Commercial Content Production & Service Oriented Architectures

By B. Devlin and M. Horton

Paper supplied courtesy of AmberFin

Most of our working lives are focused on the daily business of getting the job done. We have limited time to think 'why are things the way they are?' but as we recover from a couple of tough business years, this might be exactly the time to pause and think about how we design and run Broadcast and Post commercial content production.

In this article we're going to throw caution to the winds and attempt to unify some of the current industry debates around commodity IT and Service Oriented Architectures (SOA).

Automation, standardisation and integration have driven down costs and improved quality in many industries. Although there are important craft skills in the media industry, some argue that many non–craft areas could benefit from ideas from other industries.

In part one, we'll look at the current evidence whether software–only products using commodity IT hardware will gradually eliminate all conventional broadcast and post products.

In part two, we'll explore the related debate whether SOA could eventually replace current broadcast and post system design.

Many AMWA members are already very active in these areas, so this article is intended to draw together key debates and present them to a wider audience.

Part 1: File Based Workflow And Commodity IT Products

It shouldn't be news to anyone that we are moving away from videotape or film as acquisition formats. P2, XDCAM, Silicon Imaging, Red, Arri Alexa and many other file based camera systems have become commonplace.

This raises the question 'if you start with files, process files in the middle and (increasingly) deliver your final content as files, can't software running on generic IT hardware do the job?'

The evidence is that generic IT hardware can do the job, in an increasing number of applications, although designing the right kind of workflow is important.

In the past, broadcast and post workflows usually involved videotape. Vendors naturally built traditional hardware products to sell into these traditional workflows. They used traditional control methods to integrate with other vendors – like MOS, VDCP, RS-422, GPIs and proprietary APIs. It worked, although it was expensive to buy and run (especially labour costs) and complicated to upgrade.

As computers and disk storage grew in capability, vendors (enthusiastically or reluctantly) built products that included computers and disk storage – but were naturally careful to continue to support traditional workflows and use traditional control protocols – like MOS, VDCP, RS-422 and proprietary APIs. It worked more efficiently than before but was still relatively expensive to buy and run and complicated to upgrade.

Today, many vendors now internally use of-the-shelf commodity computers and storage as product building blocks – but these are 'locked in' and vendors don't always offer customers or integrators the choice of buying the computers and storage themselves. Plus, the computers often can't talk to other computers in a computer friendly way. They have to use traditional broadcast control methods—like MOS, VDCP, RS-422 and proprietary APIs. Some vendors still argue:

"I don't believe commercial content production can be achieved using commodity IT products – it might look cheap on the surface but actually is hugely expensive. There are issues in integration, data management and many other areas which people don't really appreciate until it's too late"

Is that statement right? This is an important question for our industry, as we try to adapt to a new business climate. Let's take a look at the evidence:

Live event content production

Go into most trucks and production rooms and you'll still see traditional switchers, camera control units, replay devices, etc. We still need cameras and monitors too. Don't throw away your BNC crimping tool just yet.

Commodity IT based products have not yet proved they can handle all aspects of processing baseband video on the fly. But, some argue, this may soon become the only area not dominated by commodity IT.

Compositing and graphics content production

This is a specialist and labour intensive craft industry right. In the past, dedicated hardware products or turnkey SGI–based products dominated this market. Now, users mostly buy software only and use commodity IT Windows, Linux or Mac platforms. CGI is also dominated by commodity IT.

DI film and TV content production

Digital Intermediate (DI) at HD, 2k or 4k involves handling huge amounts of data. In theory proprietary products should have advantages over commodity IT (for example in performance) and it's certainly true that professional high volume DI Post houses have specific needs. However, if commodity IT is architected intelligently, especially in the use of multiple GPU processing, it delivers excellent price/performance and many successful companies already use this approach.

Long form content production

The editing and finishing market is now dominated by software only products running on commodity Windows, Linux and Mac platforms. Most of the vendors who previously sold turnkey editing or finishing systems now offer software only. This market is moving away from video to file based operation. This is true for everything from craft editing systems, through to workflow tools like Transcoders, standards converters and watermarking systems.

