INTEGRATING NON REAL-TIME SOFTWARE PROCESSES INTO REAL-TIME IP-BASED PRODUCTION

Paola Hobson

InSync Technology Ltd, United Kingdom

ABSTRACT

An all-IP studio production environment opens up the opportunity to create more powerful production tools for the live workflow. Seamlessly integrating software processes into the live environment provides production directors with access to a wider range of applications. These may include inherently non real-time processes such as time reversal, creative features such as content-related graphics, or complex effects generation such as content speed-up/slow-down.

In this paper, we use the example of a synthetic slow motion application to show how an inherently non real-time process can be integrated into a live IP production environment.

This work was carried out within the context of the AMWA Networked Media Incubator (NMI) project (www.amwa.tv/nmi/). The InSync synthetic slow motion software was demonstrated at a workshop organised by NMI leaders BBC, in which we proved interoperability with a reference system and with other vendors' studio equipment.

INTRODUCTION

The move to all-IP production has generated a lot of interest from broadcasters as a means of both saving costs and enabling innovation in production methods. Recent experimentation, such as that reported by the BBC (1), has shown that removing the constraints of SDI content transfer can lead to new production workflows which allow support for software-based applications. The use of software applications, rather than dedicated proprietary hardware, introduces greater flexibility in the features and functionality which can be made available to production professionals, and allows for more rapid upgrade and development of improvements and enhancements to such applications.

During the transition from SDI to IP-based facilities, content producers will operate in hybrid system architectures, where both native IP and legacy SDI equipment are in use. The hybrid environment is facilitated by provision of edge devices such as SDI to IP and IP to SDI interfaces, as shown in Figure 1. Such a system may incorporate software and hardware processing equally - as long as the required functionality is accessible and controllable, it does not matter how it is implemented.

