

# MediaAlert – A Broadcast Video Monitoring and Alerting System for Mobile Users

Bin Wei  
Bernard Renger  
Yih-Farn Chen  
Rittwik Jana  
Huale Huang  
Lee Begeja  
AT&T Labs-Research  
180 Park Ave  
Florham Park, NJ 07932  
{bw, renger, chen,  
rjana, huale, begeja}  
@research.att.com

David Gibbon  
Zhu Liu  
Behzad Shahraray  
AT&T Labs-Research  
200 Laurel Ave S  
Middletown, NJ 07748  
{dcg, zliu, behzad}  
@research.att.com

**Abstract – We present a system for automatic monitoring and timely dissemination of multimedia information to a range of mobile information appliances based on each user’s interest profile. Multimedia processing algorithms detect and isolate relevant video segments from over twenty television broadcast programs based on a collection of words and phrases specified by the user. Content repurposing techniques are then used to convert the information into a form that is suitable for delivery to the user’s mobile devices. Alerts are sent using a number of application messaging and network access protocols including email, short message service (SMS), multimedia messaging service (MMS), voice, session initiation protocol (SIP), fax, and pager protocols. The system is evaluated with respect to performance and user experiences. The MediaAlert system provides an effective and low-cost solution for the timely generation of alerts containing personal, business, and security information.**

**Keywords: Mobile devices, multimedia processing, content repurposing, content adaptation, service platform, news monitoring, automatic speech recognition (ASR), multimedia messaging, alerting, notification.**

## 1 INTRODUCTION

Mobile devices are gaining ground in computing power, storage capacity, and communication capabilities. A few years ago, Personal Digital Assistants (PDA) could only be used to store a limited amount of personal and other information. Today, these devices support standard peripherals such as digital cameras and the Global Positioning System (GPS), and have the ability to play video in stored or streaming

modes. The processing power of mobile devices continues to advance according to Moore’s law. For example, the clock rate of the Intel XScale PXA 27x processor is 624 MHz. The network bandwidth, wireless, storage, and graphic capabilities are growing at even faster rates. Compact flash memory cards with capacities as large as 12 GB are now available and capacities as high as 4 GB are affordable. Hard disks for certain mobile devices can accommodate over 60 GB of data. In addition, the latest smart phones such as the O2 Xda II can be equipped with both wireless LAN (WiFi) cards and GSM/GPRS capabilities. Mobile devices are now more than just digital organizers of yesterday and have evolved into powerful personal multimedia and communication devices. Consequently mobile devices are becoming so ubiquitous that they are literally part of the fabric of our lives and are so intuitive to use that we hardly notice them.

The storage capacity and multimedia presentation capabilities of mobile devices enable users to carry large collections of information with them. However, new information continues to become available on a daily basis. To gain access to such information in a timely manner, mobile users need to rely on the communications capabilities of these devices. Automated content processing and notification mechanisms are also needed to accommodate the capabilities of mobile devices and to be able to sift through large amounts of information.

We believe that the promise of mobile devices can be fully realized through a well-planned service architecture that can connect all relevant devices together and provide a common platform to develop and

deploy value-added services to meet time to market (TTM) and time to volume (TTV) requirements. We believe that this rapid evolution in mobile device technologies mandates a flexible service platform that can adapt information to new devices and access protocols easily.

TV broadcast news programs are a major source of information to TV viewers because they inform them what is happening today. In particular, mobile users want to be informed and in many cases they want to be informed as soon as new content is available. Thus, content that can be sent to mobile devices expeditiously would be desirable to mobile users. Our focus in this study is to automatically extract relevant video segments from broadcast news programs according to user's interests and to make the video content accessible to users through wired or wireless devices.

In this paper, we investigate how to build a service platform that would address the need for timely dissemination of information contained in broadcast news programs to mobile users. Our goal is to provide a common multimedia processing and alerting platform that enables innovative services to be deployed quickly. We believe that the novelty in this paper lies in our use of automated content processing techniques such as multimodal story segmentation combined with personalization both with respect to content selection and to content delivery options. This content preprocessing then enables the alerting platform to perform flexible content adaptation to provide users with the highest quality presentation given bandwidth and device capability constraints. The combined implementation of automated content processing and content adaptation results in a richer set of services for the end user than could be achieved if the separate system components were presented in isolation. Content adaptation and alert dissemination must be general purpose, handling any type of input and supporting a wide range of output devices. On the other hand, application-aware content processing can improve the user experience in restricted domains. For example, rather than just down-sampling video or extracting key frames, systems for delivering broadcast news content can leverage the closed caption text and filter the extracted key frames to produce a concise representation. This pre-processed content can then be further adapted for specific devices as necessary. In the rest of the paper, we first provide a general framework for media processing and alerting services. We then describe the architectural components for enabling such a system. The implementation section then describes system details and typical usage scenarios followed by a system evaluation from the perspective of performance and user experience. We give an overview of related work. After a brief discussion of future trends and some existing issues, we conclude the paper.

## 2 MEDIA ALERTING SERVICES FRAMEWORK

In this section, we consider a general framework for media alerting services and introduce the architectural components that are detailed in Section 3 for a broadcast video monitoring and alerting system.

### 2.1 A HIGH-LEVEL ABSTRACTION

There are three basic components in any alerting service: media acquisition, alert construction, and alert delivery. First, a mechanism to obtain media sources must be available. Media sources may be from public channels or private channels and may be presented in various forms, such as text, images, audio, video or any combination of these. Second, media processing mechanisms must be available to extract information from the media content. Third, alert information must be delivered to target users within a certain time period. The delivery mechanism depends on the devices accessible to end users.

Figure 1 presents a high-level logical framework of media alerting services. This includes the types of media sources that are used, how content is acquired and processed, and the target devices or protocols that are supported. The common goal of any media alerting system is to obtain relevant content segments from media sources and to deliver them automatically, regardless of where users are and what devices they are using.

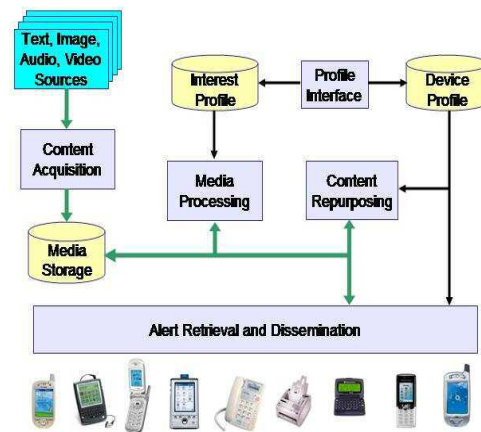


Figure 1 – General Framework for Media Alerting Services

The framework shown in Figure 1 consists of several components (a subset of them can be deployed for different implementations of media alert systems):

- **Content Acquisition:** The media sources can range from text, image, audio, to videos. Our focus is

primarily on video sources. Types of video feeds include terrestrial broadcast television, surveillance cameras, satellite broadcasts, IP streaming media. These feeds may vary widely in terms of content. In particular, the level of post-production processing in the media source has implications for media adaptation later on. Video processing techniques designed for unstructured video such as from a web camera or closed-circuit security feed may not be suitable for highly produced or structured television news material.

