# RAPID PROTOTYPING OF MULTIMEDIA ANALYSIS SYSTEMS
## *A Networked hardware/software solution*

Fons de Lange
*Philips Research, High Tech Campus 31, Eindhoven, The Netherlands*

Jan Nesvadba
*Philips Research, High Tech Campus 23, Eindhoven, The Netherlands*

Keywords: Network interfaces, Distributed multimedia analysis, Prototyping, Early feature evaluation

Abstract: This paper describes a hardware/software framework and approach for fast integration and testing of complex real-time multimedia analysis algorithms. It enables the rapid assessment of combinations of multimedia analysis algorithms, in order to determine their usefulness in future consumer storage products. The framework described here consists of a set of networked personal computers, running a variety of multimedia analysis algorithms and a multi-media database. The database stores both multimedia content and metadata – as generated by multimedia content analysis algorithms – and maintains links between the two. The full hardware/software solution functions as a test-bed for new, advanced content analysis algorithms; new algorithms are easily plugged-in into any of the networked PCs, while outdated algorithms are simply removed. Once a selected consumer system configuration has passed important user-tests, a more dedicated embedded consumer product implementation is derived in a straightforward way from the framework..

## 1 INTRODUCTION

Product innovation is a key process in consumer electronics business. Without new products and product features, no consumer electronics company can survive. However, the development of new products is a considerable cost factor, while there is no guarantee that a product will be a success in the market. This dilemma is getting bigger by the increasing rate at which increasingly complex consumer electronics products and features are introduced in the market by an increasing number of competitors from both the consumer electronics and computer industry at increasing development costs per product. So how to get out of this dilemma? First of all, product development should become faster and cheaper. Ways to achieve this is to buy software instead of making it (Lange, 2001), adopt a standard platform architecture that facilitates component integration and replacement (Ommering, 2002}, and use these in combination with well defined platform components and standardized component interfaces.
Secondly, once a product is created, it better have the right set of features that will give it a competitive edge over other products in the market. To maximize the chances of product-survival in the market, different product concepts and feature sets must be assessed. Furthermore, this must be done adequately (to really understand the functionality of the new product), it must be done quickly (for time to market reasons), and last but not least it should be easy to realize in a real product architecture (to minimize the development effort). Figure 1 gives a view on product-concept assessment, 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* (Nesvadba, 2003) such as *Similar*



de Lange F. and Nesvadba J. (2005).
RAPID PROTOTYPING OF MULTIMEDIA ANALYSIS SYSTEMS - A Networked hardware/software solution.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 104-109
DOI: 10.5220/0001232801040109

*Content Hopping* that is required to enable a user to find favourite TV programs.

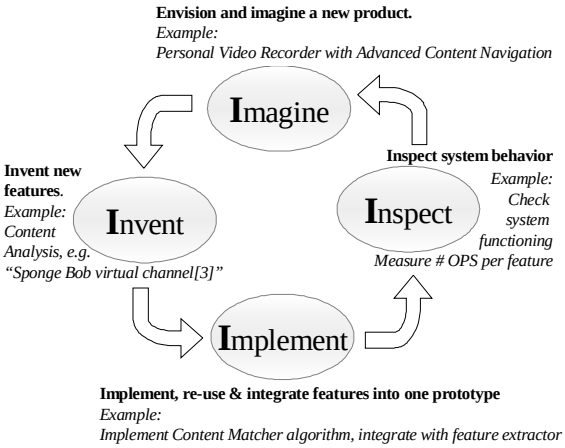


Figure 1: Early evaluation of features to assess future product concepts based on multimedia analysis.

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 – available inside or outside the company – 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. Examples of basic technologies for *content analysis* – to support the *personal video recorder* product-concept example – can be found in several research projects, e.g. in CASSANDRA (Nesvadba, 2003).
4. **Inspect**
   Once a prototype is created that implements sufficient functionality of the envisioned product, one can analyze the system behaviour, 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 behaviour 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 Figure 1, leading to an improved product concept.

As mentioned above, 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 algorithm designer. This problem is especially true for *multimedia analysis* in *storage systems*, where content analysis supports the content retrieval and navigation process.

Here, multimedia analysis is the key enabling technology for using mass storage devices, such as hard disk and flash, in consumer systems holding audio, video and photo content.