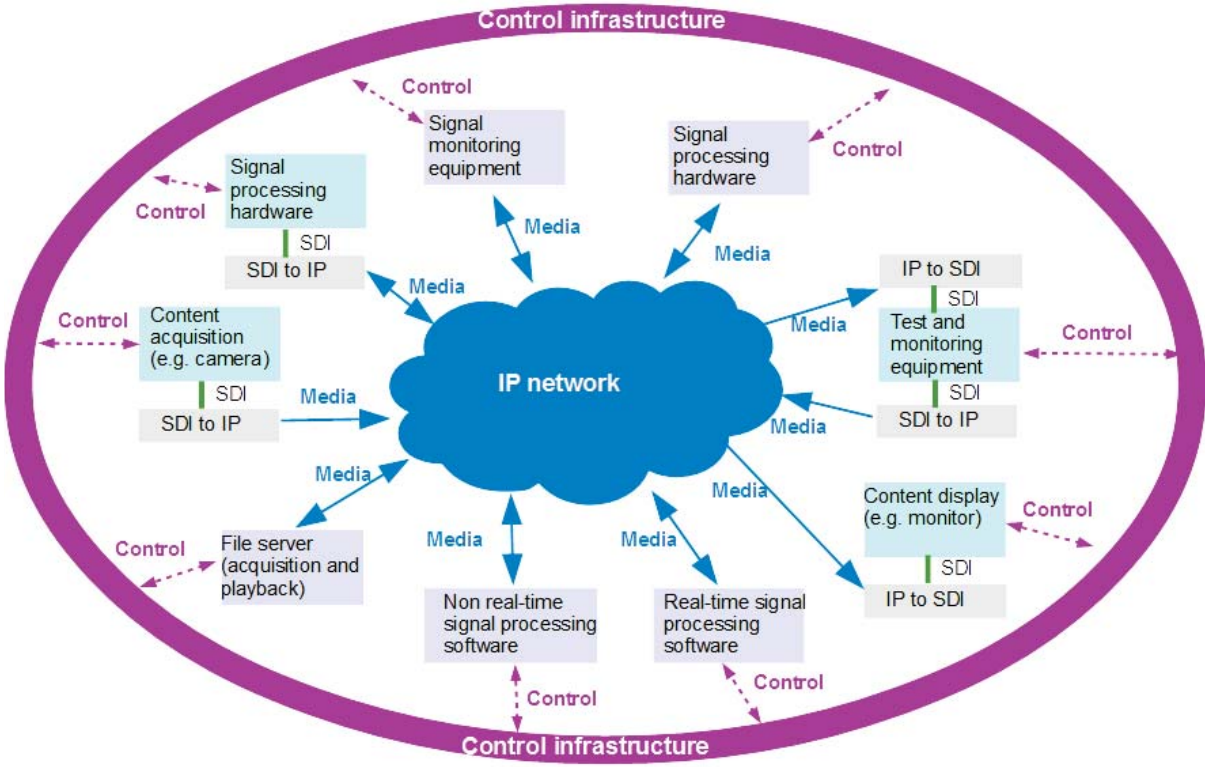

Figure 1 - Hybrid SDI/IP environment

In the example illustrated in Figure 1, we may imagine a hybrid production environment where hardware sources, such as cameras and file servers, stream real-time IP-encapsulated content via multicast to a media network, where other devices such as a vision mixer, standards converter or signal monitor, may be present in hardware or software.

However, integration of inherently non real-time processes into the live IP production environment does not immediately appear to be straightforward. Visually interesting processes such as time reversal, creative features such as content-related graphics, or complex effects generation such as time speed-up/slow-down, as currently used in live production, rely on a storage/process/playback paradigm in order to be integrated into the programme as it is produced. Provision of such functionality in software requires careful architectural thinking in order to be efficiently implemented and user-friendly.

Fortunately, within the AMWA Networked Media Incubator (NMI) project (www.amwa.tv/nmi/) an open specification was developed (www.nmos.tv/), which built on the Joint Task Force on Networked Media Reference Architecture, the use of which enabled InSync to integrate a fundamentally non real-time application (synthetic slow motion) into a live IP production environment.

The Networked Media Open Specifications (NMOS) are a family of specifications covering the various aspects of Discovery and Registration (currently AMWA Proposed Specification IS-04) and In-stream Identity and Timing which were implemented and proven within the AMWA NMI. More information about the NMI and NMOS may be found in (2) and (4), however, the aspects of NMOS which are relevant to this paper are the logical data model and the addition of relationships and time-based information to content and broadcast equipment.

The first phase of the NMI project focused on fundamental concepts of service registration and discovery, and connection of devices supporting uncompressed video and audio in a live studio environment. We provided an example non real-time application to the NMI (synthetic slow motion software), which could be used by any of the participating applications e.g. software editor, software vision mixer, content acquisition etc to extend their experience of new production possibilities.

In this paper we show how the draft NMOS specifications were implemented for a synthetic slow motion application. At an interoperability workshop led by the BBC, InSync was able to set-up and advertise the non real-time service such that other devices in a full-IP studio environment could find and request the service, route the required video essence stream to the service, and receive the synthetic slow motion results.

SYNTHETIC SLOW MOTION IN THE NMOS MODEL

Synthetic slow motion

A synthetic slow motion application was chosen as an example of a typical non real-time process which might be needed in a live IP studio environment. A common use case is in live sports production where certain important action is replayed at one third or one quarter speed to enable viewers to see the detail of what happened. In today's live SDI production workflows, slow motion is created by use of high speed cameras, which capture short periods of action at very high frame rates, then replay the chosen sections at the normal production frame rate, thus slowing down the motion. An example sports application, using synthetic slow motion instead of high speed cameras, is shown in Figure 2.

