# Television on a Tablet – How TV will look like in the future and how to deal with professionally produced Content

Maarten Verwaest[1], Davy Van Deursen[3,]
Robbie De Sutter[2], Niels Laukens[2,] Dieter Van Rijsselbergen[3],
Erik Mannens[3] and Rik Van de Walle[3]

[1] Limecraft, Ghent, Belgium
[2] VRT-medialab, Ghent, Belgium
[3] Ghent University – IBBT, Ghent, Belgium

## 1. Introduction and Participants' Interest

Personal Digital Assistants (PDAs) and tablet PCs have been around for a while, but it took the development of a new type of operating system which efficiently replaced keyboards and mice by touch interfaces, before a real market could be developed. These devices are slim, portable, permanently connected to the Internet, and yet powerful enough to read books and magazines on and to accommodate television services. Given their form factor which is designed to interact and to engage the user, tablets have the potential to define the next generation of magazines and television formats. There is an opportunity to improve the existing conventional user experience and a huge potential to maximize the value of existing assets, given that they are interactively delivered and that producers gain a better understanding of the user's requirements. By adopting tablets and interactive devices as an alternative distribution channel, producers will be able to monitor the user's behavior in real-time and to react accordingly. In this position paper, we will explain the main producer's concerns, investigate what technology is available, and how these can be matched with the consumer's interest.

Part of this work has been realized as a Proof of Concept setup, which has provided us with a clear insight in the capabilities of the state of the art technology, where we need further development and what standards are missing in order to enable a system set-up based on interoperable components. In order to clearly identify and define the mutual needs of the consumer and the producer, we will first of all lay out the basic requirements of a film or television production and make an educated guess of future end-user expectations. Given these, we will present an architecture overview, discuss the available standards and discuss some technical challenges.

**Limecraft** designs and develops production software for the creative professional. Having a strong background in television production, it is in the right position to clearly define what's on the producer's mind. It is in Limecraft's interest to have a first hand in the specification of next generation television formats and to subsequently advance the state of the art in terms of television production software.

**VRT-medialab** is the R&D department of VRT, the public service broadcaster of the Flemish part of Belgium. One of the core projects of VRT-medialab is the development of an interface system between the producer and the consumer in order to measure and to understand the user behavior in real-time.

**Multimedia Lab (MMLab)** is a research group within Ghent University and one of the founding research groups of IBBT. The research topics that are dealt with by MMLab and that are related to this workshop are media streaming, multimedia semantics, metadata technology in general. For instance, we have strong expertise in HTTP adaptive streaming technologies, developed a number of metadata models for file-based (tapeless) TV production including support for all related engineering processes (e.g., full proof-of-concept for drama & news production), and are actively participating in W3C standardization activities. More specifically, we are member of the W3C Media Annotations Working Group and co-chair of the W3C Media Fragments Working Group.

## 2. Application

Tablet PC's are perfectly suitable to accommodate portable television services. Given their form factor which is designed to interact and to engage the user, tablets have the potential to define the next generation of magazines and television formats. The user will be able to freely navigate between items, any visual object or advertisement will be clickable and for every object there will be plenty of background information available. However, while the first newspapers and magazines are currently adopting the tablet as an alternative distribution channel [1, 2], there currently seems little or no commercial interest to extend existing broadcast services by interactive applications.

Given the current state of the technology, the Internet is perfectly capable of transporting high-resolution video over the Internet. Unfortunately, when it comes to providing a user-friendly television experience, many more issues, apart from encoding, transport and play-out of audiovisual material, remain to be solved. In this section we explore the additional complications that arise when we would consider creating a television application on a tablet PC or another type of interactive device.

Considering the end-user's interest, we should take into account that a television channel usually offers a blend of programmes in a particular order such that the bulk of audience gets concentrated on particular focal areas for the purpose of maximizing advertisement time valuation, a concept known as "prime time". We should be aware of the fact that programmes are placed in a proper context of advertisements and sponsoring, auto-promotion, introductions, interstitials, cross-references to other programmes, etc. On an interactive device there is the implicit expectation that all of these references are clickable.