Since hard-disk capacity grows rapidly into the Terabyte range (Maxtor, 2004), see Figure 2, advanced viewing, searching and browsing functionality is essential for a successful introduction of mass storage devices in Consumer Electronics (CE) products. For, a 1 Terabyte device can already store over 200 DVDs, 2000 CDs, or 200,000 MP3 songs (5 MB per song). It is obvious that within the next decade, with growing hard disk capacity, it becomes an impossible job to find a particular MP3 song among 1 million others. Conclusively, 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 specific content.

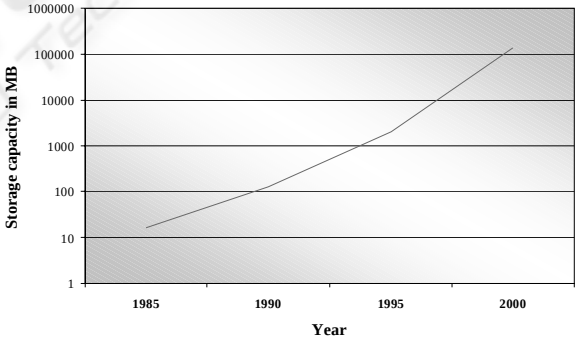


Figure 2: Trend for Hard Disk storage capacity. Creating massive capacity for multimedia content.

In the domain of AV content analysis, many new algorithms, such as commercial detection, speech/music discrimination, film detection, overlay text detection (Dimitrova, 2002) (McKinney, 2003) (Nesvadba, 2003), etc. are developed at an increasing rate at Universities and Consumer Electronics Companies and are exchanged between such parties in cross-organizational co-operations and projects (MultimediaN, 2005). These algorithms enable audio video content to be segmented such that content- viewing, browsing and searching can

be greatly enhanced. All these 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.

Frequent practice is to use embedded systems and dedicated signal processors to evaluate such signal processing features in real time (Cassandra, 2003), while PCs are used for analyzing higher level non real-time features such as graphics user interfaces for media interaction (Hollemans, 2005). Both the embedded systems and PC based feature evaluation is typically based on different architecture models than what is used on a consumer device. This will complicate the mapping of any of the evaluated (and approved) features to existing product line architectures.

All these problems can be tackled by a PC based design methodology. This is described in this paper, illustrating how this is done for complex real-time content analysis applications that are expected to be used in advanced new storage products of the future. The methodology described in this paper, enables easy integration of many algorithms at the same time, by offering scalable performance through PC networking. Furthermore, *early evaluation* of multimedia analysis features and algorithms is possible (for assessing future product-concepts) by packing each feature as-is into components and by providing standard technology and tools/platform to flexibly interconnect them. By sticking to standard Consumer Electronics interface standards such as UHAPI (Philips, 2005), the mapping of prototypes to embedded system hardware is greatly enhanced.

In our prototyping framework, each component is an independent executable program that communicates with other components via TCP/IP and UPnP networking (UPnP, 2005). 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 irrespective of their location in the network.

The next section describes the principles of the PC based prototyping approach. Section 3 describes how these algorithms and applications are prototyped and integrated in one system based on PC technology. Finally, section 4 presents the major conclusions.

# 2 PROTOTYPING APPROACH

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 Figure 3a and its implementation in software (Figure 3b). It shows two software components, i.e. Component A and Component B, that are controlled via interfaces by component Control and stream AV data to each other via a channel offering streaming interfaces.

When components are prototyped on different PCs in the network, additional entities must be introduced that handle the networking of control and streaming data. Figure 4 shows this for a system consisting of three PCs, each one running a different component. This may be necessary when a first implementation of feature *A* and *B* is badly optimized (or not hardware assisted) and together demand more performance than a single PC can handle. Figure 3b and 4 also show the important interfaces for control and streaming.

The *channel* component in Figure 3b merely implements an interface to shared memory, which is used in an embedded system to buffer the data between stream processing elements.