When it comes to creating the deliverables for this kind of content production (Broadcast, Web, Mobile, DVD etc.) some media companies already have file based 'media factory' workflows in place which rely almost

Where have all the traditional products gone?

Many companies have already moved to IT.

Storage systems at SBS in London

exclusively on commodity IT products to handle ingest, Transcode, review, format conversion, standards conversion, top and tail editing and watermarking. They are working today, this is not just a theory.

Stereoscopic content production (S3D)

A key requirement in S3D is to manipulate interocular values (the amount of S3D 'depth') on the fly without rendering. This is potentially an issue for commodity IT based products if they have to rely on CPUs. This issue has been solved by increased use of graphics cards (GPUs) which can deliver real time performance. This is good news as it brings down the costs of creating Broadcast and post S3D, allowing this new market to grow.

Designing workflows for broadcast content production

News, sports and 'close to air' programme production is currently a battleground between vendors with traditional products to sell and newer IT based players.

Because a lot of Broadcast content production is related to live content production, this may be where conventional products stage a 'final stand'. There is a huge legacy infrastructure of video routers, videotape, video disc recorders and video centric products. Even if all customers were to stop buying conventional products today, there will be a good business integrating this older world with newer off-the-shelf IT[1].

Many vendors offer turnkey systems, based on commodity IT but with qualified components to realise functionality that could not be achieved with generic IT components.

However, for newer 'green field' sites, the case in favour of commodity IT is looking more and more compelling. Broadcast content production is increasingly starting with files, processing files and delivering files. We've already seen how commodity IT products can handle most or all aspects of computer graphics, compositing, long form programmes, mastering, Digital Intermediate and S3D.

Broadcast content production which is not done 'as live' (and therefore doesn't require live baseband video manipulation) should therefore be possible using purely commodity IT technology.

One hotly debated theory is that Service Oriented Architectures — a design philosophy from outside our industry — might hold the answer to how that can now happen.

[1] Arguments against commodity IT in general and SOA in particular, often start with a valid fact but then draw an invalid conclusion. For example, a specific commodity IT based product may lack a particular capability compared to a conventional competitor. That doesn't mean that there is something inherently wrong with commodity IT – simply that there is a development issue with a specific IT based product.

Part 2: File Based Workflow and SOA

Service Oriented Architecture (SOA) is a design philosophy, not a product or a technology.

If you look up SOA on Wikipedia it will tell you that:

"Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services,which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format"

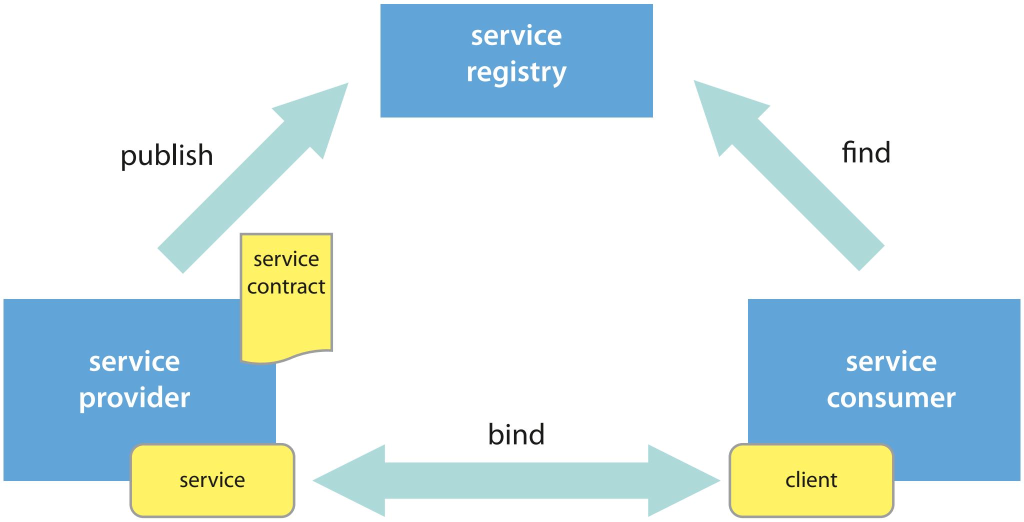
So, what does this mean in practice? One way to explain SOA design philosophy is to take a quick look at a practical example.