But there is more. Each programme begins and ends with a set of disclaimers, credentials and source references which again are expected to be clickable. We must take into account that each programme is usually based on a number of scenes or stories and that each of these is depicted by a sequence editorially constituent images or shots. A story can be a news report, a scene in dramatic production or a topic in a magazine. For creative professionals the story is the editorial logical unit of work [3, 4], but for end user it is an important concept as well, since the stories will make up the 'Table of Content' that enabled the user to randomly access individual items in the context of the programme.

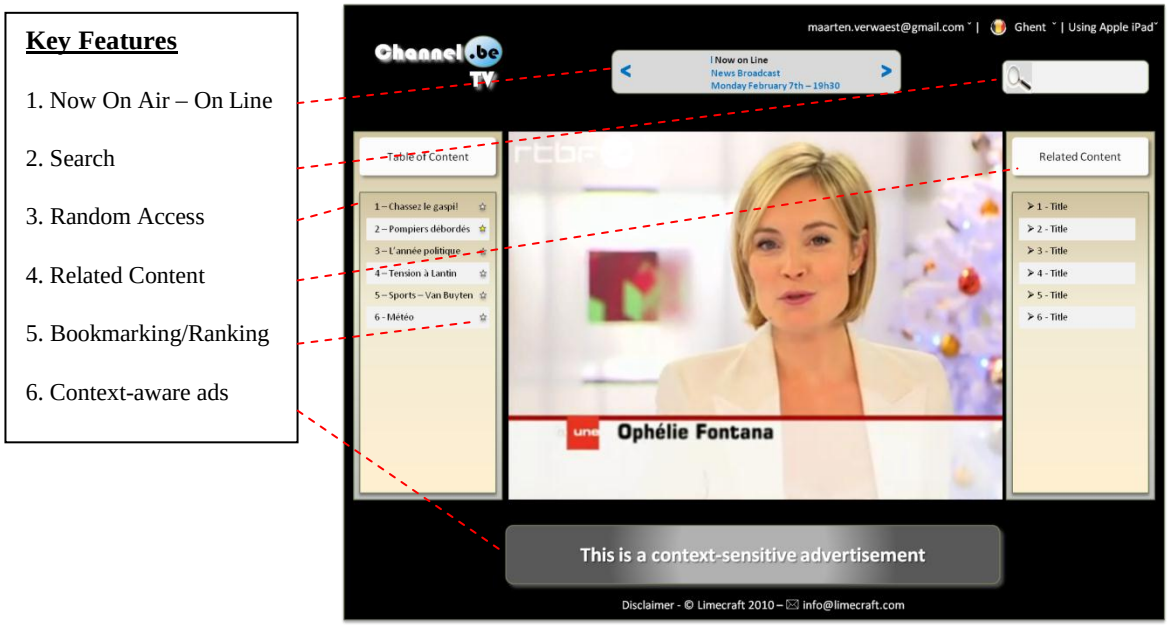


**Figure 1 – A Television Application on a Tablet or another interactive Device**

Assuming there is, apart from the ability to play-out any content on demand, a 'Now on Air' indicator, it should refer to both the programme and the item if possible. When searching for content, the proper result set is a list of

items instead of programmes. Then again, each story or item should refer to background information or to a list of related content (e.g., the entire interview, statistics, documentary resources that have been used during editing) that is again referred to as a list of individual items. In either case, the assumption that any interactivity is defined on an item basis while any professionally produced content is distributed per programme, opens up a whole range of issues and challenges.

On the other hand, we should consider the producer's requirements. In the production process, the producer is responsible for the financing and he is the one who takes the decision to start the actual production of a particular programme or film based on a forecast of the revenues. As suggested, current broadcast services are usually organized by physical or logical "channels". Each channel is operated by a channel manager who, by carefully selecting particular formats and genres from the producer, ensures a certain editorial integrity and as such he can focus on a particular target audience. The value of the advertisements that generate the revenues to finance the production of the content is proportional to the size and the coherence of the audience. By also targeting online devices, which have the potential to access any content on demand, the producer and the channel manager can not only substantially enlarge their target audience, they also get the tools to better understand the particular interest of their audience and they can monitor their audience in real-time. However, unless the producer provides sufficient in-context background information, advertisements and related content, there is a permanent risk that the end-user is distracted by out-of-context background information, other 3rd party services or his own social network. Hence, it is in their interest to provide a television application that meets the consumer requirements in terms of usability and interactivity, providing elementary metadata, enabling random access, search functionality and proper bookmarking features.

