Applying PC network technology to assess new multimedia content analysis applications for future consumer electronics storage devices

Fons de Lange[1] **Jan Nesvadba**[1]

[1] Philips Research, The Netherlands

Abstract

The paper deals with software productivity improvement for consumer multimedia devices by means of PC & component technology and shows how this is done for complex real-time *content analysis* applications used in advanced new storage products of the future.

Keywords: Multimedia content analysis, PC networking, prototyping, UPnP

1. Introduction

This paper presents the problem of "early" feature evaluation; i.e. how to assess the sense and simplicity of new features and feature combinations for yet nonexistent products and how can this be done with little effort when features are still in their infancy? Although difficult, evaluation of still immature features is a must to enable the assessment of important aspects of a possible future product. In fact, when envisioning a new product-concept, very often still a lot of features and functions are nonexistent and need to be invented first, subsequently they must be implemented and combined to give a first impression of a product. Next, some evaluation of the feature combination must be possible to judge its value and feasibility. The outcome of this process is an updated vision of the imaginary product and will fuel the generation of new ideas and the development of other new features. Fig. 1 visualizes this process of *early feature evaluation*, shown as four phases:

1. **Imagine**
   This is about envisioning and imagining a new product. An example of a *product vision* is a *Personal Video Recorder* that enables a user to find and watch any TV program that has been broadcast during the last few months for a set of preferred channels.
2. **Invent**
   Here, one has to think about the types of features and the enabling technologies required for the envisioned product. An example of an important enabling technology is *Content Analysis* [1] such as *Virtual Channel Creation* [2] that enables a user watch all his/her favorite TV programs.

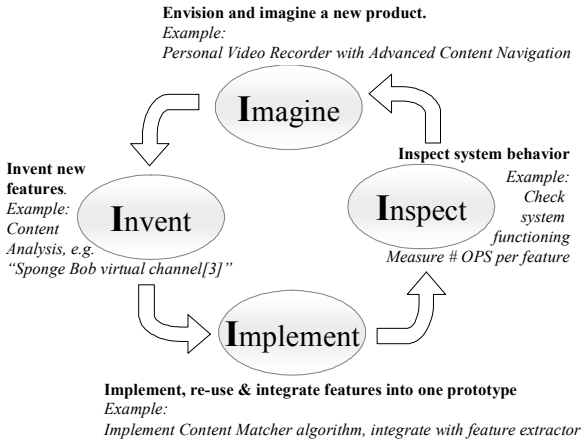

Fig. 1: Early evaluation of features to assess future product concepts based on multimedia analysis.

3. **Implement**
   To increase the understanding and learn more about the possibilities, benefits and (technical) limitations of an imaginary product, critical parts must be prototyped. An effective way of doing this is to look for any technology that is relevant to the product concept and easy to integrate with other technologies. System functionality that is crucial to the product, but which is not available anywhere must first be invented and then created from scratch.
4. **Inspect**
   Once a prototype is created that implements sufficient functionality of the envisioned product, one can analyze the system behavior, determine component interfaces / interactions and measure important characteristics of specific feature combinations, e.g. the memory, streaming bandwidth and performance requirements. Consider a specific feature for an imaginary *personal video recorder* such as a *football match detector*. By prototyping and analyzing its

behavior one can determine if it is accurate enough, if it is feasible in combination with other features – with respect to performance and memory usage – and last but not least, if the feature is attractive and easy to use. This will further stimulate the imagination and ingenuity; see Fig. 1, leading to an improved product concept.

A complicating factor is that features, if implemented and available at all, are very much in their infancy and subject to frequent changes as applied by the feature designer. This problem is especially applicable to *content analysis* for storage systems, where *content* analysis supports the content retrieval and navigation process. Since the storage capacity of different types of storage devices, such as hard disk and flash, is rapidly growing, more and more audio, video and photo content can be stored on these devices. In the near future this will grow to several Terabytes of storage within a single Hard Disk device [3], capable of holding massive amounts of AV data, see Fig. 2. It becomes clear that *AV viewing*, *AV searching* and *AV browsing* functionality is therefore essential for a successful introduction of mass storage devices in CE products.