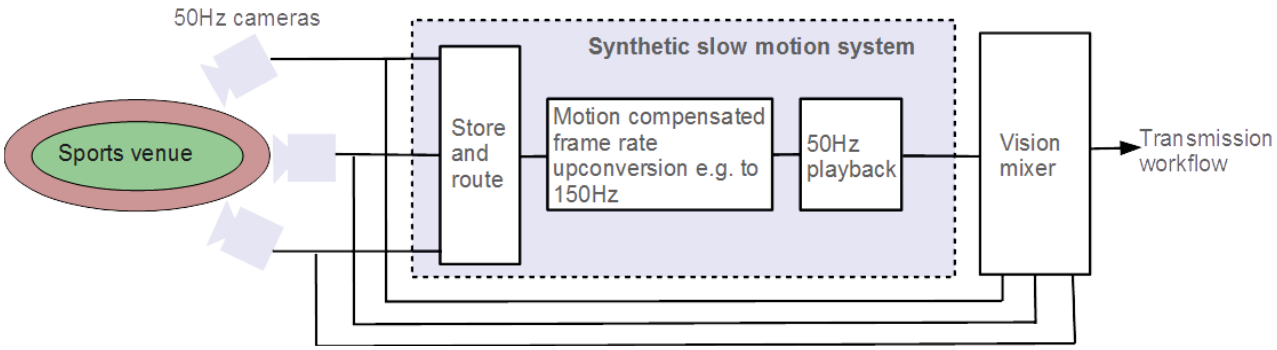

Figure 2 - Example application of synthetic slow motion in a sports production application

A synthetic slow motion process is essentially a frame rate conversion method, where the live action is captured at normal production frame rates, then the stored frames are interpolated to create additional output frames. If these are played back at the normal production frame rate, the motion will appear slower. For example, as illustrated in Figure 2, if 50Hz video is frame rate converted to 150Hz, e.g. via motion compensated interpolation, a 50Hz playback will result in objects moving at one third speed. Thus the output of the synthetic slow motion system can be fed to the vision mixer, and integrated into the programme, along with the live, normal speed cameras, as required.

Really good quality synthetic slow motion depends on the accuracy of the motion compensated interpolation, which has been reported elsewhere (3) and is not the subject of this paper.

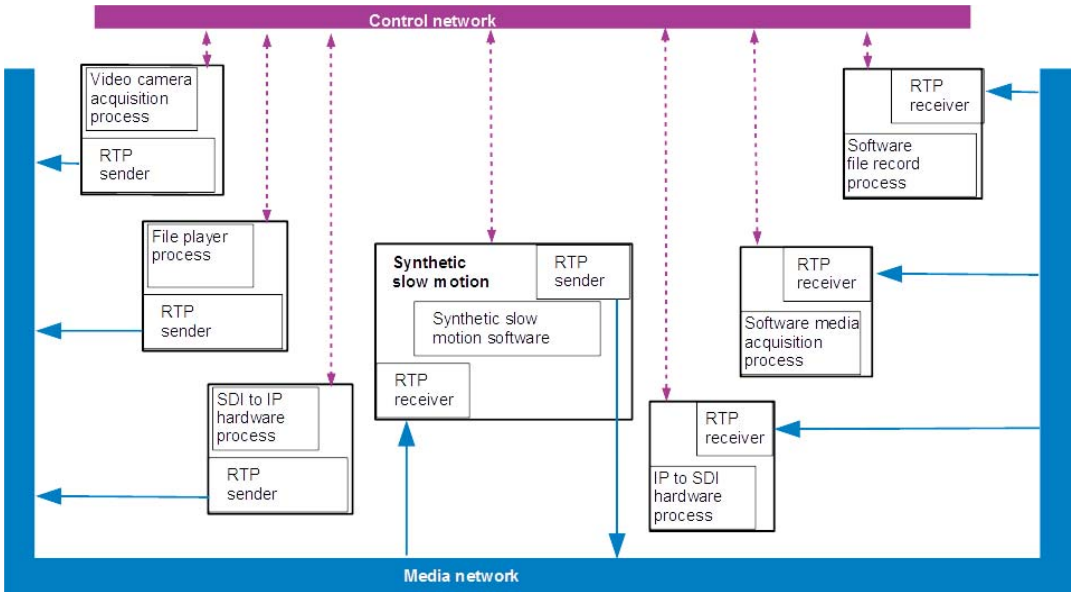

Figure 3 - Synthetic slow motion within an IP-based production system

As illustrated in Figure 3, a synthetic slow motion process may be integrated into a production environment where the source may be a live camera, a file playback device or any other source, and if the source is not native IP, an SDI to IP conversion is made. Since a certain number of frames are needed before accurate motion prediction can be made, and since the output playback time is always longer than the source playback time, the process is inherently non real-time.

The solution is for the slow motion process to essentially integrate a stream acquisition process and a stream playback process, so that incoming frames are buffered, and outgoing frames can be played back at the desired source frame rate.

