Connect-World
www.connect-world.com

A new architecture for asset management

by Paul Wilkins, Director of Solutions and Marketing, TMD

The first generation of asset management systems were designed somewhat monolithically. The TMD Mediaflex system was relatively sophisticated, in that it was built from 12+ modules which were combined as required.

Modern coding techniques call for functionality to be broken down into small pieces, micro services run in containers. This ensures that each is self-contained. It starts as necessary, and shuts down when finished, saving processor cycles and energy consumption. Should a problem arise in any particular process it is completely insulated from all others, so is isolated from the rest of the system.

A respected solution architect, Paul Wilkins has more than 30 years' experience in the broadcast and media industry, defining and designing complex workflow projects and managing the process through to successful delivery. During his career, he has worked both as an end-user and with industry leaders including Sony and Avid. Most recently he has successfully delivered complex projects for international broadcasters and media organisations, including deployments of TMD's Mediaflex® asset and workflow management system.

In his role at TMD, Paul Wilkins is responsible for customer requirements capture, workflow definition, solutions sales and developing the company's market proposition.

Asset management started out as an unfortunate necessity. Once we stopped storing content on tapes and film cans and put it onto servers, we needed to know where everything was.

To help us find the content, we added some descriptive metadata to make searching for content more powerful. Then we added some technical metadata, so we knew whether we could use the content straight away or if we had to process it. But essentially, the first generation asset management systems were simply databases.

The big leap forward was the realisation that, if the metadata existed in our computerised asset management platforms, then it could be used not just to inform but to control. We could develop workflows that queried the metadata, made decisions on what it found, and updated the metadata when a process was completed.

To give a very simple example, when a new piece of content arrives in a system, you need to know if it has passed the quality control stage. An automated workflow could query the metadata, asking if it has passed QC. There are three possible answers: it has not been to QC; it has been to QC and passed; or it has been to QC and failed. As a result of this query the workflow will branch in one of three ways. In a software-defined architecture, it can make this sort of decision without the need for operator intervention.

Broadcast workflows no longer need be defined by sequential processing by discrete pieces of equipment. As software-defined architectures become more extensive, we can develop much more sophisticated workflows. Ultimately, these will allow broadcasters to state what is required in terms of outputs, letting the underlying technology do what is necessary to achieve it.

Automated workflows improve operational productivity, by giving repetitive tasks to machines and thereby releasing staff to perform creative and fulfilling work. As a guide, people are really bad at dull and repetitive tasks because they get bored, whereas computers are really good at doing precisely the same thing time after time.

Decision-making

You can look at decision-making in two ways: those questions that can be answered by facts, and those answered by opinions. If it is an opinion – does this content meet our watershed guidelines, for example – it is an engaging and stimulating job for a human operator. If it is a fact – has this piece of content passed QC – then it is better done by a machine.

If it is a fact it can be described in metadata. Remember that the fact might be that a human operator has checked and approved something. And if it can be described in metadata, then downstream it can drive automated workflows. This allows rich workflows to be created, for instance querying the rights management system to discover which platforms a piece of content can be published on, with perhaps a priority system to push content through transcoding farms to get popular content online faster.
A broadcaster might want its main evening news packaged and online within a few minutes of its live transmission. But the automated workflows would need to check if all the content can be put online: perhaps a sports report included footage from another broadcaster which can only be shown on air. So the workflow from broadcast to online has lots of decision-making points.

Services and tasks

The first generation of asset management systems were designed somewhat monolithically. The TMD Mediaflex system was relatively sophisticated, in that it was built from 12+ modules which were combined as required.
Modern coding techniques call for functionality to be broken down into small pieces, micro services run in containers. This ensures that each is self-contained. It starts as necessary, and shuts down when finished, saving processor cycles and energy consumption. Should a problem arise in any particular process it is completely insulated from all others, so is isolated from the rest of the system.