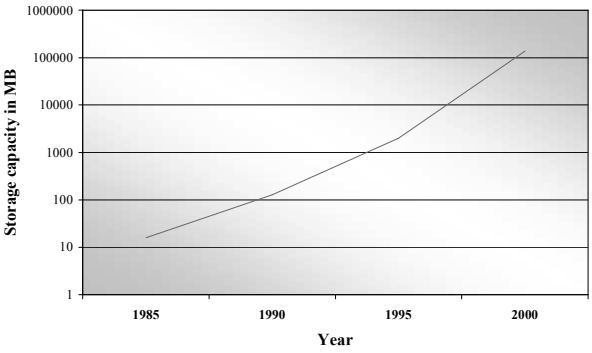

Fig. 2: Trend for Hard Disk storage capacity.

Content analysis is a key enabling technology in this respect. In the domain of AV content analysis, many new algorithms, such as *speech/music discrimination*, *scene change detection*, *commercial detection* [4][5][6], etc. are developed at an increasing rate and become available for third party use at sites inside and outside Philips, companies and universities [13]. These algorithms enable audio video content to be segmented such that content- viewing, browsing and searching can be greatly enhanced. All these *content analysis* algorithms are different with respect to how they are designed and implemented. Often they are developed with different programming tools, using different programming languages and having different communication models for interacting with their environment.

The solution that the paper describes to enable product-concept assessment for such *content analysis* enabled systems is a PC based design methodology for easy integration of features. This is done using component technology for packing each feature, *as-is*, into components and by providing standard technology and tools/platform to flexibly interconnect them. In our prototyping framework, each component is an independent executable program that communicates with other components via TCP/IP and UPnP networking [7]. Basic TCP/IP is used for streaming data over the network, while UPnP is used to enable applications to automatically find components, set-up connections between them and to control them. All this irrespective of their location in the network. Section 3 describes this approach in more detail.

The paper is structured as follows. Section 2 gives a short introduction to the field of AV *content analysis*. Section 3 describes the PC based approach to early evaluation of (*content analysis*) features and algorithms. Section 4 illustrates the effectiveness of the approach. In particular it describes the architecture of a large system, integrating many AV content analysis algorithms and an AV database. Finally, section 5 presents the major conclusions of the PC based feature integration/evaluation approach.

2. Introduction to AV content analysis

2.1. Rationale

Extrapolation of the storage capacity of a single hard disk drive, shown in Fig. 2, yields 100 TB in 10 years from now, which comes close to storing all songs in the world[1].

It is obvious that it becomes an impossible job to find a particular MP3 song among 10 million others on some storage device by just listening to the music. The only way to find content is to use metadata to summarize and describe the content, which can then be used by a user to more efficiently search for content.

Several sources of metadata for broadcast video exist. These are Teletext for analog TV in Europe, Digital Video Broadcast *Service Information* [8] and the Internet. The reliability of all these sources of metadata depends on the associated profitability for the provider of these services. Moreover, different providers use different metadata formats and metadata

[1] According to the CD database on the internet, http://www.cddb.org, there are currently 20 million songs on CD.

semantics. All this makes it difficult to reliably and consistently annotate the recorded AV material to enable easy navigation and retrieval of content.

Native content analysis, i.e. content analysis done within the consumer device, eliminates these problems. It enables *duty-free* gathering of metadata for any received and recorded AV material in a consistent way. It can be used to generate metadata in the absence of a metadata service, to enrich the provided metadata to offer extended searching capabilities, and last but not least to enable new advanced content navigation features such as *commercial skip* [6] and *virtual channel* creation [2].