- **Media Storage:** After a piece of content is acquired, the media storage keeps both its initial raw format and successive transformations and adaptations for later use.
- **Profile Interface:** The profile interface collects user interests to help extract relevant media clips; it also collects device profiles and user configuration information that are then passed to the alert dissemination component.
- **Media Processing:** Media processing is used to detect relevant content, segment it and convert it into a form that is amenable for efficient processing later on. One common process employed is media segmentation, which is critical for alerting applications because long form content does not lend itself well to dissemination over messaging protocols such as SMS nor is it readily consumable on devices with limited user interface capabilities. By automatically segmenting the media based on topics, and adding this logical data structure to the multimedia database, we can rapidly produce smaller content units that are of a manageable size to satisfy bandwidth and device storage requirements.
- **Content Repurposing:** This process is needed to perform media adaptation to support a wide array of device types.
- **Alert Dissemination:** Given alert content and a list of recipient devices, this component is responsible for taking the repurposed content and delivering it via the appropriate access protocols. An important part of this module is handling the complexity of scheduling a very large number of alerts to meet stringent time constraints. This is a “Push” operation.
- **Alert Retrieval:** This component allows mobile users to query and retrieve alert content through different access protocols. This is a “Pull” operation.

Other components needed to produce a viable service that are not shown in Figure 1 include alert reporting/tracking and operations/systems support management functions.

## 2.2 ALERT TYPES

In addition to the accuracy and correctness of media processing for extracting alert content, the delay is a factor that affects the overall user experience. It mainly depends on how the media processing is scheduled. From the user’s perspective, there are several types of media alerts with respect to latency.

*Scheduled Alerts:* Users are alerted at specific times in the day of any alerts that may have occurred since the last scheduled alert time. This is suitable for users who prefer not to be alerted at odd hours but the users run the risk of not receiving alerts in a timely manner.

*Immediate Alerts:* In this case, the system runs in real-time and attempts to minimize the latency in delivering the alerts. It may attempt to deliver an alert as an event unfolds and as the media stream is still being acquired. For example, in the case of keyword spotting in broadcast video, it is a relatively easy task to build a system that has low latency from the time the last closed caption character of the keyword was broadcast to the time that the user gets notified. Topic spotting would be more difficult, e.g., looking for events such as earthquakes or corporate mergers. This is challenging to do while the text is streaming into the system. It also raises the possibility that the system might determine that the clip matches a user’s interest profile before the clip finishes airing.

*Predictive Alerts:* In this case, the system has out of band information, such as an electronic program guide (EPG), which allows it to determine that content matching a user’s profile will be available at some point in the future. Users would be alerted to “tune in” at the appropriate time. There are some well-known systems that use EPG and user interest profiles but they don’t involve alerting on various mobile devices. Additional flexibility can be obtained if we don’t require EPGs at all. We can analyze the content to determine if upcoming content will be of interest. For example: “Mt. St. Helens erupted and we will have a live video feed coming up shortly.”

The system we will describe next provides the basis for handling all three types of alerts. However the application we developed is basically for *scheduled alerts*.

## 3 A BROADCAST VIDEO MONITORING AND ALERTING SERVICE

We have built a media alerting service (called MediaAlert) that focuses on TV broadcast news as the

media source with the goal of delivering repurposed media alerts to a wide variety of mobile devices.

In this section, we describe the architectural components of the system based on the model presented in Section 2. MediaAlert is implemented by combining a media processing platform with a content delivery platform. The former is the eClips system [4], which is based on the Digital Video Library (DVL) platform [24]. The latter is the Alert Dissemination Engine built on top of the AT&T Enterprise Messaging Network<sup>SM</sup> platform [3]. This integration has several advantages:

- A large digital video library is available for users to search and retrieve video content from as early as the 1990s.
- Various formats of media content are derived from the original captured video including transcoded video, re-sampled audio streams, associated text either through closed captioning or ASR (Automatic Speech Recognition), and key frames. The derived content provides a rich set of resources for satisfying user's requests under various device constraints.
- The content delivery platform allows the use of different protocols to communicate with different devices. Thus the optimization of content repurposing can be achieved through the knowledge of the device profiles with the necessary transcoding from the appropriate media content.

The rest of this section will describe the system components of our media alerting service in detail.

### 3.1 CONTENT ACQUISITION

MediaAlert currently records selected broadcast TV programs from several broadcasters using satellite or cable feeds based on a pre-determined schedule and according to the interests of the target audience. The structured video feeds from broadcast television are then digitized, compressed and stored in a multimedia database (see Media Storage in Figure 1).

The database also holds high level metadata relevant to the content feeds including electronic program guide (EPG) information such as program title, air date, broadcaster, etc. The content acquisition subsystem could take the form of a bank of digital video recorders linked to a centralized content store.

The EPG data is too sparse to provide focused, concise, multimedia information that is relevant to the users. To address this, we automatically process the content of media streams, individually and collectively using multimodal processing techniques to build a rich content-based index for information retrieval, media segmentation, and media adaptation. Media processing is described next.

### 3.2 MEDIA PROCESSING

After the content is acquired, the media is processed to identify and segment relevant pieces of information. The details of the media segmentation techniques used are beyond the scope of this document and can be found in [1].

The results of the processing include high-level content features such as the locations of topic boundaries, topic keywords, and representative images for each topical content segment. Additionally, mid-level features are extracted as part of the processing and these are also maintained in the multimedia database. These include locations of scene boundaries, representative images for each scene (a.k.a. key frames), and an approximation of the dialog either in the form of closed caption text or the results of speech recognition (e.g., a word lattice or 1-best transcription.)

### 3.3 CONTENT REPURPOSING

The high and mid-level content features described above can be exploited to enable the alerting system to support a wide range of device types [1]. Examples of media adaptation will be discussed in detail in the sections that follow and can be seen in Figures 7 and 8.

The interest profile obtained through the profile interface (described in more detail in Section 4) is used to find content that matches the user keywords. The device profile dictates the kind of content that is compatible with the user devices. The content is then repurposed depending on the destination device and is sent using the Alert Dissemination Engine (ADE), which is described next.

### 3.4 ALERT DISSEMINATION

The Alert Dissemination Engine (ADE) is a middleware solution that allows limited mobile devices to communicate with each other and to securely access corporate and Internet content/services. It consists mainly of *gateways* and *servers* and is an instance of the AT&T Enterprise Messaging Network (formerly known as iMobile-EE [2]).