While at first sight and given the state-of-the-art technology it should be possible to implement an application as described with reasonable complexity, a detailed analysis of some real-life use cases impose additional non-functional requirements, major challenges and a number of fundamental and unresolved problems.

In particular, we have been able to identify the following issues:

- **Real-time processing** – depending on the channel, 20 to 80% of the programme content is assembled ad hoc by a live master control process. By consequence, it is in general not possible to produce the on demand content as a batch process and to make it available before the actual broadcast. Assuming the on demand content should be available as soon as possible after broadcast, which includes transcoding, segmentation, indexing, and provisioning of background content and interactive extensions, all of these processes should be executed in real-time and simultaneously. Downstream, the end-user application should be able to render open files, i.e., implement streaming protocols for the audiovisual material as well as the associated text or metadata tracks;
- **Frame-accuracy** – In a professional context, a programme of a movie is an assembly of audiovisual material and artifacts such as logos and captions. These are mixed during the *master control* processes. The location of the item-boundaries and the artifacts should be frame-accurate and hence cannot be precisely expressed by decimal fractions of a second and it should be possible to use a SMPTE time-code as is the case in most professional standards [5];
- **Object identification** and **Visual Hyperlinks** – background information should be provided per object and not per image or per sequence of images. So we need the means to define and to identify a *Region of Interest* in space and time that can serve as an anchor for hyperlinks. The ability to identify particular objects make the objects in the video (and not the video as such) a referable resource of the semantic web;
- **Data tracks** – apart from the ability to display video, audio and plain text (HTML5 refers to Web Video Text Tracks or WebVTT) and much richer applications would be possible when being able to include tracks of structured data as well. The ability to identify particular shots or scenes by spatio-temporal references (date and place of recording), actors or other named entities as they appear and the ability to formally annotate particular sequences of audiovisual material by subject references would enable a Wikipedia-alike application whereby video is the container instead of the wiki page. Moreover objects (and not the video-clips as such) will become accurately searchable by specialized search

engines. An example of data-tracks that implement time-coded and structured information is the SMPTE DMS-1 (Descriptive Metadata Schema) standard [12].

- **Synchronization** – An extremely complex problem is the synchronization of video, audio and data. Since we should achieve frame-accuracy, and given the need to display images at 50 fps on large high definition panel displays, we cannot rely on SMIL or another script language and script rendering applications to make an ad hoc assembly of the available components and their associated out-of-context metadata. We will need **proper media and file formats** that are capable of supporting markup by which the synchronization is ensured by design;
- It would be fair to anticipate **adaptive bitrate delivery protocol** to deliver audio, video, and any synchronized markup to the end-user, allowing optimization of bandwidth consumption while maximizing video quality;
- **A bi-directional Application Programming Interface (API) to exchange time-coded events** – While modern browsers are being designed to support HTML5, the ability to apply and to interact with content markup using zones of interest, and throwing back events to an application based on which the user can interact, still requires the use of proprietary players like Microsoft Silverlight, Adobe Flash or QuickTime. Applications as described above will require a frame-accurate timer object shared by the browser and the media player and a bi-directional timer API that enables exporting event information.
- **Support for professional formats** – In the near future, we would like to extend the consumer-oriented use cases, by using HTML5 as the basis for professional applications like browsing through the archive, logging material, etc. Therefore the applications that consume HTML5 will need to support professional video file and media formats, including various flavors of MXF [11], thus preventing the need for transcoding and rewrapping the material.

# 3. Architecture

In order to draft an application and standards architecture, given the application requirements as described above, we have use a layered model based on the OSI model for network communication [9, 10], which should be refined in the audiovisual domain by making separate decisions on the tokenization and session control parameters that were defined by OSI layer 5 (Session), and by clearly distinguishing between the encoding and the container which would be part of OSI layer 6 (Presentation).