2.2. Content analysis process

Audiovisual content analysis can be applied to a variety of content types such as broadcast commercial content, downloaded content or private home video content. In consumer electronics (CE) devices the content management, retrieval and navigation has to be intuitive, easy to use and 'lean-backward' oriented. As a consequence the interaction with the user has to be performed on a semantic meaningful level, which requires high-level semantic metadata about the audiovisual content. Semantic metadata can be extracted through the combination of various low- and mid-level descriptors of multiple modalities such as audio, video and text. The audio and audio-visual descriptors are extracted by analyzing the audio, respectively audio-visual signal, either in the baseband (pixels for video or PCM[2] for audio) or the compressed domain e.g. MPEG-2.

Fig. 3 visualizes the overall process of semantic metadata extraction.

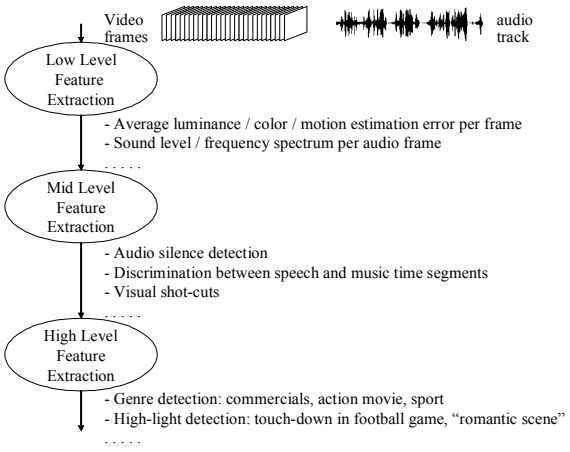

Fig. 3: Overall content analysis process.

[2] Pulse Code Modulation.

It consists of three phases: *low-level-, mid-level- and high-level feature extraction*. At each stage in the process, metadata can be stored and retrieved from memory, to be used by the next step in the process and by navigation applications.

3. PC based feature assessment

When considering new product concepts based on multimedia content analysis, one should not concentrate on minimizing the hardware, CPU / memory requirements for each content analysis feature, but instead one should assess their usefulness in combination with other features. As a consequence, the assessment of product concepts in the early stages of design requires a powerful prototyping system with ample CPU and memory resources. Moreover, implementation and integration of new experimental algorithms should be easy and fast. Finally, the mapping of a selected set of algorithms to an embedded system must be straightforward and with minimal effort.

The prototyping system we use for the assessment of CE products with multimedia content-analysis features satisfies these requirements through powerful PCs, off-the-shelf PCI cards, standard PC development tools and networking technology. Content-analysis algorithms are modeled as black box components with standardized interfaces that are invariant across all platforms, which facilitates the mapping process. Only the algorithms need to be tuned for a specific hardware platform, while optimized implementations of interconnection technology are available for each platform, offering the same interfaces across all platforms.

As an example of this approach, consider a simple streaming application as depicted in Fig. 4a and its implementation in software (Fig. 4b).

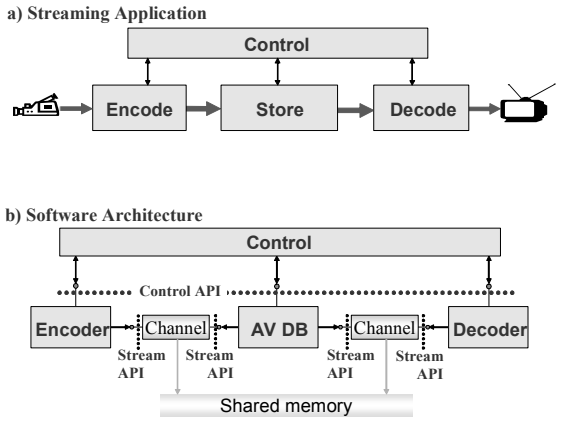

Fig. 4: Streaming application and corresponding software architecture.