Gateways handle protocol specific interfaces to mobile devices and perform authentication, device profiling and session management functions. Servers, that perform the task of verifying device accounts and scheduling are replicas and can be load balanced for enhanced reliability. The system operates as a dynamic environment, with gateways and servers discovering and adjusting their capabilities dynamically. Both gateways and servers can be dynamically added/removed to the system. Interconnecting the gateways and servers is a message based

communication infrastructure using both point-to-point and multicast models.

### 3.4.1 Gateways and Devlets

The platform provides gateways that host devlets (protocol interfaces) for a multitude of protocols: email, http, pager, voice, fax, SMS, instant messaging. Multimedia messaging is supported through the use of an MMS gateway that retrieves the picture/video content from the Media Storage and sends it to an MMS service provider through an HTTP connection (see Section 4.4 for details). Being the access points for both end user/devices and external systems, the gateways perform session initiation and management functions; within each user session, we maintain an associated delivery context.

The message oriented devlets are based on a messaging framework that covers the protocol specific implementations. It provides a clean separation between the application messaging protocols (for example ‘pager’) and the network access protocols used to deliver the messages (for ‘pager’ we usually have the choice of SMTP or SNPP). The framework offers support for message delivery tracking, selective retry policies, delivery channel monitoring, outbound to inbound message matching, and resource/bandwidth allocation. More details of the messaging framework can be found in [3].

In the context of the notification engine implemented within the platform, only some protocols are used as delivery channels, in particular, the message based asynchronous protocols: mail, SMS, instant messaging (Jabber, AIM, etc), pager, voice, and fax. Their main characteristics are that a recipient can be uniquely identified through a permanent protocol specific address: email address, phone number, etc. As a consequence, it is possible to perform a ‘push’ of a message towards the end recipient.

### 3.4.2 Servers and Infolets

The components that make up a Server’s behavior are called *infolets*. Infolets implement the associated application logic and usually provide the access to one or more sources of information. Since the infolet output needs to be provided with respect to the delivery context established for the user session, the ADE offers a framework for information transcoding that can be used by the infolet provider but the ADE does not perform automatic transcoding itself. A particular class of infolets, called *services*, is dedicated for programmatically exposing functionality to external systems in contrast with other classes which provide content to the end user devices. The different components of the ADE are implemented as a set of

Web Services operating on top of the infrastructure. For further details on this platform please refer to [3].

Device	Description	Messaging Capabilities
1) PPC 2002 Smartphone	Siemens SX56	GSM/GPRS/SMS
2) MMS Phone	SonyEricsson T610	GSM/GPRS/SMS/MMS
3) Alphanumeric Pager	Skytel pager	Email/Paging
4) Numeric Pager	Metrocall pager	Email/Paging
5) Blackberry	Blackberry 6710	GSM/GPRS/Email/SMS
6) Cell Phone	Nokia 3310	GSM/SMS
7) PPC 2003 Smartphone	O2 Xda II	GSM/GPRS/SMS/WiFi/Bluetooth

Table 1 – User Device Descriptions



Figure 2 – User Devices

## 4 IMPLEMENTATION

MediaAlert supports the delivery of alerts with a range of media content including text, images, audio and video. The devices that are supported range from devices with limited display and processing capabilities such as pagers which can only handle limited text information or regular voice phones which can only receive phone calls, to PDA devices with video streaming capability. An assortment of devices currently supported by MediaAlert is shown in Figure 2. Device descriptions and messaging capabilities of these devices are shown in Table 1. Our purpose is to provide users with the flexibility to use any device. In the following sections, we present the implementation details of the prototype system.

## 4.1 USER PROVISIONING

The user can interact with the system in two ways. First, the user utilizes a Web interface to provision their devices and their interest profiles. Second, as new content is acquired that matches the user profile, the user will receive alerts on their selected devices. We will describe the provisioning component in this section. The alerting component will be discussed in the next section.

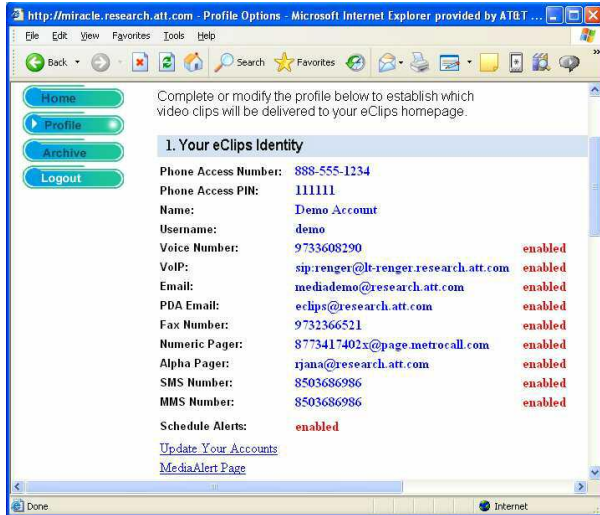


Figure 3 – User Device Profile

Devices/ Protocols	Voice	VoIP	Desktop Email	PDA Email	Numeric Paging	Alpha Paging	SMS	MMS
1) PPC 2002 Smartphone	X	X		X			X	
2) MMS Phone	X			X			X	X
3) Alphanumeric Pager						X		
4) Numeric Pager					X			
5) Blackberry	X		X	X		X	X	
6) Cell Phone	X						X	
7) PPC 2003 Smartphone w/ Wi-Fi	X	X		X			X	X

Table 2 – Device Protocols

Figure 3 shows the user device profile Web page where the user provisions his or her devices. We maintain a distinction between the user contact list and the user notification list. Users can choose a subset of their devices from the contact list to be used for notification purposes (these are shown as “enabled” in Figure 3).

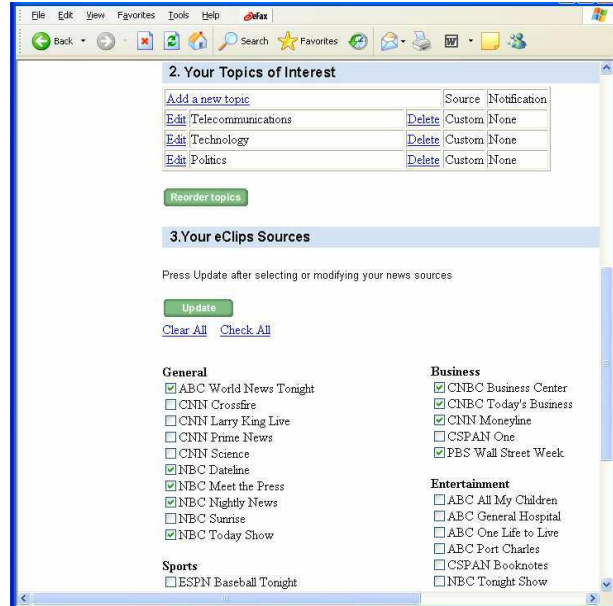


Figure 4 – User Interface Profile

As shown in Figure 3, the user can access the alert content via phone, VoIP, or other standard protocols. Audio can be delivered by making an alert call to a phone or to a VoIP (Voice over IP) client using SIP (session initiation protocol). Alternatively, the user can call a toll free Phone Access Number to hear the audio content to directly access the VXML interaction. In each of these cases, the Phone Access PIN is used to authenticate the user. Alerts can also be delivered to Email, Fax, Numeric or Alphanumeric Pager, and to SMS or MMS enabled devices. The protocols supported by the assortment of devices in Figure 2 are shown in Table 2. MediaAlert requires all user and device information to be pre-provisioned. Consequently, relevant user profile information is already available at the time of the alert generation in order to efficiently perform the dissemination.