Traditional Tight Coupling vs. Loose Coupling

'Loose Coupling' means that units of functionality have little dependence on each other and the impacts of change are minimized. A loosely coupled design means that there is little or no dependency on the integration of product X with product Y. So, loose coupling can be good.

The opposite of 'Loose Coupling' is 'Tight Coupling' - products are integrated with each other in a narrow, specific and often inflexible ways. So, whilst it may be quick to get products working on day one of a project, any change to the business may prove to have expensive repercussions in terms of re-engineering work. Someone has to pay for that, one way or another. Tight coupling can be bad[2].

[2] In a large project which takes months or years to implement, the project requirements may have shifted or even radically changed by the time the project is complete. Making changes in mid-stream with tightly-coupled systems, even before the system is commissioned can be expensive, or even impossible.



SOA thinking will be utterly alien to traditional broadcast design, as the concepts of this diagram may indicate, but could hold the key to efficient use of IT

Let’s imagine we create a web page that will say: “Hello World”. If we use a ‘Tight Coupling’ approach, we will do it like this:

```
<html>
  <head>
      <title>News</title>
            <script type=”text/javascript”>function img(){
            document.body.background=”green_look.png”;}</script>
  </head>
  <body onload=”img();”>
      <table style=”width:auto;margin:1;background:#ccffcc;”>
        <tr><td style=”font-family:arial;font-weight:bold;”>Hello</td></
tr>
        <tr><td>Hello World</td></tr>
      </table>
  </body>
</html>
```

That all works fine for now - but then our business changes. We need an iPad version. We need to add a French language version. Our station’s look changes. So we need to change the web page business rules, structure, language and look:

```
<html>
  <head>
    <title>Actualité</title>
    <script type=”text/javascript”>function img(){
            document.body.background=”blue_look.png”};</script>
  </head>
  <body onload=”img();”>
    <table style=”width:400px;margin:6px;background:#ccccff;”>
        <tr><td style=”font-family:times;”>Bonjour</td></tr>
        <tr><td>Bonjour Tous Le Monde</td></tr>
    </table>
  </body>
</html>
```

Ouch! We need to make lots of *individual and specific* changes, which is time consuming when you need to update or change things.

This is a bit like what happens in broadcast automation when you try to replace one vendor's product with another and you need to bring in a team to sort out proprietary APIs. If we were to design the same web page using a 'loose coupling' approach, we would go about things in a totally different way:

- Structure is defined in HTML
- Styling is defined in CSS
- Logic in some suitable language (such as php, java, javascript, etc.)
- Page data and text stored in database

Although there is some additional initial thought needed in setting this up for the first time, change and expansion are easy:

- You can change language without touching styling
- You can change styling without touching structure
- You can add new content safely
- You can replicate and scale using existing IT tools (database replication etc.)

The SOA approach

A SOA approach achieves loose coupling by considering the service that a particular function provides for a business. For example, the service that a transcoder provides is to convert files that you have into files that you need. This has interesting repercussions:

- If the file that you have is already in the format that you need, then the transcoder doesn't have to do anything. A transcode could be the same as an FTP transfer. This is a *good* thing. If the business process doesn't actually have to do anything a lot of the time, then it is low cost and efficient.
- Converting the file that you have into the file that you need might involve a rewrap or decode then re-encode with a different codec. This is the traditional engineering definition of the word 'transcode'.
- If the file that you have is the wrong aspect ratio, resolution, frame rate, or is interlaced, then the transcode could become an aspect ratio conversion, an up / down conversion, a deinterlacer or a full motion–compensated standards conversion. From a business process point of view, these are all the same process, it's just the CPU load that changes.