It shows three software components, i.e. *Encoder*, *AVDB* and *Decoder,* which are controlled via interfaces (*Control API*) by component *Control* and stream AV data to each other via channels offering streaming interfaces (*Stream API*). Moreover, Fig. 4b also shows that channels are implemented using shared memory, a common practice in embedded systems.

3.1. Early evaluation of features

For fast, early evaluation of new advanced experimental features, it is preferred to pack all features into components without any modification (as far as possible). This means that such features are typically hardly optimized, because the first priority now is to assess their functionality. Evidently at this point it is a waste of effort to optimize any features, before appreciating what they may have to offer. This means that in many cases a single PC is not enough to assess the combined functionality of a number of features. Depending on the number and type of features, even the most powerful multiprocessor PC may be incapable of doing so.

A solution is then to use multiple PCs, run CPU hungry components on different computers and have them communicate control and streaming data over the network. As an example, Fig. 5 shows a simple network of 3 PCs, where each runs one or two components of Fig. 4b in different process address spaces. Evidently, this is not possible without introducing additional entities that handle the networking of control and streaming data for each component. This is described in the next subsection.

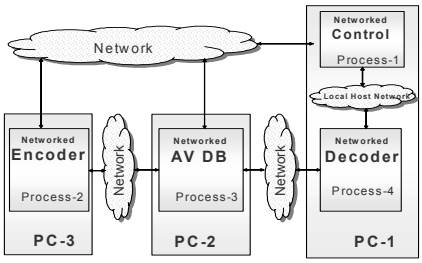

Fig. 5: Streaming system of Fig. 4, with networked enabled components, mapped to 3-PC network.

3.2. Networked components

To enable control applications to find streaming components in the network, create streaming connections between them and allow components to communicate control and streaming data via the network, each component must be extended with networking functionality. Fig. 6 shows this for the example of Fig. 5.

As shown in Fig. 6, each component is extended with a stub, which interfaces with the network. At the control side, a new entity is introduced, the *connection manager*. It creates a *proxy* for each component it discovers in the network. This is done with UPnP networking [7]. To this purpose, the connection manager functions as an UPnP control point, while each *component stub* is an *UPnP device*, announcing its presence in the network. As a result, the connection manager can create *component proxies*, and hand the associated interface pointers to the application.

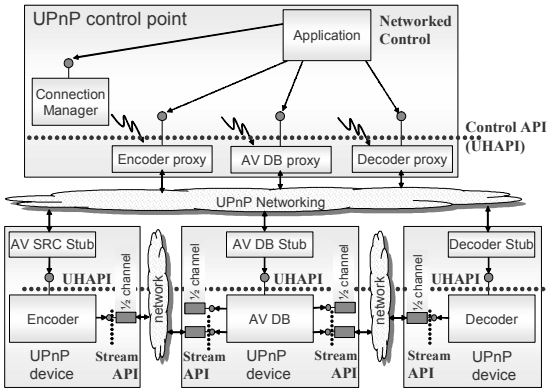

Fig. 6: Proxies, stubs and half channels to enable networking for applications and stream components.

Each *component proxy* implements exactly the same interface as the component itself. Moreover, once created by the *connection* manager, each *component proxy* only communicates with its associated *component stub* via the network using UPnP networking. The *component stub* translates the network commands into straight *control API* calls for the bare, non-networked component it is attached to. As a consequence, when optimization has taken place, and all components can run on the same computer within the same process address space, then all network proxies and stubs can be removed and the application can directly connect to the components.

Fig. 4b and 6 also show the important interfaces for control and streaming. It is key that both the streaming and the control interfaces are well defined and widely used in embedded system architectures of consumer products, as to facilitate the mapping of PC prototypes to real product architectures. In our prototyping framework, all components and interconnection technology adhere to the UHAPI [9] interface standard for controlling stream processing functionality and YAPI [10] as the interface for passing streaming data between signal processing components. All these interfaces must be preserved

when implementing systems on a PC network, see Fig. 6.