Figure 4 shows the user interest profile Web page where the users provision their topics and associated keywords as well as the program sources for each topic. Each topic in the profile can have different keywords and can use a different subset of the available program sources. An alert is only sent if the keywords for a topic in the interest profile match content in the program sources associated with this topic. Keywords for topics are correlated against closed caption text, speech recognized audio segments and other metadata like EPG. This is described in more detail in [4].

## 4.2 ALERT GENERATION

After each TV news program is acquired and processed, the audio and video are transferred to the

Media Storage/Media Server which would ultimately stream the content to the devices that support streaming. The metadata and closed caption text are sent to the Index Server where the content is indexed.

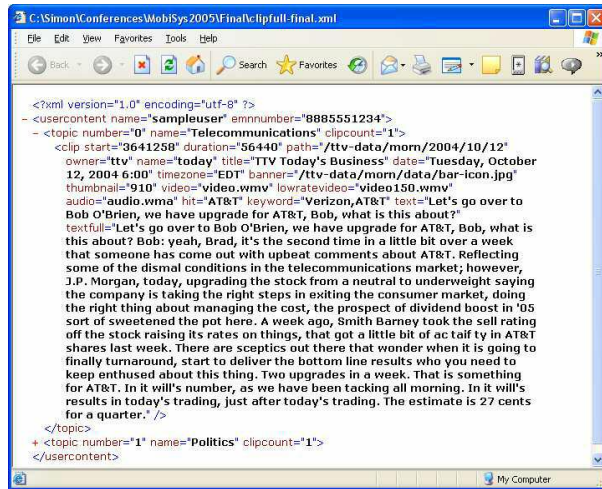


Figure 5 – XML Content

Various approaches can be used to match new content with user profiles. In the current implementation, a task runs at specific times in the day for each user and identifies new content that matches the program sources and keyword requirements for each topic. The first step in this task is to flag the alert content for each user. All new content since the last alert time is written out to an XML file for each user as shown in Figure 5. This file includes data from the index and the data from post-processing. The above mentioned task not only extracts the relevant clips but also repurposes the content and interfaces with the ADE via Web Services to send out the alerts automatically. This is described in the next two sections.

Other approaches could entail real-time word spotting of the closed caption text as the content is being acquired. The alerts could effectively be sent immediately. Of course, the clipping segmentation and indexing algorithms would be less effective since it would not have the advantage of analyzing the entire broadcast.

#### 4.3 CONTENT REPURPOSING

To support different protocols and devices, we repurpose the content to match the device requirements in the profile. The content repurposing is accomplished by using the relevant information in the XML file in Figure 5. For instance, the alert fax contains the full text of the clip (textfull attribute) whereas all the other alerts use the synopsis text (text attribute).

Content/Protocols	Voice/VoIP	Desktop Email	PDA Email	Fax	Numeric Paging	Alpha Paging	SMS	MMS
1) Callback Number					x	x	x	x
2) Hyperlink to Video		x	x					x
3) Program Icon		x						x
4) Program Name	x			x				
5) Date	x	x	x	x		x		x
6) Topic	x	x	x	x		x		x
7) Duration	x	x	x	x				
8) Thumbnail		x		x				x
9) Synopsis Text		x	x			x	x	x
10) Full Text				x				
11) Audio	x							

Table 3 – Content vs. Device Protocols

Table 3 shows the various content elements used for each device protocol. Note that most of the content elements come directly from the XML file while others are derived from the information in the XML file. For instance, the synopsis text is stored in the “text” attribute of the “clip” element in the XML file as can be seen in Table 4.

Content	XML Element	XML Attribute
1) Callback Number	usercontent	emnumber
2) Hyperlink to Video	clip	video
3) Program Icon	clip	banner
4) Program Name	clip	title
5) Date	clip	date
6) Topic	topic	name
7) Duration	clip	duration
8) Thumbnail	clip	thumbnail
9) Synopsis Text	clip	text
10) Full Text	clip	textfull
11) Audio	clip	[derived]

Table 4 – XML Content vs. Device Content

For text-only devices such as pagers and SMS devices, we provide the text content to the devices including a callback number. The text may be truncated to satisfy device requirements. Figure 6 shows the alert for an alphanumeric pager. The different elements from Table 3 are labeled accordingly. Note that for some devices/protocols, it is possible to send a hyperlink to the video. The media streaming server must be engineered to handle such video-on-demand requests based on the number of expected concurrent users. Our prototype uses Microsoft media streaming server. Other

video types can be created during the media adaptation process.

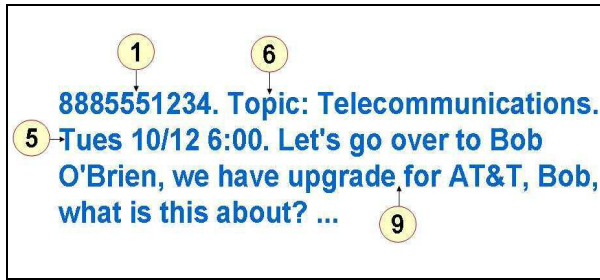


Figure 6 – Alphanumeric Pager Content

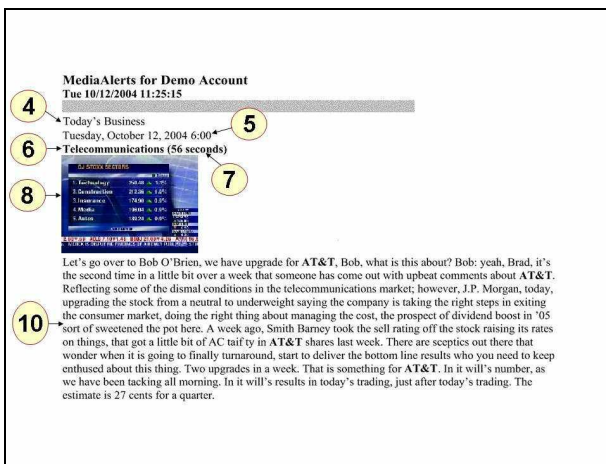


Figure 7 – Fax Content

The voice content is delivered to a phone or VoIP client via a VXML/SIP gateway via a remote dial option of the gateway. The audio alert content is created from the original video during media processing. The user can navigate the call using Touch-Tone commands or speech input. The prompts are played via TTS (text-to-speech) but the audio alert content is played back from the audio file. As previously discussed, the users can always dial the callback number at their preferred time instead of having the system call them.