TMD has re-implemented Mediaflex on these modern design principles. Our architecture builds these small packages of software into "services", which become the building blocks of the functionality. We call this architecture UMS: unified media services, and talk in terms of media services applications running on our UMS platform.
It breaks the full functionality of a computer system – managing the metadata to trigger automated workflows and asset management – down into a series of small, clearly defined processes. These services can then be grouped into tasks, and tasks into workflows. When you set up a workflow, the controlling software layer calls up just the services required, spinning up the functionality then releasing it.

The data is passed from one service to another in a standardised form. It means that each is complete and self-contained, so the whole platform becomes much more resilient. It is also readily scalable because more services can be spun up as required.
It also allows configuration at the design level. A standard configuration might include a watch folder for arriving content. If your particular implementation needs more than one watch folder, then simply specify more instances of the watch folder service.

By configuring the tasks you need, it is also practical to develop targeted systems which are cost-effective because they are restricted in capabilities. The TMD Aperture baseband ingest system, for example, needs the encoding and transcoding task, but not the hierarchical storage task. Paragon, designed for archive management, needs the hierarchical storage but not the encoding tasks.
This architecture makes it easy for third parties to add their own functionality. Either they develop their own apps for the UMS platform, or they provide their API to allow TMD to develop the service as a core part of the structure.
At TMD we are good at asset management, workflows and integration, and we freely acknowledge that other vendors are equally good at related technologies in the workflow. We have done extensive work with companies like Telestream, for example, to provide transcoding without the operator ever being aware that the content has left the TMD environment.

Language

If it is to be an open system, then the message bus which communicates instructions from micro-service to micro-service must be standardised. We use AMQP, the advanced message queuing protocol, a widely-used open standard for passing business messages between applications or services. It is designed with the specific purpose of connecting systems, feeding business processes with the information they need.
AMQP, in turn, has standardised directories of message formats, which can be implemented to achieve the workflows we require.
The applicability of AMQP across the IT industry is important for two reasons. First, broadcast infrastructures are increasingly becoming software-defined architectures running on COTS hardware, and just as we benefit from the IT industry's vast R&D resources for hardware and operating systems, so we should take the benefit of robust and readily-scaled protocols like AMQP.

Second, and perhaps even more important, we must recognise that the ultimate end-game of automated workflows is that it becomes a part of the enterprise management system. In effect, a well-designed system should allow someone using SAP or other enterprise resource planning system to initiate new workflows expressed in business terms. Further, they should expect, as part of the integration, responses which clearly reveal the cost of providing that workflow or service, in order to make commercial decisions.
So a messaging system which is extensible outside the Mediaflex-UMS environment simply makes it faster and more reliable to create these enterprise level analytics and audit trails.

An important part of the re-engineering of Mediaflex into Mediaflex-UMS was the recognition that the cloud will be an extremely important element in service design. So the architecture is designed to support implementations on premises, in the cloud, or a hybrid combination of the two.

Simplicity

The advantages of Mediaflex-UMS can be summarised as flexibility, scalability and ease of communication. Systems can be built, in a modular fashion, which meet specific needs, confident that the scaling will have no impact on performance.
As an example, OnPoint is the media services application for production asset management. It includes all you need to build wish and make lists, drive the post production process and manage deliverables to broadcasters, but because it does not need a lot of the heavy-lifting workflow tools around rights management, ingest and multiple transcodings, we are able to offer it as a cost-effective option.
Similarly we have Paragon as a dedicated archive management media service, which provides the ability to migrate content from nearline to archive, using any storage solution including LTFS LTO robotic libraries and MAID3 disk systems.

These are dedicated packages which are easy to implement, but they are also very clearly TMD Mediaflex products. They can be extended to larger systems should the business needs change, want to add Archiving to your post workflows? then simply plug Paragon into your OnPoint application. A service-oriented architecture like Mediaflex-UMS can be fully virtualised, so it can run on premises, in a data centre or in the cloud, and it can scale from small to virtually infinite: there are Mediaflex implementations cataloguing and controlling more than a million objects.

The future of asset management is not about the metadata, but what you do with it. Efficiency and productivity depend upon automation wherever it can be applied, and that in turn depends upon accurate metadata. With the capability of tight integration into an enterprise resource planning system, the asset management should be regarded as a key commercial tool.