The *channel* component in Fig. 4b merely implements a streaming interface to shared memory, which is used in an embedded system to buffer the data between stream processing elements. For the networked case (Fig. 5, 6) the channel is split in two *half channels* to preserve the streaming interface. The buffering of streaming data can be done indirectly via some *memory server* in the network or directly by standard network buffers.

This approach enables adequate assessment and testing of new features and combinations of features before doing the actual product development. As a result, requirement specifications will be more accurate, feature interactions are better understood beforehand and resource requirements of features can be obtained by measurement.

4. Content analysis system

We have built a content analysis system based on the technology described in the previous section. It is a system demonstrating more than 40 different content analysis algorithms running on 11 PCs that concurrently stream data to one another via the network. These algorithms analyze audio/video signals in real-time; their output is displayed on different computer displays, see Fig. 7 and stored into a SQL database.

Fig. 7: Part of system set-up: 6 LCD screens showing multimedia analysis results.

At the same time, the audio / video signal is stored in a real-time file system. Offline, further content analysis is done, e.g. to detect and analyze key-frames, and results are communicated with a graphics user interface. Among other things, the GUI shows key-frames as thumbnail pictures on the screen, which a user can select to start a replay.

4.1. Architecture

The overall system architecture is depicted in Fig. 8. A digital signal is captured, decoded and re-encoded with the Chrysalis MPEG2 Codec [11], which extracts some low-level parameters out/of the video signal, e.g. average luminance and color per frame. The same thing is done for audio but no special hardware is used for this except for a simple audio capture card for PC.

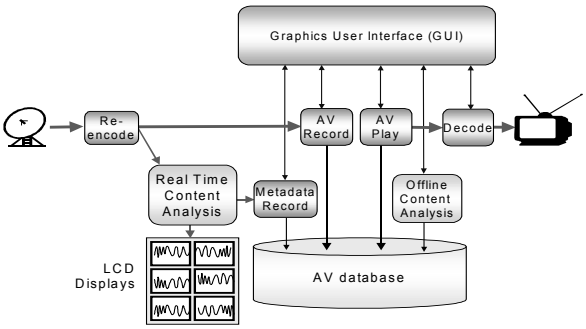

Fig. 8: Top level view of content analysis system

The actual content analysis part, *Real Time Content Analysis* in Fig. 8, is quite complex: the complete set of real-time content analysis algorithms has been clustered into 10 streaming tasks and distributed over 11 PCs. They communicate with each other, the database and 3 LCD-display viewing-tasks via stream channels. Fig. 9 shows this part of the architecture in more detail.

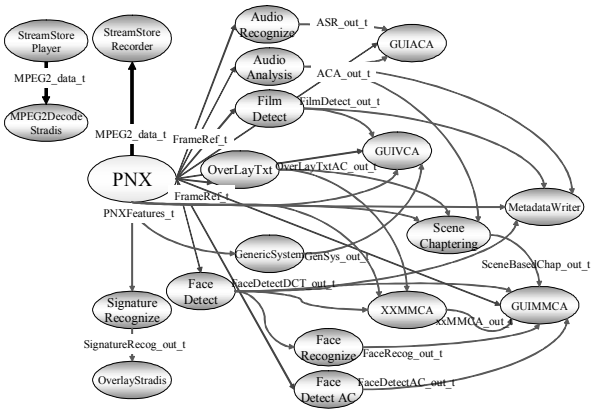

Fig. 9: AV content analysis tasks concurrently running on different PCs, communicating via the network.

