A Mobile Phone-based Context-aware Video Management Application

Janne Lahti, Marko Palola, Jari Korva, Utz Westermann[*], Kostas Pentikousis and Paavo Pietarila

VTT Technical Research Centre of Finland, FI-90571 Oulu, Finland
e-mail: firstname.lastname@vtt.fi
[*] University of California, Irvine, CA 92697-3425, USA
e-mail: westermann@acm.org

ABSTRACT

We present a video management system comprising a video server and a mobile camera-phone application called MobiCon, which allows users to capture videos, annotate them with metadata, specify digital rights management (DRM) settings, upload the videos over the cellular network, and share them with others. Once stored in the video server, users can then search their personal video collection via a web interface, and watch the video clips using a wide range of terminals. We describe the MobiCon architecture, compare it with related work, provide an overview of the video server, and illustrate a typical user scenario from the point of capture to video sharing and video searching. Our work takes steps forward in advancing the mobile camera-phone from a video playback device to a video production tool. We summarize field trial results conducted in the area of Oulu, Finland, which demonstrate that users can master the application quickly, but are unwilling to perform extensive manual annotations. Based on the user trial results and our own experience, we present future development directions for MobiCon, in particular, and the video management architecture, in general.

Keywords: mobile camera phone application, mobile video management, video metadata and digital rights management

1. INTRODUCTION

Mobile phone manufacturers are increasingly adding new models with multimedia support and most modern medium- to high-end cell phones come with an integrated audio/video player, a camera to capture still and moving pictures, and some media editing software. The "coolness factor" fuels the popularity of mobile camera phones (MCP) and increases the volume of user-created media content. MCPs can record videos of up to several minutes, depending on the amount of memory available. Videos cannot be reasonably stored permanently on the mobile device due to the limited memory capacity available. Thus, users are in need of services that allow them to store their videos somewhere else, also because users want to create collections of their clips and share them with friends and relatives. Some will even opt for making part of their digital content available from a web site, or add it to their blog entries. Although none of these is news, nevertheless, there is very little automation in this process, and mobile applications for video management are not as commonplace as one might think despite the fact that MCPs have been on the market for quite some time.

The main problem of mobile content management is two-fold: how to automate permanent video clip storage, and how to do so in a way that is user-friendly, allows for easy clip lookups, and enables the user to share videos with others. Even though there are standardized ways to share videos over the cellular network, including the Multimedia Messaging Service (MMS) [1] and the upcoming IP Multimedia Subsystem (IMS) [2, 3], they must all be supported by the mobile operator, require infrastructure expenditures, and allow mainly for point-to-point video sharing. Moreover, a common characteristic of these services is that they do not include a solution for video management, and do not take full advantage of the context information available at the point of video capture. Our development effort takes a different approach and attempts to (a) enable the MCP to serve as an valuable video production tool in addition to being a video consumption channel, and (b) permit users to share videos with others irrespective of the level of network operator support for multimedia services.

Of course, one can argue that an MCP is not the ideal device for video management because of the limited CPU, memory and input/output capabilities, battery power consumptions considerations, and the diverse video formats and

Multimedia on Mobile Devices II, edited by Reiner Creutzburg, Jarmo H. Takala, Chang Wen Chen,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6074, 60740O, © 2006 SPIE-IS&T · 0277-786X/06/$15

other technologies supported by the operating system. On the other hand, a MCP application can assist users to describe a video by associating it with annotations about, for example, where and when a video clip was shot, and who should be able to access it. The MCP application may have at its disposal different kinds of context information including sensors, a GPS device, a calendar service, and other Internet resources. All these are valuable sources for metadata that can be used to characterize the digital object and possibly uniquely identify it. Metadata plays a critical role in managing video content to the extent that Sarvas *et al.* [4] argue that it is not possible to manage media content effectively without them. The value of the metadata is illustrated in the following section, which presents a typical scenario where the user captures a video clip, associates it with metadata, and later searches for this particular video within her collection using an Internet-connected desktop PC. This paper details a solution to the problems related to user-created media content and annotation, and addresses some issues pertaining to sharing and storing videos using MCPs, and the challenges presented to the mobile video management system. The rest of this paper is organized as follows. Sections 3 and 4 discuss the implementation objectives of such video management system, and its architecture, respectively. Section 5 summarizes the results of a real user field trial evaluation, and Section 6 presents our current work in progress along with some future directions, which address certain shortcomings found in the first generation of the system. Finally, Section 7 reviews related work, and Section 8 concludes the paper.

