

Chapter 2.9

Enabling Multimedia Applications in Memory–Limited Mobile Devices

Raul Fernandes Herbster

Federal University of Campina Grande, Brazil

Hygo Almeida

Federal University of Campina Grande, Brazil

Angelo Perkusich

Federal University of Campina Grande, Brazil

Marcos Morais

Federal University of Campina Grande, Brazil

INTRODUCTION

Embedded systems have several constraints which make the development of applications for such platforms a difficult task: memory, cost, power consumption, user interface, and much more. These characteristics restrict the variety of applications that can be developed for embedded systems. For example, storing and playing large videos with good resolution in a limited memory and processing power mobile device is not viable.

Usually, a client-server application is developed to share tasks: clients show results while servers process data. In such a context, another hard task for limited memory/processing devices could be delegated to the server: storage of large data. If the client needs data, it can be sent piece by piece from the server to the client.

In this article we propose a layered architecture that makes possible the visualization of large videos, and even other multimedia documents, in memory/processing limited devices. Storage of videos is performed at the server side, and the client plays the video without worrying about

storage space in the device. Data available in the server is divided into small pieces of readable data for mobile devices, generally JPEG files. For example, when the client requests videos from the server, the videos are sent as JPEG files and shown at an ideal rate for users. The video frames are sent through a wireless connection.

The remainder of this article is organized as follows. We begin by describing background concepts on embedded systems and client-server applications, and then present our solution to enable multimedia applications in memory-limited mobile devices. We next discuss some future trends in mobile multimedia systems, and finally, present concluding remarks.

BACKGROUND

Embedded Systems

An embedded system is not intended to be a general-purpose computer. It is a device designed to perform specific tasks, including a programmable computer. A considerable number of products use embedded systems in their design: automobiles, personal digital assistants, and even household appliances (Wayne, 2005). These limited systems have some constraints that must be carefully analyzed while designing the applications for them: size, time constraints, power consumption, memory usage and disposal, and much more (Yaghmour, 2003).

These constraints restrict the variety of software for embedded systems. The development of applications which demand a large amount of memory, for example, is not viable for embedded systems, because the memory of such devices is limited. Extra memory can also be provided, but the total cost of application is very high. Another example is multimedia applications, such as video players: storing and playing large videos with good resolution in a limited memory and processing power mobile device is a very hard task.

There are specific platforms that were developed to perform multimedia tasks: embedded video decoders and embedded digital cameras, for example. However, other considerable parts of embedded systems, like personal digital assistants (PDAs) and cell phones, are not designed to play videos with good quality, store large amount of data, and encode/decode videos. Thus, it is important to design solutions enabling multimedia environments in this variety of memory/processing-limited devices.

Layered and Client-Server Architectures

Layered architectures share services through a hierarchical organization: each layer provides specific services to the layers above it and also acts as a client to the layer below (Shaw & Garlan, 1996). This characteristic increases the level of abstraction, allowing the partition of complex problems into a set of tasks easier to perform. Layered architectures also decouple modules of the software, so reuse is also more easily supported. As communication of layers is made through contracts specified as interfaces, the implementation of each module can be modified interchangeably (Bass & Kazman, 1998).

Most of the applications have three major layers with different functionalities: presentation, which handles inputs from devices and outputs to screen display; application or business logic, which has the main functionalities of the application; and data, which provides services for storing the data of the application (Fastie, 1999).

The client-server architecture has two elements that establish communication with each other: the front-end or client portion, which makes a service request to another program, called server; and the back-end or server portion, which provides service to the request. The client-server architecture allows an efficient way to interconnect programs that are distributed at different places (Jorwekar, 2005). However, the client-server architecture

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/enabling-multimedia-applications-memory-limited/27100

Related Content

Multimedia Social Network Modeling using Hypergraphs

Giancarlo Sperli, Flora Amato, Vincenzo Moscato and Antonio Picariello (2016). *International Journal of Multimedia Data Engineering and Management* (pp. 53-77).

www.irma-international.org/article/multimedia-social-network-modeling-using-hypergraphs/158111

Contiguous Placement on Hierarchical Storage Systems

Phillip K.C. Tse (2008). *Multimedia Information Storage and Retrieval: Techniques and Technologies* (pp. 156-160).

www.irma-international.org/chapter/contiguous-placement-hierarchical-storage-systems/27010

A Texture Preserving Image Interpolation Algorithm Based on Rational Function

Hongwei Du, Yunfeng Zhang, Fangxun Bao, Ping Wang and Caiming Zhang (2018). *International Journal of Multimedia Data Engineering and Management* (pp. 36-56).

www.irma-international.org/article/a-texture-preserving-image-interpolation-algorithm-based-on-rational-function/201915

Design Principles for Crisis Information Management Systems: From Closed Local Systems to the Web and Beyond

Cynthia Marie Nikolai, Troy Johnson, Michael Prietula, Irma Becerra-Fernandez and Gregory R. Madey (2018). *Digital Multimedia: Concepts, Methodologies, Tools, and Applications* (pp. 420-439).

www.irma-international.org/chapter/design-principles-for-crisis-information-management-systems/189485

An Innovative Multiple-Object Image Retrieval Framework Using Hierarchical Region Tree

Wei-Bang Chen and Chengcui Zhang (2013). *International Journal of Multimedia Data Engineering and Management* (pp. 1-23).

www.irma-international.org/article/an-innovative-multiple-object-image-retrieval-framework-using-hierarchical-region-tree/95205