The AV stream is separated into an audio and a video stream. The Automatic *Speech Recognition* module extracts spoken words from the audio stream and stores them as ASCI into the meta-database via the *Metadata Writer* component. In parallel the *Audio Analysis* module classifies the audio content into classes such as speech, music, noise, silence and cheering [4]. The video stream is the input for various Video / Multimedia Content Analysis (VCA, MCA)

modules, such as the Film Mode discriminator, which separates into video and moving pictures. Furthermore, the Face Detection module identifies faces and their location [12]. Consecutively, the Face Recognizer tries to find the matching ID, i.e. name, of the face instance by means of a biometrics database. Additionally, the Signature Recognition/Matching matches extracted video signatures with a signature database to identify repeating content. An Overlay Text Detection module identifies and localizes text instance, e.g. subtitles, in the video content.

The central component, indicated by *PNX*, performs both low-level feature extraction and MPEG2 encoding. Moreover, it generates a time-code on video frame basis, which is fed to all content analysis components. They use the time-code to provide a time stamp for all generated features. This way all subsequent processing is able to display and store the extracted features in a synchronized way.

Each component transmits the features it has extracted over a network channel to another component. Since each component generates unique metadata features, each pair of communicating components must *know* the type of the data being transmitted. To this purpose over 12 metadata data types were defined, see Fig. 9, enabling each component to correctly interpret any metadata received.

The stream processing components/ tasks depicted in Fig. 9 stem from many different sources/development groups within Philips. By encapsulating them by a thin shell – implementing the required streaming and control interfaces according to the description in section 3 – they were easily integrated into one content analysis system.

5. Conclusions

To enable fast evaluation of content analysis systems, to come to sensible solutions that are easy to use, PC based prototyping is a must. This is extremely important to be able to understand the feasibility of future CE storage products that heavily depend on advanced *content analysis* features.

However, to prevent PC based system solutions being created that cannot be implemented on more resource constrained systems having a different software / hardware architecture, a number of boundary conditions must be met. Among other things, this can be achieved by adhering to standardized interfaces for control and streaming, by using interconnection technology that is designed for the platform at hand, and by optimizing the feature-implementation for each underlying HW/SW platform. Experiences with the large-scale prototyping activities we have carried out at Philips Research, see e.g. [9], for the assessment of future content-analysis systems, show that a PC based prototyping approach enables the integration of many different media processing features in a short time and that it allows for accurate analysis of the resource (CPU/ memory) requirements of such components.

6. References

[1] J. Nesvadba, "Real-time multimedia analysis", http://www.research.philips.com/technologies/storage/cassandra , 2005

[2] Ma, Q., 2001. Virtual TV Channel Filtering, Merging and Presenting Internet Broadcasting Channels. In SIGNotes Information Processing Society of Japan.

[3] Maxtor, "Big Drives", Maxtor Technologies, http://www.maxtor.com/_files/maxtor/en_us/documentation/white_papers/big_drives_white_papers.pdf , 2004.

[4] McKinney, M., 2003. Features for audio and music classification. In 4th International Symposium on Music Information and Retrieval. Baltimore, Maryland

[5] Nesvadba, J., 2004-1. Low-level cross-media statistical approach for semantic partitioning of audio-visual content in a home multimedia environment. In *Proc. IEEE IWSSIP'04*.

[6] N. Dimitrova, "Real time commercial detection using MPEG features", 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy France July 1-5 2002.

[7] UPnP, 2005. The Universal Plug and Play Forum. In http://www.upnp.org .

[8] http://www.dvb.org

[9] UHAPI, "A new Application Programming Interface for the CE Industry", http://www.uhapi.org .

[10] E. de Kock "YAPI: application modeling for signal processing systems", Proc. 37th DAC, 2000.

[11] http://www.semiconductors.philips.com/products/nexperia/home/products/dvd_recording/pnx7100/

[12] J. Nesvadba, "Face Related Features in consumer Electronic environments". IEEE SMC, Den Haag, The Netherlands, 2004.

[13] MultimediaN, Multimedia analysis, database technology, and human computer interaction. http://homepages.cwi.nl/~mk/multimedianonline/