Otherwise, from left to right, our architecture accounts for a number of processes and applications in the overall production and distribution chain, starting with the actual production (camera work in the studio or field production, editing, sound engineering), mastering or 'packaging and labeling', the actual distribution and finally the play-out application.

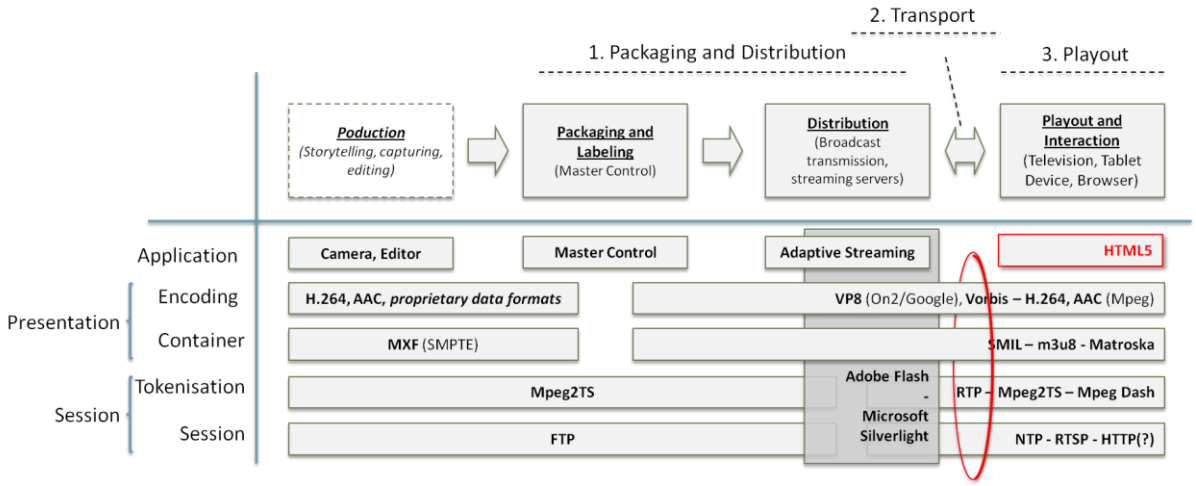


**Figure 2 – Overall Application and Standards Landscape**

In the context of this particular document, we make abstraction of the production processes, as long as we are aware that during the production processes the structure and the content of every single episode and item is determined, potentially linked to background information. In a best case scenario, the production process can deliver all relevant metadata that is used during the packaging processes to produce navigation features and markup of the audiovisual material.

The packaging processes are responsible for the conversion of pro-grade media- and file-formats, usually MPEG-2 Video or H.264 I-frame only and up to 8 channels of uncompressed audio which make up 50 Mbps for standard definition material and about 100 Mbps for high-definition content. The bulk part of the production applications uses MXF as a file format. So the packaging process unwraps MXF, transcodes the media (VP8 or H.264, and AAC) and wraps it again in a file-format suitable for distribution, which can be Adobe Flash, Silverlight-compliant, Matroska or any other format. The packaging process will usually also insert some elementary metadata in the header of the file (labeling). As the packaging processes become more intelligent, it is expected that one or more streams of data will be included, preferably synchronized with the video and the audio such that the play-out processes can interpret the data even when randomly accessing media streams. For example, our current research includes the insertion of hyperlinks as streams of structured data.

Considering the distribution process, we propose an http-based streaming process. Proprietary streaming implementations have been available since the mid '90s - Real networks (1995) and Microsoft Netshow (1996) – and RTP and RTSP have been standardized in 1996 and 1998. However, proprietary solutions have been pushing the functionality of streaming services. HTTP-based streaming services were initially implemented to bypass company firewalls but actually prove to be capable of offering a very reliable and scalable streaming implementation to the extent that http as a protocol is fully supported by Adobe, Microsoft and Quicktime. More recently all major brands have implemented a method for dynamic bit-rate manipulation referring to resp. dynamic streaming, smooth streaming and http adaptive streaming. This particular feature has recently been picked up by MPEG (DASH – Dynamic Adaptive Streaming over HTTP) [13].

