition is not required, but is left up to the discretion of the application developer. The links in this case would be maintained by the application that created them.

The System Agent known as the Resolver serves two fundamental purposes. First, the Resolver can manage links, such as above, *for* the application. It can maintain links between different types of assets. In addition, the Resolver can also maintain a map of logically equivalent assets. For instance, if a low-resolution browse quality video clip is created to represent a higher-resolution clip, the two clips can be linked together under a single name. Although they are linked the assets can still be accessed and manipulated independently.

This linking enables a number of important features. Because the two are now linked together, an application can use the low-resolution media to represent the high-res media.

This is useful in searching. An application could choose to show this low-res video within its user interface as a method for refining a search for high-res media. It would prove particularly useful if the high-res media were not in an on-line format. Another example is a news script that references a number of video clips. The textual information in a script can be a rich source upon which to search. By having a link between the video clips and the script, users looking for video can search out key words in a script that point them to that video.

These links also become useful when it is time to clean up assets. The links between the script and video can be used to delete all the video associated with a particular script. If the video is used by another script, those links can be revealed so only the non-referenced material is deleted.

The individual assets that are linked by the Resolver do not have to be created by the same application. A high-res video clip, a low-res video clip, and a text transcript of that clip can all be created by different applications and still be logically associated by the Resolver. In fact, the creating applications do not need to know anything about the associations.

The second primary function of the Resolver is to provide a location independent link to assets through the use of a globally unique ID. Logically equivalent assets can be associated under a single ID – this is the name that is used by applications to reference those assets. This allows the application that owns that asset to physically move it without affecting others' access to it.

Security

When an application allows other applications to access its assets, security becomes a concern. This is one of the primary purposes of the Agent Name Server (ANS). The ANS

provides a method for managing security that bridges the existing password security systems of the applications. It also provides a framework for the addition of security to non-secure applications.

This allows the security systems of different applications to work together. Allowing them to expose their assets and know that the permissions they have put in place for users, groups, etc. are still adhered to.

Brokers

The broker is the fundamental interface that applications use to publicize themselves and their assets. Any application that wants to make its information or assets available to the rest of the ContentShare system needs to have a broker to represent it. This concept of having brokers represent applications allows the applications to communicate effectively with each other. Without this type of interface, an application would be required to have a custom interface to every other application.

There are a number of different types of brokers; each named according to what they are brokering.

Asset Broker (AB)

The Asset Broker is one of the most common in a ContentShare system. Its primary job is to broker the metadata (the data about the data) as well as operations on the metadata, belonging to an application. In most cases of broadcast applications, the data the metadata belongs to is a high-resolution video clip. This metadata could be anything including the name of the video clip, who the editor was, when it was created, airing rights, etc. Anything that can be used to describe the primary asset could be considered metadata.

The job of brokering takes place in a number of phases. The broker is the interface through which an application registers itself to the ASK. It must also register what types of assets or services it provides and how it describes those assets.

Once that is done, it is ready to be a broker for the metadata. Any application that wants to be able to give out information to others must have an AB to represent it.

Query Broker (QB)

The job of the Query Broker is to help other application's User Agents query or search into the data the broker represents. In other words, the QB gives other applications the ability to search into its owner's data without requiring the other applications to know anything about how that data is stored or searched.

Many databases already have an efficient method of searching the data within their own system. In this case, a QB can be wrapped around that search tool so that others searching into the database do not need to know how the search tool operates. A QB can also provide an interface to search engines providing cached searches.

If an application wants others to be able to search into its data, it needs a QB to help them do it.

Transfer Broker (TB)

The Transfer Broker specializes in transferring content. For example, any device that resides on a Fibre Channel network needs a TB to represent its assets. Those devices might include the Profile Video File Server, an Avid NewsCutter Effects Edit System, the Grass Valley Group Network Archive, etc.

The TBs communicates initially to establish that the transfer can actually take place. They also use a common method to manage activities such as transfer queuing and requests to set transfer priorities. TBs also provide a notification service for the status of the transfer.

User Agent

Applications use brokers to represent themselves for the purposes of allowing others access to their data. However, others do not go directly to that broker with their request for information they employ a User Agent. Any application that wishes to speak *to* other applications whether for searching, requesting information, etc., needs to have a User Agent. The User Agent speaks to and carries on conversations with the brokers that represent the other applications.

Simply put, an application needs a User Agent if it wants to communicate with other applications and it needs brokers if it wants to be able to respond with information of its own.

By creating agents and brokers to represent their applications, developers can drastically decrease the development effort necessary to integrate with other applications that also use the ContentShare platform. This platform allows developers to write to a single interface specification rather than create a custom interface for every other application with which they need to integrate. It also gives them a measure of abstraction from those other applications. A vendor can make changes to its software without requiring everyone else to change the method in which they communicate with that vendor.

More in-depth technical information on the ContentShare architecture can be found in the *ContentShare™ Technical Overview.*

Role of third-party developers

How, then, do all these interfaces get created? Grass Valley Group is developing the ContentShare platform components as well as brokers for the Profile Video File Server and the Network Archive system. In addition, Grass Valley Group is creating a utility called the ContentShare Explorer. This utility searches and browses material that can be found on a ContentShare system. Usually, this function is tailored to the particular application the operator is using. To that end, Grass Valley Group is licensing the source code of the Explorer to qualified developers. The source code can serve as an example of how agents and brokers are used, as well as shorten the time-to-market for developers. Licensed developers can take any portion of this utility and tightly integrate it into their own workflow.

However, Grass Valley Group cannot create brokers for every application that exists in our industry. Therefore, third-party software developers will play a key role. Because the brokers represent the application, having a complete understanding of how that application works is helpful in creating the broker. Grass Valley Group created a Software Developer's Kit (SDK) to assist developers in creating brokers for their systems. The SDK describes how to use User Agents to speak for their systems and how to use the ContentShare platform in general. More than a dozen software developers have already agreed to leverage the ContentShare platform with their software, and this list continues to grow.

Grass Valley Group set up two programs to support the use of the ContentShare platform: the ContentShare Licensed Developer Program and the ContentShare Platform Integrator Program.

ContentShare Licensed Developer Program

This program helps developers use the ContentShare platform to increase the value of their software. Participants in the program have full access to the latest SDK, in-depth technical information, platform development tools, advanced engineering support, and collaborative marketing.

ContentShare licensed developers would concentrate on the creation of brokers for their applications. These may include Asset Brokers, Query Brokers, and/or Transfer Brokers depending upon the types of services they provide. Grass Valley Group created the licensed developer program to provide the tools to create brokers. It also helps developers ensure that their applications and brokers work with a ContentShare system and to provides a mechanism to license the use of ContentShare technology for commercial use with their application.

ContentShare Platform Integrator Program

One of the key benefits of ContentShare lies in the reduction of development required to integrate a complete system. With a few standardized interfaces between applications, rather than many custom ones, system integrators with software knowledge can now build a complete system without complex software development. To help integrators, Grass Valley Group created this program. Its aim is to give the skills necessary to use the ContentShare platform to bring applications together for their customers.

Conclusion

The ContentShare framework is a revolutionary method to connect applications. It allows application developers to create standardized interfaces to their products for the purposes of exchanging assets and information. These interfaces not only drastically reduce integration complexity, but also allow developers to deliver an unprecedented level of integration to their customers. ContentShare capitalizes on a distributed approach to the asset management challenge that lets applications continue to maintain their own, highly-tailored methods for managing their assets while allowing these applications to work together as a single system.

For further information, please visit www.grassvalleygroup.com/contentshare