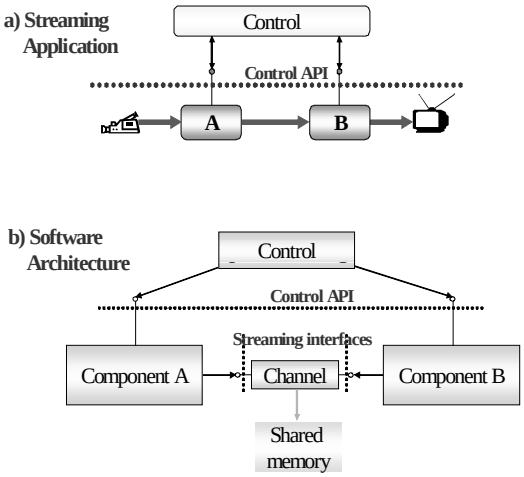


Figure 3: Streaming application & corresponding SW architecture

For the networked case (Figure 4) 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.

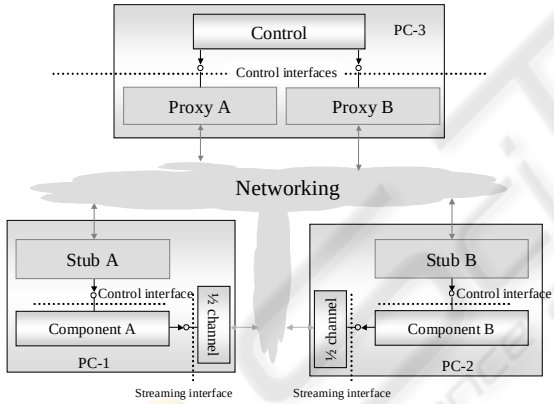


Figure 4: Control and streaming in the multi-PC network

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 (Philips, 2005) interface standard for controlling stream processing functionality and C-HEAP (Nieuwland, 2001) /YAPI (Kock, 2000) 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 Figure 4.

The approach as described above 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.

# 3 IMPLEMENTATION RESULTS

We have built a content analysis system based on the technology described in section 2. It is a system demonstrating more than 40 different content analysis algorithms running on more than 8 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 Figure 5, and stored into a SQL database. 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.

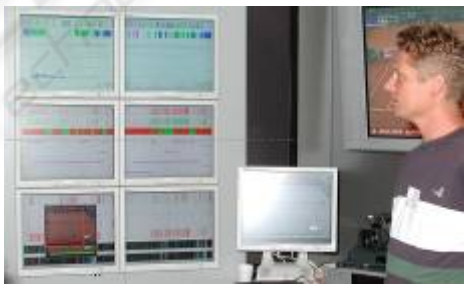

Figure 5: Part of system set-up: 6 LCD screens showing multimedia analysis results.

Among other things, the GUI shows key-frames as thumbnail pictures on the screen, which a user can select to start a replay of the stored audio/video content from the position that corresponds to the thumbnail picture, see Figure 6.

The overall system architecture is depicted in Figure 7. A digital signal is captured, decoded and re-encoded with the Philips PNX7100 MPEG2 Codec which extracts some low-level parameters out/of the video signal, e.g. average luminance and color per frame.


Figure 6: Menu for chapter-based content browsing, generated from multimedia content analysis.

The same thing is done for audio but no special hardware is used for this except for a simple audio capture card for PC.

The actual content analysis part, *Real Time Content Analysis* in Figure 7, is quite complex: the complete set of real-time content analysis algorithms has been clustered into 11 streaming tasks and distributed over 8 PCs. They communicate with each other and display tasks via stream channels. Figure 8 shows this part of the architecture in more detail. It shows several components, performing audio and video content analysis, MPEG encoding and storage.

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 Record* component. In parallel the *Audio Analysis* module classifies the audio content into classes such as speech, music, noise, silence and cheering (McKinney, 2003). 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 2:2/3:2 pull-down mode (Haan, 2000). Furthermore, the Face Detection module identifies faces and their location (Nesvadba, 2004). 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.