Finally we will consider a number of play-out application features that, as a whole, must create a television user experience. That means video will be embedded in a larger application context as described above. Html5 is a developing standard and a key technology to create the link between the interactive application framework (the user agent) and the video player runtime, where the former will be some sort of browser process and the latter a video player process or thread. In fact the video-tag in HTML5 is the API that bridges the two contexts. In general, it should cover of the application requirements identified in section 2.

## 3.1. Server-side - Processing of a live signal, Metadata capturing and communication

Given the application requirements described by Section 2 and the assumption that we are distributing professionally produced content, the architecture of a system that repurposes or re-masters content for online distribution, referred to as the 'Distribution Head-End', should at least contain the following blocks.

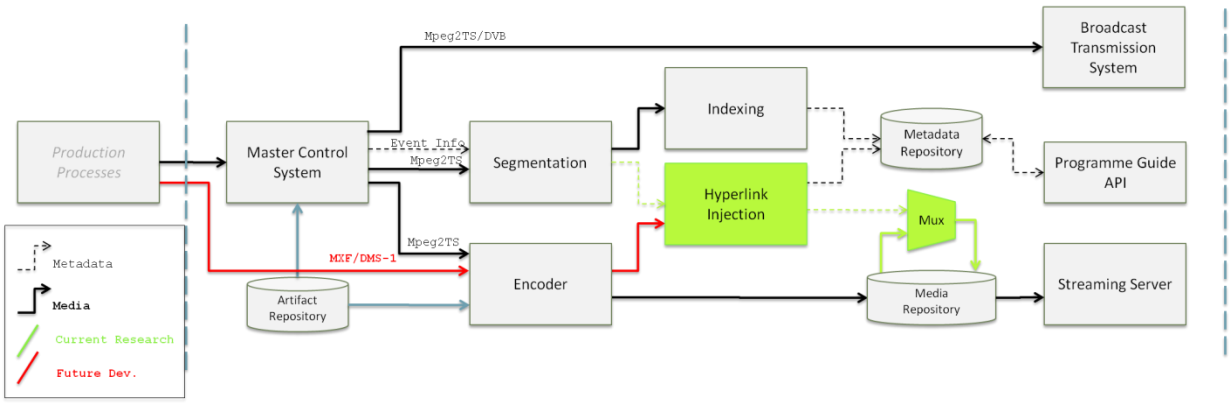


**Figure 3 – Architecture of the Distribution Head-End**

The master control system is the assembly machine that consumes streams or files of video material and produces one or more continuous streams of video, initially intended for conventional television distribution (see upper stream on Figure 3). Most commercial master control system use an artifact repository that contains logo's, graphical templates, character generators, etc and it is authorized to apply these on the images. Moreover the master control system interprets subtitle-files and it *burns* subtitles in the pixels.

For the purpose of our Proof of Concept application, we have extracted a secondary stream out of the master control system that is not contaminated by artifacts under the form of an MPEG-2 transport stream (TS). The stream is sent trough an encoder, delivering chunks of bare encoded audiovisual material (see lowest stream on Figure 3). In parallel, the secondary stream is passed through a series of segmentation and indexing algorithms that populate a time-coded index of the content, referred to as the Metadata Repository that is used by the Programme Guide API.

In order to advance the state of the art, we are currently developing a method and a system to embed hyperlinks in the audiovisual material, which is, in the context of the packaging and distribution process, primarily a matter of properly storing information about the anchor and the hyperlink destination. Assuming this information is embedded and interleaved in the media file, the interface (see section 3.2) will make abstraction of it and further down the stream the play-out application must be capable of interpreting and resolving these hyperlinks.

In order to avoid proprietary encoding of data tracks and ditto interpretation downstream, we have found that conventional consumer-oriented file formats, including MP4, Matroska, m3u8 are not suitable by design to embed structured data. The Matroska container can contain plain text, but would require additional application logic to parse the text in order to retrieve the anchor, title and destination of a hyperlink. As an alternative, we are currently investigating the use of the professional file format MXF which is capable of supporting structured metadata that is frame-accurately synchronized. However, the use of MXF as part of the television user experience would require support for MXF by HTML5-implementing user agents.