However, that covers just the processing requirements at the centre of the slow motion process. For real usability of the application, it is necessary for the production system to find the application, initiate its use, route the desired signal to the application, and route the resultant slow motion video to the next device in the workflow, ready for broadcast.

This is where the Networked Media Open Specifications (NMOS) fundamental concepts of discovery and registration, identity, connection management and timing are used. A full description of the NMOS specification and its elements are available in (4), so will not be repeated here. At the centre of NMOS is a logical data model such that video, audio and data inputs and outputs, and control parameters are communicated via logical interfaces on a common network. Not only does this facilitate communication, it also enables multiple devices to run concurrently on the same physical host.

Application of the NMOS model

At the highest level, in order to be useful to, and interoperate with, other elements within the live IP production environment, our synthetic slow motion application (FrameFormer) had to:

- create a unique, persistent device identity on our specified Node (e.g. physical server) for each instance of FrameFormer which we wanted to make available to other devices on the network
- register and advertise our set of Node, Device, Source, Flow, Sender and Receiver resources
- correctly connect to other Senders and Receivers such that content can be streamed
- initiate, run and terminate the synthetic slow motion generation as appropriate to the content provided by other devices

A simplified overview of one example use of FrameFormer is shown in Figure 4. The first step is for the Node (in our workshop testing this was an off-the-shelf server) to discover the network registration service via mDNS. From there, our Node registers with the system registry via the NMOS Node API, and continues to maintain itself in the registry via periodic health calls.

For each instance of FrameFormer that we want to make available to other devices in the network, we register a unique, persistent DeviceID. The DeviceID includes a label which is human readable, which enables other users of the system to decide if this is a service that they wish to use. Since only one video stream at a time may have the synthetic slow motion applied, we may instantiate multiple synthetic slow motion applications on our Node, each having its own unique DeviceID. In this way, we may advertise on the registration server, for example, both a one-third speed slow motion and a one-quarter speed slow motion device, which will have separately identifiable receive and send logical interfaces.

An example of the return from a query of the available devices using the NMOS QueryAPI is shown in Figure 5.

Note in Figure 5 that the senders and receivers fields are null, which means that no other device in the system has yet requested a synthetic slow motion process. The next step is for FrameFormer to establish its RTP receiver (complying to RFC-4175) and RTP sender. These will be advertised in the registry as shown in Figure 6, which makes the application interfaces visible for other devices to connect to.