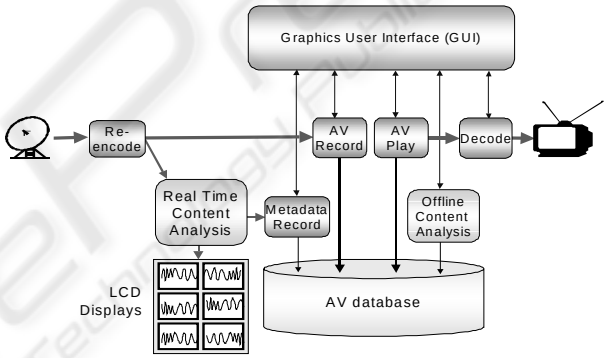


Figure 7: Top level view of content analysis system

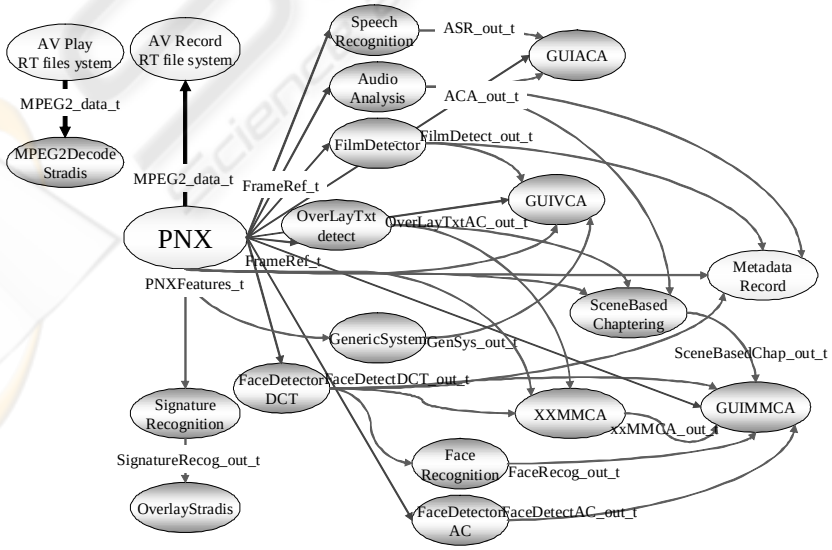


Figure 8: Multimedia analysis tasks concurrently running on different PCs, communicating via the network.

Finally, two Multimedia Content Analysis (MCA) modules extract high-level semantics such as the Scene Boundary Detector (Nesvadba, 2004-1), which segment the content items into meaningful semantic scenes. In parallel, a Commercial Block Detector (Dimitrova, 2002) indexes commercial instances.

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 Figure 8, enabling each component to correctly interpret any metadata received.

All stream processing components/ tasks stem from many different sources/development groups within Philips. By encapsulating them by a thin shell – implementing the required streaming and control interfaces – they were easily integrated into one content analysis system.

# 4 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 *Consumer-Electronics* (CE) storage products that heavily depend on advanced content analysis features.

To assure that PC based system solutions are created that can be mapped onto more resource constrained systems with a different architecture, standardized interfaces are used for control and streaming (Philips, 2005) (Nieuwland, 2002) (Kock, 2000), 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 (Nesvadba, 2003) 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.

# REFERENCES


Haan, G. de 2000. Video processing for multimedia systems. In *ISBN: 90-9014015-8*, Eindhoven.

Hollemans G., 2005. MediaBrowser: In *http://www.research.philips.com/technologies/misc/homelab/downloads/homelab_365.pdf*

Dimitrova, N., 2002. Real time commercial detection using MPEG features. In 9th International Conference *on Information Processing and Management of Uncertainty in Knowledge-Based Systems,* IPMU 2002.

Kock, E. de, 2000. YAPI: application modeling for signal processing systems. In *Proc. 37th DAC.*

Lange, F. de., 2001, The Philips-OpenTV Product Family Architecture for interactive Set-Top boxes. In *4th product family engineering, Springer Verlag, ISBN 3-540-43659-6, pages 187-206.*

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

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

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.*

Nesvadba, J., 2004. Face Related Features in consumer Electronic environments. In *IEEE SMC, Den Haag, The Netherlands.*

Nesvadba, J., 2005. Local real-time multimedia analysis. *http://www.research.philips.com/technologies/storage/cassandra*

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

Nieuwland, A., 2002. a Heterogeneous Multi-processor Architecture and Scalable and Flexible Protocol for the design of Embedded Signal Processing Systems. In *Journal of Design Automation for Embedded Systems, Kluwer Academic Publishers, vol. 7, no. 3.*

MultimediaN, 2005. Multimedia analysis, database technology, and human computer interaction. In *http://homepages.cwi.nl/~mk/multimedianonline/*

Ommering, R., 2002. Building product populations with software components. In *ISCE 2002, pages 255-265.*

Philips, 2005. UHAPI, a new Application Programming Interface for the CE Industry. In *http://www.uhapi.org* .

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