In either case we are planning a closer interaction of system components of our back-end components in the near future by which we will be able to short-cut the conventional master control system by directly feeding the encoder MXF-files from the central *Media Asset Management* system. Assuming the MXF file is containing accurate and properly embedded master data in the form of MXF DMS-1, the encoder that will be directly triggering the hyperlink injector, as opposed to the current approach where the hyperlink injector is a best-effort image analysis system.

## 3.2. Interface

Since the media items are prepared at server-side with their accompanying metadata and other context information, these metadata need to be delivered to the end-user in standardized, efficient, and synchronized way. Therefore, we propose to send these metadata inside the media container, synchronized with the audio and video tracks using frame-accurate timing information. Moreover, since the user agent needs to play and interpret the media items in real-time, audiovisual data and metadata information must be interleaved in the media container. In terms of standardized solutions, Timed Text [6] combined with multi-track container formats such as MP4 or WebM could be a good start. However, we should investigate to which extent technologies such as Timed Text are suitable to represent the metadata used by our media items, as well as how well they integrate with existing container formats.

Next to synchronization between audio, video, and metadata in the container format, there should also be a standardized way to point from within the metadata to audiovisual data, both in the temporal and spatial dimension. Currently, Media Fragment URIs [7] appears to be an ideal candidate to point to spatio-temporal locations in media using a URI. However, to use them in a next-generation TV environment as defined in this paper, there are still a number of shortcomings. For instance, frame-accuracy is obtained by means of SMPTE time codes, but there is no support for high definition formats (i.e., 50 fps). Also, spatial fragments (i.e., pointing to a region of interest) are limited to rectangular shapes. Further, these shapes are static within the whole media fragment.

Finally, the packed media items need to be delivered using a standardized media delivery protocol. Currently, a new range of streaming techniques based on HTTP is arising, which are characterized by an adaptive behavior. These techniques typically segment the media items into separate chunks on the server. During this segmentation phase, an index file is created, containing a list of these media segment files and additional metadata. When multiple quality versions are available, the end-user can decide to switch from one version to another, simply by requesting the next chunk from a different quality version. This way, the quality of the delivered media items can be smoothly changed based on for example changing network conditions. The latter looks promising in the context of TV on a tablet given the changeable usage environment of tablet users. Apple, Microsoft, and Adobe all came up with their own proprietary version of HTTP adaptive streaming. Currently, MPEG is finalizing a standard for HTTP streaming called DASH (Dynamic Adaptive Streaming over HTTP) [13]. However, may questions remain unanswered regarding the use of HTTP media streaming on the Web:

- Will W3C recommend one particular HTTP adaptive streaming standard?
- In case DASH is the preferred specification, how to deal with its complexity, which media formats will be supported (WebM, MP4, MPEG-2 TS?), will there be patent issues, …?
- How to integrate HTTP adaptive streaming with HTML5 (see also Section 3.3)?
- How to deal with situations where it is not desirable that the end-user always decides when to perform adaptations on the media delivery process? In this case, some kind of content negotiation protocol should be defined between client and server, so that the server can make decisions based on the client's environment.

### 3.3. Decoding – Play-out, Interaction

To implement the end-user application, HTML5 and more specifically its <video> element with corresponding API seems to be a promising technology. It will be supported by browsers on regular PCs, but also on tablets and smartphones and possible also on televisions. However, there are still a lot of issues if we consider using HTML5 to implement the application sketched in Section 2.

First of all, the <video>-element must support HTTP adaptive streaming technologies, as discussed in the previous section. Currently, a number of threads on the FOMS-mailinglist [8] are already discussing this integration.

Second, the API of the <video>-element must be aware of the synchronized metadata that is sent together with the audiovisual data. More specifically, events should be triggered when new metadata is available. In our case, typically when a new media item is started, the metadata attached to this item should be interpreted and displayed. Currently, three major problems appear when looking at the current version of the HTML5 draft:

- There is currently no support to request metadata from the API. Note that the Media Annotations Working Group is currently working on an API for requesting media properties, but these properties are too high-level and do not fulfill our requirements.
- There is no support for event-based triggering when new metadata is available.
- The video player cannot be controlled in a frame-accurate way.