Taking all the bullet points above into consideration, from point of view of a business process, nearly all the video processing functions that were traditionally called "conversion products" are now just a subset of transcode functionality. They all provide the same service to the media business.

Once a SOA approach is adopted then integration, data management and many other areas could become much simpler. The cost and complexity of automating processes and of integrating multiple vendor systems into a unified workflow could fall.

The difficulty with SOA is defining the services. There are many services that are obvious, such as transcoding[3]. Other services (billing, playout, media movement etc.) are a more complicated to define. As an industry we will get the benefits of scale if we can define a set of services that are common to several broadcasters, content owners and other media companies.

These common services, when accessed with standardised service interfaces herald the possibility of "plug & play" in the broadcast infrastructure. If the IT industry can allow us to plug in a USB disk that "just works" regardless of whether the disk comprises Flash, Spinning disk, Optical or some other exotic technology, then the broadcast industry ought to be able to make a transcoder or data mover "just work".

SOA and stereoscopic broadcasting

Early S3D broadcasting pilot schemes are often based around video–centric workflows: HDCAM SR tapes (or in the worst case two tapes), BNC cables and proprietary systems. That makes perfect sense for now, especially for live programme making.

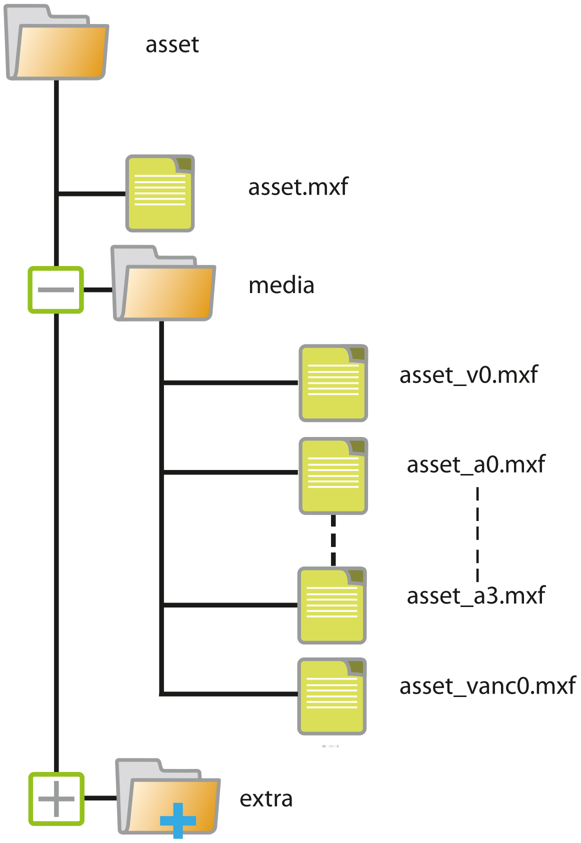
However, that is likely to be a temporary phase. If S3D broadcasting is going to become widespread, then it

[3] However the definition of 'Transcoding' is growing very broad. Today transcode systems may include a very wide toolset, such as ingest, top–and–tail editing, audio track mapping, audio track synchronisation, VTR emulation, text and logo insertion, aspect ratio conversion, format conversion and standards conversion.

will need to become file based, automated wherever practical and use the same commodity IT hardware as is the current industry trend. It's absolutely critical that S3D broadcast images are technically correct but it also needs to be commercially viable and video–centric working with proprietary products is likely to prove far too expensive. S3D broadcasting is not a 'special case' for proprietary products.

What's the best way to store S3D on disk? In an environment where it needs to be manipulated and subjected to manufacturing / publishing style workflows then MXF AS–02 looks like a good candidate. MXF AS–02 is effectively a loosely coupled way of storing the individual sequence of pictures and synchronising them with the audio.

It offers a robust and format neutral container system to move left and right eye images safely in a file based environment. There will also be a need to create an MXF interleaved OP1a style wrapper for applications where the content needs to be moved or stored with little or no modification. There is work in SMPTE now to develop this new approach.