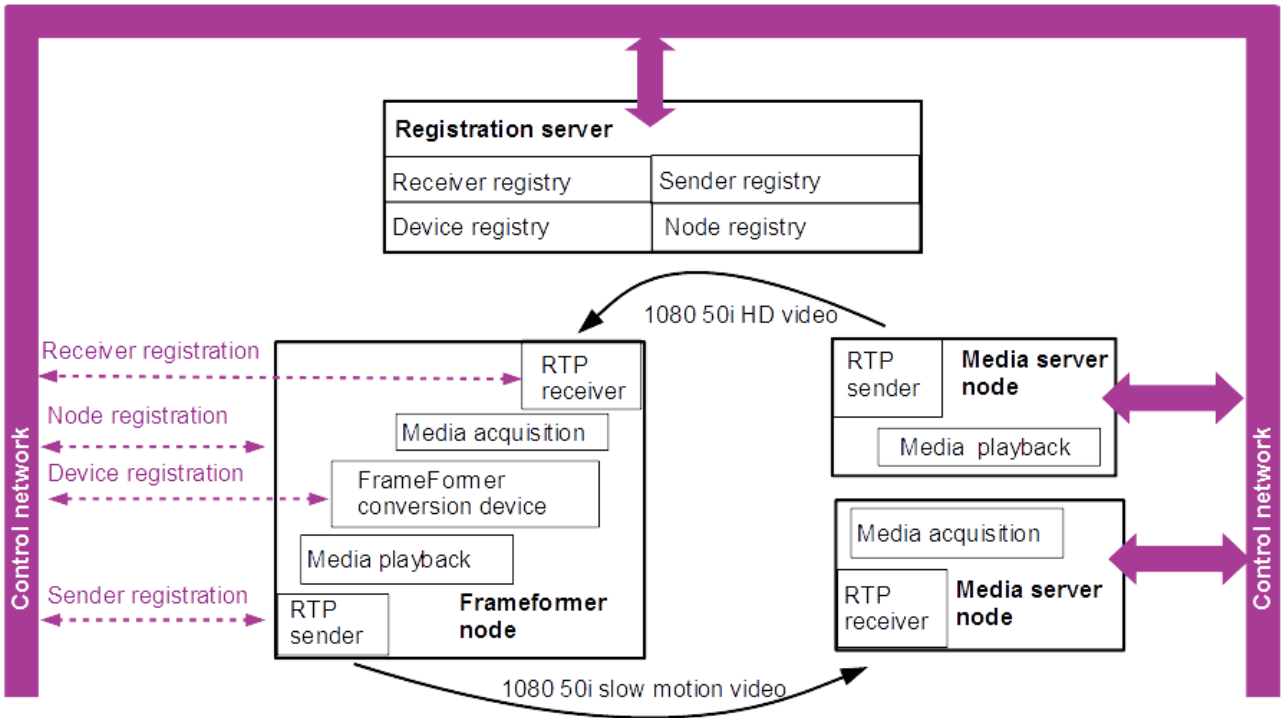

Figure 4 - Simplified overview of synthetic slow motion within the NMOS model

```
{
        "receivers": [ ],
        "label": "FrameFormer device",
        "version": "1452766253:909595792",
        "senders": [ ],
        "type": "urn:x-ipstudio:device:v1.0/frameformer",
        "id": "41500a23-500c-4307-b195-bbfd787673b2",
        "node_id": "64efcbd9-6059-4fc0-af83-1585455b8d92"
},
```

Figure 5 - FrameFormer advertised DeviceID

```
{
        "description": "FrameFormer video sender",
        "label": " FrameFormer video sender ",
        "version": "1452766253:909642887",
        "manifest_href":  "http://192.168.15.127:40001/x-ipstudio/node/v1.0/sdp/",
        "flow_id": "d4b5377b-0377-402e-b91d-90d673f9bc2f",
        "id": "02936e04-2c01-4aa9-ad30-c1c601afb869",
        "transport": "urn:x-ipstudio:transport:rtp.mcast",
        "device_id": "41500a23-500c-4307-b195-bbfd787673b2"
},
```

```
72  {
73      "description": "FrameFormer video receiver",
74      "tags": { },
75      "format": "urn:x-ipstudio:format:video",
76      "label": " FrameFormer video receiver ",
75      "version": "1452766253:909695041",
77      "device_id": "41500a23-500c-4307-b195-bbfd787673b2",
78      "caps": { },
80      "id": "794d42f9-16f9-4538-b8f6-e113b6a4417",
81      "transport": "urn:x-ipstudio:transport:rtp.mcast",
82      "subscription": { }
83  },
```

Figure 6 - FrameFormer sender and receiver

Using a synthetic slow motion application in programme production

Once the synthetic slow motion device is available for use, other devices can now establish communication and use the application. Referring again to Figure 4, the Media server node in the centre right of the diagram could simply be a live camera, which via the NMOS model, establishes an RTP sender connected to the FrameFormer RTP receiver. The camera will have registered multiple Flows, each of which represents fundamental essence relating to the content. For example, there may be one or more audio flows from the camera's on-board microphone, and there may be one or more video flows, for example an uncompressed and a compressed video output. The user decides to route just the uncompressed video flow via FrameFormer, so the camera SenderID in the registry will reference a specific FlowID which relates to the uncompressed video essence. The user must also select a destination (an RTP receiver which must exist on another device) to receive the slowed-down video. This may be a media acquisition device (e.g. a software file recorder) or could be a feed to be mixed into the programme going to air (e.g. via a software vision mixer).

Receiving a video flow is the trigger for FrameFormer to initiate its functionality. Firstly, the incoming frames are acquired and stored locally on the InSync node. When sufficient frames have buffered, the conversion to the required higher frame rate starts. Output video is stored until the clip has completed, then the slowed-down video is streamed from the FrameFormer RTP sender to the desired RTP destination (as illustrated in Figure 4).