For devices that can handle text and images, such as fax machines and MMS phones, a combined text and image representation is generated for delivery. Figure 7 shows a typical fax alert. For fax alerts, we generate html files that are sent to a fax broker which passes on the alert. For MMS phones, we compose the image portion and the text portion into an MMS message suitable for delivery by an MMS broker. Several clips can be concatenated and sent in one MMS message. For devices that can receive html formatted email, we send html email to the users directly. Figure 8 shows a typical desktop email alert. The thumbnail is a link to

the video so clicking the thumbnail will stream the higher bit rate video.

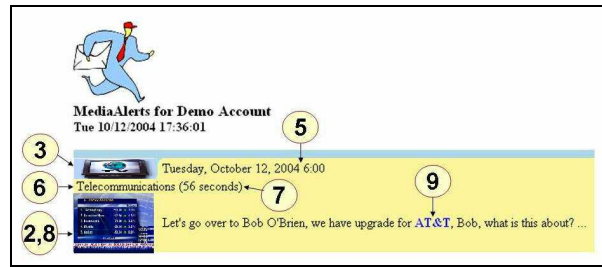


Figure 8 – Desktop Email Content

For video enabled mobile devices, such as PocketPC PDAs, we send a video link through the PDA email. The user can then stream the video by clicking on the link.

#### 4.4 ALERT DELIVERY

The Desktop Email, PDA Email, Fax, Numeric Pager, Alphanumeric Pager, SMS, and MMS content is sent via the relevant ADE gateways. The Voice and VoIP content is delivered to the Phone Number or SIP address using the VXML/SIP gateway.

Since we use different devices which may be on different networks, the data transmission varies across different networks and protocols. We use two examples (MMS and Fax) to demonstrate the process involved in delivering alerts on various networks.

Each MMS phone registered for alert delivery is identified uniquely in our system by its phone number and is mapped to a particular user. Our MMS gateway interacts with an MMS service provider using an HTTP connection to their server. Typically, an MMS provider hosts an MMSC gateway and maintains connections to cellular carriers globally [5].

Unlike the other gateways in the ADE, the MMS gateway repurposes the content and sends out the alert. For the other gateways, the content is repurposed and is passed as plain or html text or as an html file to the destination gateways. In the case of the MMS gateway, the URL of the XML file is passed. The MMS gateway parses this XML file to locate all the images, text, and other content elements relevant to an MMS alert. It then retrieves these elements from the Media Storage and then sends the MMS message out to the MMS service provider.

For the fax gateway, we send email to a fax broker called eFax. First, we enhance the gateway to support email with attachments. Second, we solve the issue of accessing images inside the firewall. The fax gateway first receives html content, which contains image URLs inside our Intranet that eFax cannot directly access.



There are several ways to solve this, such as opening a port, using a reverse proxy, or converting html to other document formats. We decided to convert the html file to a PDF document that we can email to eFax as an attachment. The end result is a fax alert that contains text and images as shown in Figure 7.

## 5 SYSTEM EVALUATION

With our implementation, we evaluated the performance of the system and the user's experience. The performance was evaluated by measuring the execution time of various components under different conditions. Together with the user experience, the study helps us to better understand the system behavior and to improve and optimize the system.

### 5.1 PERFORMANCE

Since MediaAlert consists of media processing and alert dissemination as two relatively independent components, we treat the two separately in the performance studies so that we can get more detailed data for each component. First, we discuss the performance on media processing. Then, we discuss the alert dissemination and message delivery component.

Content Processing Steps	Time (sec)	Percentage
Video transcoding	38	42%
Audio transcoding	6	6%
Caption text/JPEG processing	7	8%
Caption alignment with speech	39	43%
Other	1	1%
<b>Total</b>	<b>91</b>	<b>100%</b>

Table 5 – Content Processing Times (100 second video)

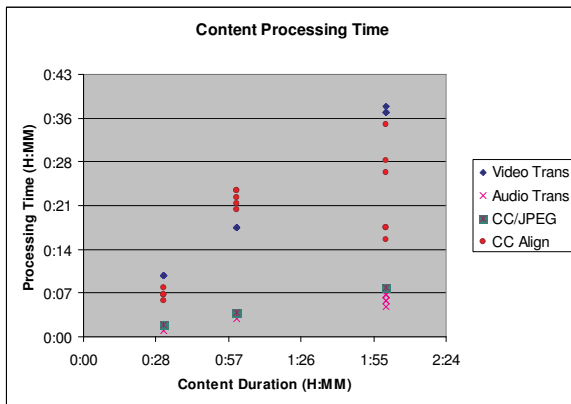


Figure 9 – Content Processing Time vs. Content Duration

### 5.1.1 Media Processing

Media processing occurs after the video acquisition and has two components: content processing and clip generation. The content processing applies to the full video and the clip generation applies to the portions of the video content that are sent as an alert.

The full video processing steps include: video transcoding, audio transcoding, closed caption (CC) and JPEG processing, and CC alignment with speech. Table 5 lists the elapsed time and the percentage of the total time in each step for a 100 second video. The total processing time is 91 seconds or 91% of the source video time which means that the current system can handle video streams in real-time. Figure 9 is the data from the acquisition of 17 video streams, from 30 minutes to 120 minutes. In this Figure, the transcoding time depends almost linearly on the length of the source video. Video transcoding takes about 31% of the video time; audio transcoding is 6%. CC/JPEG processing is 7%. The time for CC alignment processing depends on the content, ranging from 13% to 39% of the video time. The acquisition is done on a 1GHz dual-processor Pentium III for one broadcast input.