Finally, interaction with the media content (e.g., clicking on objects in the video) is possible with HTML5 to a certain extent. However, we believe that a standardized solution for hyperlinks in media content on the Web should exist (cf., the way hyperlinks are standardized for text). More specifically, the <video>-element API should support interactive spatio-temporal regions, which are addressed by means of Media Fragment URIs. Further, events (such as click events) should be attached to these URIs in a standardized way.

## 4. Future Work

Future work will include the direct manipulation of media files without the intervention of a master control system, actually setting up a specific master control system that delivers content natively produced for IP-based and interactive distribution. Where IP-based distribution is considered a line-extension of the main broadcast

channel as for now, it is expected that, on the mid-long term, conventional linear television distribution becomes a side-deliverable of IP-based distribution.

As we will be consuming material from the central *Media Asset Management* system instead of repurposing the existing broadcast signal that comes out of the master control system, we will also be able to interpret the embedded metadata and to directly inject visual hyperlinks, where as today we can only insert the metadata after encoding, by using downstream process that cause additional delay and further inaccuracy in terms of synchronization.

Finally, we will be investigating the use of HTML5 in strict a professional context, i.e. for constructing production applications. To that extent, we will explore the possibilities to somehow include support for MXF as a file format.

## 5. Conclusion

We have discussed how tablet PC's and interactive devices in general can be used to extend existing conventional broadcast services by using the Internet as the primary distribution medium. We have assumed a number of consumer requirements, i.e., the use of a table of content in order to be able to freely control the flow between the items, how related content should be integrated per individual item. We have discussed that, in the near future, individual objects in the video will become addressable and usable as anchor points for hyperlinks.

In order to enable robust and interoperable support, we will need to advance the state of the art in terms of file formats, making existing formats more intelligent and flexible (MP4 or Matroska) or by adopting a suitable professional file-format (e.g., MXF) in the domain of consumer applications, such that embedded and frame-accurately synchronized structured metadata is supported.

## References


1. C. Anderson, "Wired Magazine's iPad Edition Goes Live", Wired Magazine, June 2010.
2. The New York Times, Online Services, Available at: http://www.nytimes.com/services/mobile/, Retrieved 07/01/2011.
3. E. Mannens, M. Verwaest and R. Van de Walle, Production and multi-channel distribution of news, Multimedia Systems, 14(6):359–368, 2008.
4. D. Van Rijsselbergen, M. Verwaest, B. Van De Keer, R. Van de Walle: Introducing the data model for a centralized drama production system. In: Proceedings of the IEEE International Conference on Multimedia & Expo 2007, pp. 615–618, July 2007.
5. EBU. P/Meta Metadata Metadata Library v2.1. Technical Report 3295, July 2009, Available at http://tech.ebu.ch/metadata/p_meta.
6. G. Adams, ed. Timed Text Markup Language (TTML) 1.0. W3C Recommendation, November 2010. Available at http://www.w3.org/TR/ttaf1-dfxp/.
7. R. Troncy, E. Mannens, S. Pfeiffer, and D. Van Deursen. Media Fragments URI 1.0. W3C Working Draft, December 2010. Available at http://www.w3.org/TR/media-frags.
8. http://lists.annodex.net/cgi-bin/mailman/listinfo/foms
9. ISO/IEC Standard ISO-7498-1:1994, Basic Reference Model: The Basic Model, 1994.
10. ITU-T X.200 -- Basic Reference Model: The Basic Model, 1994.
11. SMPTE 377M – The MXF File Format Specification, 2004.
12. SMPTE 380M – Descriptive Metadata Schema – 1, 2004.
13. MPEG advances Dynamic Adaptive Streaming over HTTP (DASH) toward completion, Guangzhou, CN – The 94th MPEG meeting was held in Guangzhou, People's Republic of China from the 11th to the 15th of October 2010 (http://www.itscj.ipsj.or.jp/sc29/29w02911.pdf).