Synthetic slow motion application testing

In a typical experiment, carried out during the NMI Phase 1 workshop, hosted by the BBC, a web interface created by the BBC was used as a router to map a 1080 50i video camera source to our FrameFormer receiver. When the RTP video packets were received, FrameFormer re-assembled the video frames, and when sufficient frames were buffered, conversion to the higher frame rate started. We used 250 frames in many of our experiments, since these represent 10 seconds of video which is typical for slow motion replay. Once the slow motion frames had been generated e.g. 750 output frames would represent one third speed for 250 input frames, we decomposed the video into RTP

packets which were then routed to a partner receiver (e.g. a hardware IP to SDI converter) and displayed on a monitor.

During the NMI Phase 1 workshop, the synthetic slow motion application was tested with a variety of NMI partner devices e.g. software file playout applications, hardware SDI to IP devices, and camera to IP reference streams. It was found that any device conforming to the NMOS specification and supplying a valid RFC-4175 compliant video stream could obtain the required slow motion process, and all NMOS compliant receivers could interpret and store or display the slowed-down video output from our slow motion process.

Our experience of the NMOS model in April 2016 was very positive and led to a number of suggestions for extension which would enable further applications to be added to a live IP-based production environment. Ideally, the DeviceID would allow advertising of more complex service capabilities which might be required by other applications. For example, another device might need to know if the synthetic slow motion service could handle a certain type of compressed video or a non-standard picture format before choosing to use the service. Our experimentation at the NMI Phase 1 workshop used only 1080 50i and 1080 59i video, and the standard used for any one slow motion process needed to be known in advance.

Furthermore, it would be beneficial if the NMOS model could allow control parameters to be passed to the application, to allow specific set-up scenarios, as well as dynamic adjustment to the device's functionality. So, for example, in our testing, the amount of slow motion was fixed to one third speed. Ideally, the device requesting the service would be able to pass certain control parameters which would choose the amount of slow motion at the point of initiation of the service (e.g. one quarter or one half speed). Such control parameters would be useful for other potential applications in the studio, e.g. choice of output frame rate and picture format for standards conversion, or request for other functionality within a chosen application e.g. video legalization, aspect ratio conversion, picture enhancement etc.

Finally, the addition of flow control mechanisms to the NMOS model was out of scope for NMI Phase 1, but is being considered in Phase 2. Such control mechanisms would, for example, enable the requesting device to signal the synthetic slow motion service to stop at a specific point in time. In the March NMI workshop, the FrameFormer service terminated when no more frames were received, but according to the prevailing NMOS model, this service could re-start whenever new incoming frames were received. A corollary to this was that it was not possible, at that time, to protect the receiving port from a competing sender. Therefore if an operator routed in error a new sender to FrameFormer part way through a slow motion process, the new frames would be processed instead of the desired frames from that point on.

CONCLUSIONS

In this paper we have shown how the NMOS model enables non real-time applications to be integrated seamlessly and easily into real-time IP production, using a synthetic slow motion application as an example. The NMOS specifications enable hybrid IP/SDI workflows to be readily implemented and managed, such that hardware and software products can co-exist in an efficient and effective way.

Through the interoperability testing at the NMI Phase 1 workshop, we proved how easy it is to offer a non real-time application as a service in a live IP-based studio environment,

thereby opening up opportunities for broadcasters to access exciting new production tools and methods.

Our experience in interoperability testing led to a number of ideas for extensions to the NMOS model which would enable a wider range of applications to be offered in the live IP production environment.

REFERENCES

1. Rawcliffe, A. 2015. Covering the Glasgow 2014 Commonwealth Games using IP Studio. BBC White Paper WHP 289. March 2015.
2. Brightwell, P. 2016. Future-proofing Live IP: an emerging roadmap towards an industry architecture. IBC 2016 September, 2016.
3. InSync Technology, 2015. Achieving synthetic slow-motion in UHDTV. InSync White Paper (http://www.insync.tv/documents/2015%20White%20Paper%20Slow%20Motion.pdf). August 2015
4. AMWA NMOS https://github.com/AMWA-TV/nmos February 2016.

ACKNOWLEDGEMENTS

The author would like to thank Dan Burgess and Brendon Allen for their contributions to the implementation. We also thank the BBC for leading the AMWA NMI project and hosting the interoperability workshops.