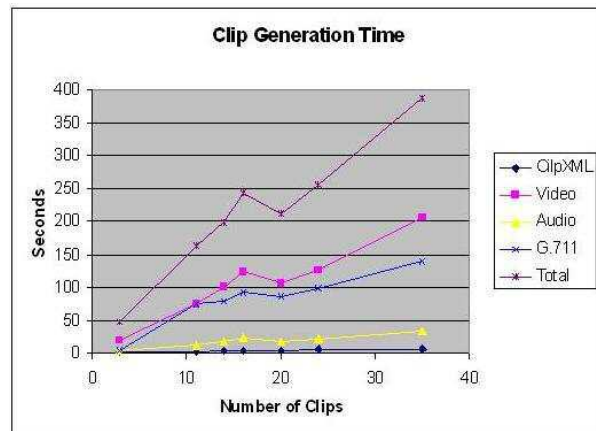


Figure 10 – Clip Generation Time vs. Number of Clips

Figure 10 is the timing information for clip generation which includes generating the clipping information XML file, extracting video and audio clips, and obtaining uncompressed audio (G.711) for use with the VXML audio playback. We have performed these tests by varying the number of clips from 3 to 35 by adjusting the time window of the search. Among the processing steps, clipping video is the most time consuming part. Uncompressing audio also takes a relatively long time because we use off-the-shelf tools directly without optimizing for our particular purpose. A portion of the same data is also available in Table 6, which gives the average clip duration and the average clipping time to process one clip. From this Table, we

see that clips with an average duration of 87 seconds long take 13 seconds to clip on average which is 15% of the clip duration time. This implies that the system is very efficient in processing clips. Compared with the content processing time, the clipping processing time is much less. The data was collected on a 2.4 GHz dual-processor Pentium machine.

Clips	Average clip duration (sec)	Average clip processing (sec)	Processing/Duration
3	105.67	16.00	15%
11	96.00	14.82	15%
14	87.57	14.21	16%
16	87.63	15.19	17%
20	80.40	10.60	13%
24	79.63	10.63	13%
35	69.26	11.03	16%
Avg.	86.59	13.21	15%

Table 6 – Clip Duration/Processing Times

### 5.1.2 Alert Dissemination

In this section, we evaluate the performance of the alert dissemination component system for email alerts. Figures 11 and 12 show the measurements at the client side from the time when the alert injection requests are sent to the time when the EMN server responses are received. This is the time that is required to satisfy an alert specification and output a time slotted schedule. This schedule determines when each alert is disseminated. The client machine simulates alert requests with multiple threads at the recipient mailboxes.

We used an EMN testing framework, which is capable of simulating multiple threads of alert requests. In Figures 11 and 12, the number of threads is configured as 1, 2, 4, 8 and 16. The number of endpoints (number of recipients of an alert) that each client thread generates varies from 32, 64, 128, 256, 512 to 1024. To show performance measurement results of these configurations, we use representative endpoints of 32, 64 and 128 as shown in Figure 11. The results with endpoints from 256 to 1024 are shown in Figure 12. For each figure, the five white bars on the left indicate the number of threads being used. From left to right, the number of threads is 1, 2, 4, 8 and 16. These white bars are for the cases when there is only one EMN server. Similarly the five dark bars indicate the cases when there are two EMN servers. Note that due to the distributed nature of the platform it is possible to load balance the alert processing via identical EMN servers and therefore accommodate a larger number of users. As a rule of thumb, new servers should be started automatically if the sustained load exceeds a certain safety threshold, thereby maintaining scalability. The increasing number of endpoints shows the workload change. We put the one-server and two-server cases side

by side for ease of comparison. In this measurement, the client uses a 1.4GHz dual-processor Pentium machine running a Linux 2.4 kernel. Gateways and EMN Servers use 2.4GHz quad-processor Pentium machines with a Linux 2.4 kernel.

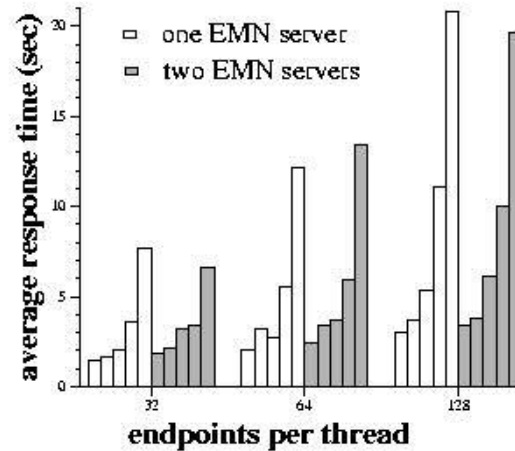


Figure 11 – Average Server Response Time through EMN ADE: Part I

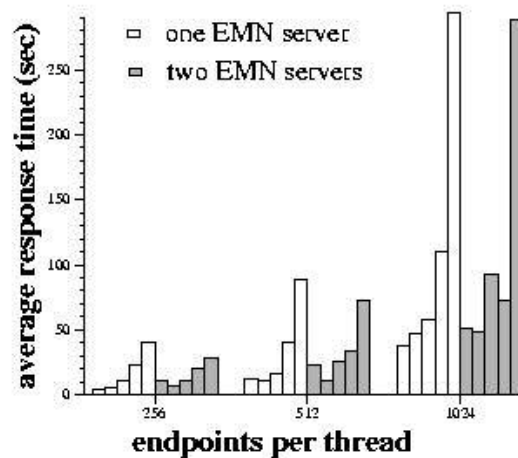


Figure 12 – Average Server Response Time through EMN ADE: Part II

Figures 11 and 12 indicate that the server response time increases when the number of requests increases. As we can see from the figures, the system can process alerts that include 16384 email recipients (16 threads with 1024 endpoints each) in less than 300 seconds with either one or two servers. The actual end-to-end email dissemination time took less than 700 seconds. However, even for large requests, the two-server configuration with two JMS queues has similar performance as the one server case. This indicates that the system bottleneck is not at the server engines.

Because all the requests need to access the common Oracle database, the database access appears to be the bottleneck in the system.

Based on the data we collected, media processing tends to take longer than alert dissemination, especially for a small number of users. To balance the system performance, we need to first reduce the media processing time. The number of Broadcast News channels is limited by the number of available channels but the number of users can be increased arbitrarily. To be cost effective, a good balanced system relies on the scale of the service.

The final delivery time from the server to an end device, which depends on the access mechanism available on the device, can also add a significant delay to the alerting process. Let's take the delivery of MMS messages to a mobile phone as an example. In one experiment, a one-character MMS message took 49 seconds. For messages with multiple text and picture components, it took 90 seconds for 20KB of data and 114 seconds for 40KB of data. The time fluctuations between different runs can be substantial. Sometimes, MMS message delivery can take more than 10 minutes. Because the messages have to traverse different networks which are beyond our control (the MMSC handles messages in a store and forward mechanism), the delay can be very long and unpredictable. In such extreme cases, we need a mechanism to detect the abnormality immediately and to switch to an alternative device for the user.

## 5.2 USER EXPERIENCE WITH MEDIAALERT

In this section, we discuss our preliminary experience with MediaAlert on several mobile devices.

As shown in Table 2, the Blackberry 6710 is a versatile GSM/GPRS mobile device with voice, email, SMS, and paging capabilities. New email is pushed automatically to a Blackberry device without the mobile user having to access a mailbox explicitly. Also, the Blackberry can handle large amounts of email text by simply requesting for more email from the Microsoft Exchange server. This makes the Blackberry an ideal device for receiving comprehensive text alerts.