2. A TYPICAL USER SCENARIO

We exemplify the use of the mobile video management system with a scenario involving a tourist in a sightseeing tour. The aim in this section is to provide an overall understanding of how the system works, essentially a high-level user-centered view of the requirements, before presenting, in the following section, the main objectives.

Figure 1 portrays Alice while on vacation in Oulu, Finland. Alice spends some time walking around the city center area, visiting several attractions, when she notices the Oulu cathedral clock tower from a distance. After approaching the church, she uses her MCP to record a short video of the cathedral, its clock tower, and the surrounding area. Alice uses MobiCon, an MCP application, which allows her not only to capture the video, but also to upload it to a server and notify her friends about recent findings during her tour of Oulu.

Figure 1: Alice records a video of the Oulu cathedral (left), annotates it with the term “Object > Buildings” (center), and later is able to search for it using the web interface of the Candela video management server (right).

After recording the video, Alice annotates it by selecting the prefedefined concepts “Holiday” and “'Buildings” from the application metadata menu, and enters two keywords (“Church” and “Oulu”) to describe the video clip more accurately. Alice saves the clip locally on the mobile phone memory card. Shortly afterwards, Alice decides to share the video with Bob, a friend of hers who is very interested in church architecture. With MobiCon, this is a simple process: Alice selects the video clip using a menu, chooses Bob from her contact list, and grants him the rights to watch the clip. She can subsequently upload the clip to the video server and MobiCon will automatically send a text message using the Short Message Service (SMS) to Bob with information on how to access the video. After receiving the text message, Bob can watch the video by opening its URL straight from his mobile phone.

After returning from her holidays, Alice can login to the video server using a web interface and access her large clip collection. Mobile videos tend to be short in length and relatively “focused” on a single theme. Unfortunately, Alice quickly finds herself having to deal with hundreds of videos with not-so-descriptive file names. Trying to locate the "church video" in the entire collection can quickly become tiresome and this is where video metadata annotation proves helpful. Although Alice is not likely to remember the exact time or date that the video was captured or uploaded to the video server, she does remember where she was and what the main theme of the video was. Searching for "Oulu" limits the set of possible videos and searching for "church" narrows the size of the search result set to a handful of clips. Once located, Alice can watch the movie clip delivered by the video management server in the most appropriate format for her platform.

3. IMPLEMENTATION OBJECTIVES

A mobile video management application has five main functions: video recording, metadata annotation, video storage, video sharing, and locating video clips in a collection. Any MCP application, such as MobiCon, should be robust and rich in functionality, yet easy to use and engaging, despite the restrictions imposed by the small display size and minimal keyboard. Moreover, application developers must pay attention to the way resources are used: network traffic should be minimized, battery power should be conserved when possible, and CPU and memory ought to be utilized with frugality. These restrictions come on top of the classic mobile phone application development nightmares (device incompatibilities, network application debugging, immature SDKs, and different operating system versions with undocumented bugs) making the development of an application like MobiCon challenging.