The folder and file structure of MXF AS-02

FIMS: Bringing It All Together

AMWA and the EBU have come together to develop a Framework for Interoperable Media Services (FIMS). The goal is very simple — what services are required to create a set of "software building blocks" from which the industry can construct the range of facilities and infrastructures needed to meet the demands of the new file based world.

The project is concentrating on both the "Framework" in which the services must operate and a number of key individual services that must operate within the framework. The project is open to any person or company that signs the participation agreement and membership of AMWA or EBU is not a prerequisite for participation.

One area in which AmberFin has made a contribution is the definition of an abstract transcode interface. Whilst this may not sound like an earth shattering piece of technology, the implications of an agreed interface at this level are quite profound.

Today, a transcoder is usually thought of as a codec conversion device. If you want to "flip" MPEG-2 to MPEG-4 then you need a transcoder. If you want to flip QuickTime to MXF then you need a transcoder. This, however, is a very engineering-centric view of the world. If we look at the service that the transcoder provides, then we realise that a transcoder is converting files that the business has, into the files that the business needs.

This definition of a transcoder reflects more on the way in which a transcoder might be controlled rather than what it actually does. This definition of a transcoder results in the transcode API being a way of getting the right files to the right place at the right time, regardless of the processing required to get there. As we saw earlier in this article, some transcoders now have a very broad toolset — they can be mini 'media factories'. The API proposed by AmberFin goes one step further to ensure loose coupling and vendor independence.

We observed that many users of transcode devices referred to their profiles as "My HD profile is like the SD profile except we have 6 channels of audio" or "My online profile is like the iTunes profile except coded at 600kb/s". In fact the majority of profile descriptions were "like" some other profile. We also observed that prior to going into operation with a profile, there was a quality control step where it was verified that the profile would work in a given environment. This was often referred to as "on-boarding" a customer or deliverable.

Putting all this together results in an abstract transcode API centred around the business needs of the media company, where there is an implicit assumption that the users of the interface will QC a transcode profile, regardless of whose transcoder implements the profile.

The API can then be used on a job by job basis to make similar profiles by making defined vendor-independent changes to that profile. This allows vendors to differentiate themselves based on speed, picture quality, throughput or whatever else makes a difference to the product.

This also allows users to take advantage of any transcoder that implements the API without needing to implement custom code. This heralds the prospect of a true plug and play environment where a media company needing to increase capacity, increase quality, implement a new delivery specification, perform standards conversion or simply perform a technology refresh can choose the best of breed product and plug it into an existing environment. After all, as one customer told us—it's not rocket science, it's just a transcoder[4].

Conclusion

We have seen that almost all forms of commercial content production can be achieved using commodity IT products. We have also seen that issues in integration and data management may be solvable by ideas from outside our industry.

The media industry is in the midst of turning to file based working in an attempt to get control of costs in an environment where technology is forcing a deluge of new codecs, delivery specifications and technology changes (HD, S3D) upon it. This flurry of new delivery mechanisms does not coincide with a massive increase in the amount of cash available to deliver them. In fact the opposite is often true. Many media companies are forced to publish or "manufacture" their outputs in more formats this year than they did last year and to do it with a smaller budget.

No matter how good the media companies technology partners are, this trend can only continue if there is some consolidation in the media industry in terms of the number of codecs, wrappers, delivery specifications and APIs that need to be delivered.

AMWA within FIMS, and companies like AmberFin within AMWA, are key in driving these technology consolidations to ensure that practical, workable facilities that are customised to a media companies needs can be constructed from commodity, general purpose, interoperable software tools.

To find out how you can influence the work if FIMS so that it is right for your company, go to www.amwa.tv and follow the links.

[4] The same customer went on to tell the story of a friend of his who was actually a rocket scientist. His claim to fame was the amount of money he spent blowing stuff up before getting any designs to actually work. Hardly a good model for a cash–strapped media industry!

This white paper was supplied to the AMWA by AmberFin, Ltd.

Further white papers on MXF, AAF, XML, and SOA applied to advanced media workflows can be downloaded from the AMWA website at www.amwa.tv. 12/2010