Initially, we included the callback number only for protocols with limited text abilities (such as paging and SMS). For example, we cannot send more than 140 characters (and sometimes 100) in an SMS message to most phones. The callback number is included so that users can call back to get the complete audio clip. On the Blackberry 6710, any sequence of digits (such as 9735551212) that looks like a phone number is clickable and the clicking initiates a call to that number. This is true for both SMS messages and email messages received on that device. This feature makes it very

convenient for retrieving audio alerts and has increased our voice usage of the Blackberry 6710. It has also prompted us to add a callback number to email alerts. Similarly for an MMS message, the MMSC sends the user a notification that a new message is waiting. The receiver can then download the message immediately or download it later (user pull rather than a user push request). Although an MMS message can encompass a wide range of content types, it is a logical extension of SMS, making it easily adoptable for today's generation of mobile users. Another advantage of MMS for this kind of alert is that the message is delivered as a single multimedia message and not as a text message with attachments. This minimizes the steps that the user has to take to retrieve the content.

On the recent Xda II device (Smartphone with PocketPC 2003, WiFi, and Bluetooth), we experimented with low bit rate streaming video alerts using the Microsoft Media Player. Due to copyright issues, we conducted our experiments on an internal Lab server. A lower bit rate of 150 kbps was used for streaming content to these PDAs. Overall, the mobile user is able to watch the streamed video comfortably without much data loss. As most 2.5G and 3G wireless networks are still limited in bandwidth, we expect the retrieval of high quality video alerts to become feasible first on WiFi networks. Hopefully, the convergence of 3G and WiFi/WiMax on a new generation of cell phones will allow the users to retrieve videos with varying degrees of quality depending on the cost/network availability.

The above comments on user experience are based on *scheduled alerts*. Other types of alerts can be handled in a similar fashion or treated differently. Further user experience study would be needed to ensure that we would be meeting user needs. It is also possible in the future to extend the user profile to incorporate other attributes like location, presence and context.

## 6 RELATED WORK

Palmer et al describe a system [7] where speech recognition and machine translation techniques are applied to TV news programs. This work goes into multi-lingual capabilities where real-time information is detected and matched against subscribed keywords from English and Arabic news sources. While the demonstration explains how to extract interested content automatically and display it in a cross-lingual environment, content repurposing and the problems of alerting mobile users are not addressed. Sumiya et al. present a system [8] to bring broadcast news content to the Web by using metadata with a zooming feature to alter the level of detail being viewed. This work focuses on repurposing the content for Web browsing. There has also been some research conducted in topic detection

from TV programs, in particular, robustness in topic boundary identification amidst transcription errors [9].

Google Alerts are email updates of the latest relevant Google results (Web, news, etc.) based on the user's choice of query or topic. Our proposed platform takes this one step further to incorporate live TV broadcast feeds. It will be an interesting experiment to see the effectiveness of MediaAlert when compared to GoogleAlerts [10] or CNN Alerts [11] when monitoring a developing news story. The key features that differentiate our system from other commercial services are automated video content acquisition and monitoring, topic segmentation, and media content adaptation for mobile devices. In addition, we also support speech recognition (using a two hundred thousand word vocabulary automatic speech recognizer). There are a number of products that offer mobile-enterprise focused alerting and notification. IBM has also announced an additional package to its Websphere Everyplace Access Suite [12]. These additions depend on vendor specific products like Sametime (now Lotus Instant Messaging) and presence platforms [13].

Microsoft has recently announced .NET alerts - a message and notification service that can be used to deliver customer communication to desktops, cellular phones and personal digital assistants [14]. This solution is more powerful in the Intranet since it makes use of Microsoft related products deployed within an enterprise, for example, Microsoft Active Directory, SQL Server notification services and .NET Framework. MediaAlert presents an extensible architecture ready for integration with existing enterprise software in a standardized and vendor agnostic manner. In our work, we have concentrated on the notification aspects, trying to offer an open generic common interface to alert management software that incorporates the business logic, workflow and decision aspects.

The media processing work in our approach is based on earlier work [15]. Our previous work [16] aimed to have the ability to transfer emergency video clips captured from a mobile user to other relevant users in a timely fashion. In this case, there was no media analysis, voice or text processing involved.

## 7 DISCUSSION

A whole new category of converged mobile devices will soon be upon us. PDAs will be combined with cell phones and radios and televisions. We will not predict the winners and losers of this device war but we believe that there will be more need for all types of alerts in order to best utilize the capabilities of these converged devices. Unless the system or network takes over more of the work, the user will be overwhelmed with choices and inhibited by the User Interface to such an extent that

the entire concept of converged devices will falter. These combined devices will also enable users and designers to skirt the edges of copyright law in ways that will allow new services to exist. In the realm of mobile devices and expanding on the concepts of immediate and future alerts, we can discuss the convergence of cell phones and digital television as it impacts alerting services.

Next generation cell phones will enable users to receive the content from HDTV broadcasts [6]. Wireless handsets could receive hundreds of high-definition digital channels, not over the cell network but from the HDTV broadcast network or from a separate satellite network that could carry hundreds of digital channels. The article in [6] claims and we concur that this technology would be "event-driven". We expect this capability to open up new "event-driven" alerting possibilities for the mobile user, beyond the simple notion in the article about watching specific sports or news shows.

We anticipate a converged future for devices, where they will not necessarily be all-in-one devices but they will all communicate and interoperate. As an example scenario, imagine that an alert is sent for a particular event on news or TV. If the event occurs, the user would be notified and could instantly watch the clip of interest. The main advantage of this scenario over the previous alerting scenario is in the realm of copyrights. If we wanted to deliver a service that would create video clips of interest for a user, we would be required to get licenses from all the providers of multimedia material. That proves to be a daunting task and has held up much of the useful online multimedia applications. If instead of having to capture and rebroadcast the material, we could simply monitor it and as we locate it send that information out to users who could then watch it live. In this way, we can provide the alerting service without violating copyright laws. This scenario opens up other possibilities for the user. As an example of the converged future, there could be a video recording device at home that could be notified over the Internet that this segment was of particular interest and needs to be recorded. The difference between this and an automatic program recorder is that our service would only be recording the specific clips that were of interest to the user. Once they were recorded at the user's home, that user could watch them at leisure. The user could even have the clips streamed from the video recording device directly to the video-enabled mobile handset. We envision getting support from the handset makers and the cell service providers because they are both desperate to generate new service revenue and have invested billions of dollars in upgrading to high speed digital networks. This would certainly have appeal to the handset makers as a way to enhance the capabilities of their phones. The streaming to the device

would be of great interest to the cell service providers as a way to generate new revenue, particularly if this were set up as an emergency alerting system for public service workers like police, fire and first aid.