Video *recording*, the first function, is relatively straightforward to implement with vendor provided SDKs. However, the application should be robust during this phase and capable of handling critical events (including incoming phone calls and text messages). As illustrated in the previous section, video *metadata annotation* is necessary for searching stored clips in an efficient manner and is a central part of the design of MobiCon. We can identify three main issues that need to be addressed: (a) when shall the application collect the metadata, (b) which types of metadata should be stored so that locating the video will be easier later, and (c) how can the application acquire all this information in an elegant and robust manner. Practice indicates that, at least for home videos, the best time to annotate video clips is right after capturing them. However, at this point it is difficult to predict which information will be most valuable in locating the video later on. For example, for a resident of Oulu using MobiCon frequently while in town, the term "Oulu" will not be a good choice in order to discriminate between a large set of videos. For Alice, though, the same term will allow her to easily select all videos from her last visit to Northern Finland. Thus, one can argue that the best choice is to collect as much information as possible, and preferably do so in an automated way. On the other hand, this may lead to generating mostly low-level data or content features, such as exact dates and color histograms, which are neither easily recollected by most humans nor convey a direct meaning to them. Therefore, the application must allow the user to provide additional high-level information, and assist him in this by making the process as easy as possible.

The limited phone resources make *video storage* and *video sharing* particularly interesting problems to address. Permanent storage cannot be provided by the phone in a scalable manner: compared to other kinds of mass storage devices, such as hard drives, memory cards for mobile phones are still expensive and very limited in capacity. Users typically end up transferring their videos to their PC and then share them with others. This, of course, is not a truly mobile video management solution. An MCP application should use the network to store the videos and allow others to view them in an asynchronous manner. For video sharing, there are no universally supported media formats, and the device capabilities and the capacity of the access networks vary greatly. Because of this, the original video ought to be provided in several alternative formats using different encoding parameters. However, video transcoding is a computationally demanding process which cannot be performed in real time. Thus, alternative versions should be generated before the video can be retrieved, typically soon after a clip is uploaded. Furthermore, an important aspect in video sharing is the ability to control how the receivers use the shared video clips and limit redistribution, if necessary. One way of handling the video storage and sharing is to place the videos on a server. This way the MCP application needs to upload the video only once while remaining able to share it with an arbitrary number of recipients. The server can also take care of all security and computational intensive aspects of video storage, and enable the user to *search* for video clips later. In short, by making an MCP application directly pluggable to a searchable repository that takes advantage of the captured metadata leads to avoiding time-consuming and inefficient lookups of video clips based on filenames and file dates alone.

To sum up these requirements, MobiCon should be able to capture video clips using the internal mobile phone camera, assist the user to annotate each clip with metadata, store the clip to a remote video management server, permit him to share video clips with others and, last but not least, enable him to search large collections of video clips using a mobile phone, PC, or any other device with web access. Users should maintain full control of the entire video management chain, from source to destination, by tapping on a standards-compliant digital rights management (DRM) framework.

4. SYSTEM ARCHITECTURE

The VTT Candela system, named after the European ITEA project CANDELA (Content Analysis, Networked Delivery and Architectures) was developed as a solution for general video management. It includes tools for video creation, analysis, annotation, storage, search and delivery phases. The Candela system was originally developed for the personal home video domain but, due to its modular and tiered architecture, it can be used as a basis for different video management applications by reusing some of the generic components and adding domain-specific ones. An important part of the entire architecture is the MCP application MobiCon, which can be used both on a standalone basis and as part of the entire system, as shown in Figure 2. Before delving in the details of the MobiCon architecture, we briefly introduce the Candela system; more details about the Candela architecture are given in [5, 6].

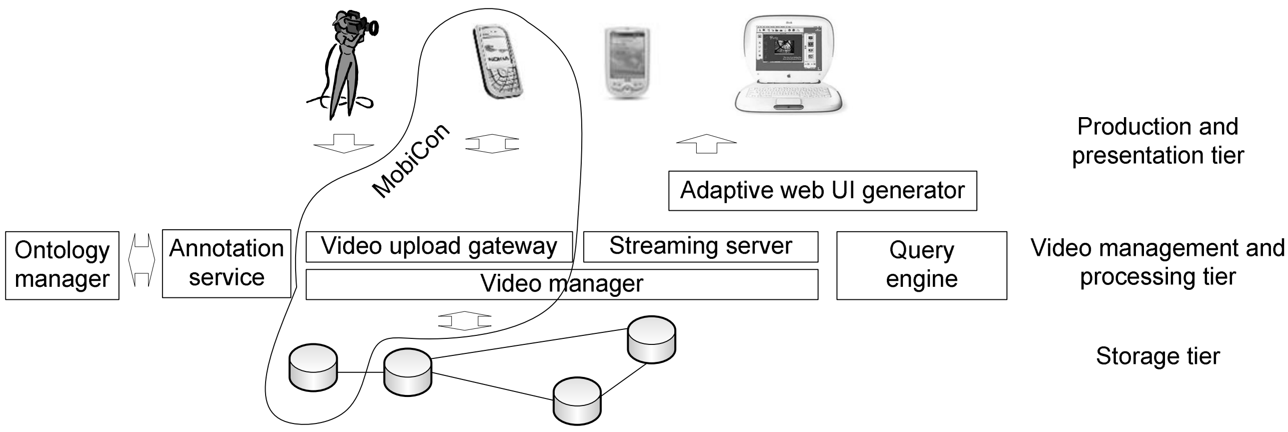

Figure 2: Candela and MobiCon system architecture.

4.1. Candela

Users can add video content to the Candela system produced by several different types of sources. The traditional method for home video content is to digitize analog content from a camcorder, or to use a digital one and upload the content to the system. Besides the home video domain, the Candela system can be a part of an enterprise-centered application such as, for example, an automatic security system handling the video content generated by a set of surveillance cameras. In the domain of personal mobile multimedia, the content is created in a mobile phone with video capture capabilities. In all application domains, the content is annotated automatically or semi-automatically, utilizing content analysis methods and input from sensors such as GPS positioning devices or context information which is available for example in a user's calendar. This information is captured in MPEG-7 metadata descriptions [7], which are stored along with the actual video data to support searching and managing video storage.

Candela uses the commercial Solid FlowEngine relational database system which provides scalability from a single in-memory database on a miniature embedded device to complex distributed and duplicated fault tolerant settings, which allows us to deploy more advanced configurations in the future. When storing, MPEG-7 descriptions are mapped to a relational schema so that SQL can be used for querying. In order to provide better results, the Candela query engine broadens user queries by suggesting additional search terms, which are, based on a domain ontology, closely related to the ones specified by the user or more descriptive. On the other hand, user profiles are used to restrict the amount of found matches to those that are the most relevant to the user.

Candela supports a very broad set of end user terminals ranging from cellular phones to high-end desktop computers. In order to provide a user friendly experience, the web-based interface dialogs are generated dynamically by using open source Apache Cocoon framework for XML transformations. This allows us to customize the amount of information presented to the user and the dialog between the user and the system to the capabilities of the used access device. Once the user has found an interesting video, a streaming video player is launched. This is either one of the off-the-self video players or Candela video browser which shows visualized metadata in addition to the video itself. The visualization allows the user to navigate through the video and find the relevant parts of it easily [5].

Providing optimal video quality for the end-user given the diversity of the source material, differences in user terminal capabilities and characteristics of networks, especially in the mobile domain, is a challenging task. MPEG-4 standard offers some scalable video coding solutions, where the changing network capabilities for video delivery can be taken into account in real-time by inserting enhancement layers to the video stream in case of more available bandwidth. However, it was concluded in the project that the state-of-the art in scalable video coding does not offer reasonable quality as compared to the non-scalable stream at the same bitrate [8].

If the media source is a mobile phone with a low resolution camera, the need for transcoding is not as obvious as for more bandwidth consuming content, but overall we want media to be accessible across platforms and at the moment there are no uniformly supported video formats. Thus the solution was, at the expense of storage, to transcode the material to a representative set of formats and bitrates and develop a content negotiation plug-in for Helix streaming server in order to choose from those.