## 8 CONCLUSIONS

An automated system for multimedia content monitoring and alerting was described. Unlike existing systems that rely on manually generated clips/stories, this system uses multimodal story segmentation algorithms to find and isolate short relevant segments of video within a video program. Moreover, it relies on multimedia processing techniques to repurpose the content for delivery to a range of mobile devices with a wide range of presentation capabilities from text-only to full-motion video. The repurposing process is not only aimed at producing a representation of the information that accommodates the limitations of the device at hand, but is also aimed at creating alternative presentations that significantly reduce the amount of bandwidth needed to deliver the information.

Targeting this application mainly for mobile devices and using it to generate alerts require higher selectivity in choosing the information and better isolation of the information. Such high selectivity is critical for preventing the generation of false or trivial alerts. This can be achieved by a combination of better information processing/retrieval techniques and good judgment on the part of the user in providing the right combination of keywords and phrases for each topic of interest. Our experience indicates that imposing additional proximity constraints in the retrieval process is an effective way for increasing the relevance of extracted content and reducing the possibility of false alerts. Another issue is the generation of multiple alerts with very similar information from different sources. We are working on utilizing effective similarity measures to detect and eliminate such cases.

The combination of automatic media monitoring and alerting not only provides an effective system for timely delivery of personalized information and timely business information, but is also an effective way for automatically discovering and delivering security related information.

MediaAlert has been designed to be a carrier grade solution both in terms of architecture scalability and flexibility to innovate and deploy new services rapidly. The media processing engine can ingest a large number of simultaneous real time broadcast quality feeds while the dissemination engine can handle a large number of concurrent alerts to meet stringent timing requirements.

## REFERENCES

- [1] D. Gibbon, L. Begeja, Z. Liu, B. Renger, and B. Shahraray, "Multimedia Processing for Enhanced Information Delivery on Mobile Devices," *Emerging Applications for Wireless and Mobile Access*, MobEA II, New York, May 18, 2004.
- [2] Y. Chen, H. Huang, R. Jana, T. Jim, S. Jora, R. Muthumanickam, and B. Wei, "iMobile EE – An Enterprise Mobile Service Platform," *Wireless Networks*, Vol. 9, No. 4, pp. 283-297, July 2003.
- [3] S. Jora, R. Jana, Y. Chen, M. Hiltunen, T. Jim, H. Huang, A. Singh, R. Muthu, "An alerting and notification service on the AT&T Enterprise Messaging Network," *Proceedings of IASTED – Internet and Multimedia*, Feb 21-23, Grindelwald, Switzerland, 2005.
- [4] L. Begeja, D. Gibbon, K. Huber, A. Lee, Z. Liu, B. Renger, B. Shahraray, M. Zalot, G. Zamchick, "eClips: Customized Video Clips," talk at WebSummit 2001.
- [5] 3GPP2 Multimedia Messaging System: MMS Specification Overview  
[http://www.3gpp2.org/Public\\_html/specs/X.S0016-000-A.pdf](http://www.3gpp2.org/Public_html/specs/X.S0016-000-A.pdf).
- [6] NY Times, "Coming Soon to Your Pocket: High-Definition TV Phones," October 21, 2004.
- [7] D. Palmer, P. Bray, M. Reichman, K. Rhodes, N. White, A. Merlino, and F. Kubala. "Multilingual Video and Audio News Alerting," *Human Language Technology Conference*, May 2004.
- [8] K. Sumiya, M. Munisamy, and K. Tanaka, "TV2Web: Generating and Browsing Web with Multiple LOD from Video streams and their Metadata," *The Thirteenth International World Wide Web Conference*, May 2004.
- [9] Richard Schwartz, Toru Imai, Francis Kubala, Long Nguyen, and John Makhoul, "A maximum likelihood model for topic classification of broadcast news," in *Proceedings of Eurospeech '97*, Rhodes, Greece, Sept. 1997, pp. 1455–1458.
- [10] Google Alerts - <http://www.google.com/alerts>
- [11] CNN Alerts - <http://www.cnn.com/EMAIL/>
- [12] T. Olavsrud, "Websphere adds wireless notifications, messaging", InstantMessagingPlanet.com, May 2003, <http://www.instantmessagingplanet.com/wireless/article.php/2201571>
- [13] Sametime – Lotus Instant Messaging, <http://www.lotus.com/products/lotussametime.nsf/wdocs/homepage>
- [14] Microsoft .NET alerts v6.0 - <http://msdn.microsoft.com/msdnmag/issues/03/12/XMLFiles/default.aspx>
- [15] L. Begeja, B. Renger, D. Gibbon, K. Huber, Z. Liu, B. Shahraray, R. Markowitz, P. Stuntebeck,

- “eClips: A New Personalized Multimedia Delivery Service,” *Journal of the Institution of British Telecommunications Engineers (IBTE)*, April-June 2001.
- [16] B. Wei, Y. Chen, H. Huang, and R. Jana, “A Multimedia Alerting and Notification Service for Mobile Users,” *Emerging Applications for Wireless and Mobile Access, MobEA II*, New York, May 18, 2004.
- [17] Y. Chen, H. Huang, R. Jana, S. John, S. Jora, A. Reibman, and B. Wei, “Personalized Multimedia Services Using a Mobile Service Platform,” *IEEE Wireless Communication and Networking Conference*, Orlando, FL, March 2002.
- [18] S. Jun, Y. Rong, S. Pei, and S. Song, “Interactive Multimedia Messaging Service Platform,” *Proc. ACM Multimedia*. pp. 464-465, Berkeley, CA, November 2003.
- [19] Eric Manning, “The Utility Model for Multimedia Servers and Networks: A Retrospective,” *Keynote Address, ICCIT*, December 2002.
- [20] Microsoft 9 Series Digital Rights Management, <http://www.microsoft.com/windows/windowsmedia/9series/drm.aspx>.
- [21] Microsoft Windows Media-on-Demand Producer, <http://www.microsoft.com/technet/prodtechnol/nets/how/downloads/wmodp.msp>.
- [22] D. Gibbon, L. Begeja, Z. Liu, B. Renger, and B. Shahraray, “Creating Personalized Video Presentations using Multimodal Processing,” *Handbook of Multimedia Databases*, Edited by Borko Furht, CRC Press, pp. 1107-1131, June 2003, ISBN 0-8493-7006-X.
- [23] Han Chen, Kai Li, and Bin Wei, “Memory Performance Optimizations For Real-Time Software HDTV Decoding,” *Journal of VLSI Signal Processing, Special Issue on Media and Communication Applications on General Purpose Processors: Hardware and Software Issues*, accepted for publication, 2005.
- [24] Gibbon, D. *et al* (1999). “Browsing and Retrieval of Full Broadcast-Quality Video,” *Packet Video*, NY, NY, April 25.