4.2. MobiCon

The MobiCon client-server architecture is shown in Figure 3. MobiCon consists of two different software components: the UploadClient, which is a mobile Java (J2ME) application running on a mobile phone and UploadGateway, which is implemented as a Java servlet in the Candela server. Architecture is based on the need to provide wireless access over a mobile phone network to enable storing video clips on the server where it is also possible to run more computation-intensive operations such as video transcoding.

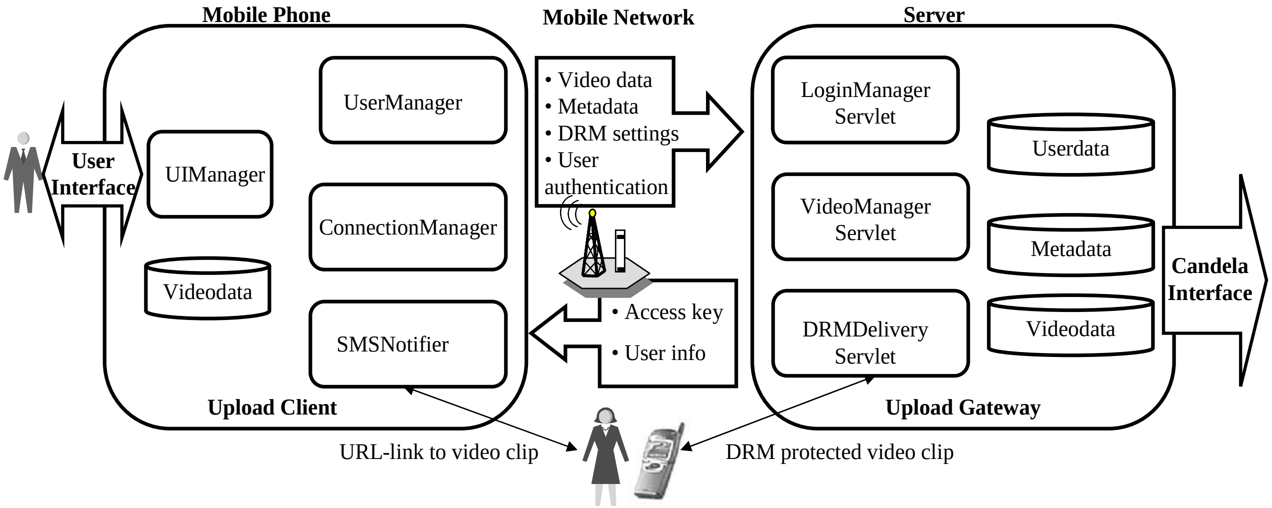

Figure 3: High-level description of MobiCon.

MobiCon naturally needs to be easily installed without any extra tools or additional instructions. The server allows distribution of MobiCon application easily to mobile phone users by using Over-The-Air (OTA) specification from the Open Mobile Alliance, which enables mobile applications to be downloaded and installed over the cellular network. After installation, the user is authenticated by the server using a username and password to log on the Candela system. The username and password are transferred to the UploadGateway and as a reply to successful authentication user profile information is transferred back to the UploadClient where UserManager stores user information (name, address, etc.), which are also used as metadata of captured video clips. The UploadClient no longer asks for username and password after the first time: for user convenience, it is assumed that the user stays the same after the first. Mobile phones are personal devices which are rarely lent to other people or left unguarded, the decision was made that the risk is not too large compared to the benefits achieved.

The UIManager is a controller component which is loaded first when the application is started. The UIManager coordinates the video capture using the mobile phone's camera, the saving of the video data to the Java Record Store system, and the sending of video sharing SMS messages to the other users. UIManager also provides user interfaces that are presented in the next Section. The ConnectionManager handles the connection between the UploadClient and UploadGateway providing data transfer using HTTP-protocol over the packet networks such as GPRS/EDGE/WCDMA. ConnectionManager delivers the captured video data, its metadata, user name, and DRM options to the Upload Gateway. If the user had chosen DRM protection for content sharing, the UploadGateway reply contains the identification keys to

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.