WP9 - Validation on Platform

Deliverable D9mm.2 - Multimedia Application Component Specification

WP9 - Validation on Platform: Deliverable D9mm.2 - Multimedia Application Component Specification
by Adrian Matellanes

Table of Contents

Chapter 1. Introduction

1.1. Document Purpose

In this document we will identify the OCERA components that will be used in the development of the multimedia application.

This is not an architecture document, nor a requirements document, which are present in their corresponding deliverables. Anyhow, to make the reading easier we have added a brief application description.

After the application description, we will present the OCERA components to be used by the application and which of the services and features they provide will be exploited.

Chapter 2. Application Description

In this chapter, we will present a brief description of the application for document completeness. Further information on requirements and concrete architecture are described in other documents.

2.1. Overview

Visual Tools will develop a digital video viewer for StrongArm based handheld platforms that will be integrated with its range of Digital Video Recorders and Transmitters (DVRT) using Wireless Communications. We will call this application Visual Tools Wireless Viewer, VTWV from now on. It is an application oriented to videosurveillance.

As already stated in previous documentation, this application is both, high resource consuming and running on a low performance platform (StrongARM) and thus the Resource Reservation components will be extensely used.

In what respects communications, the nature of the transmission media (radio) enhance the necessity of controlling resources since, depending on particular conditions, the scenario is prone to interferences and can increase the resource consumption when transmiting the video stream.

We will develop an lightweight X Window application to receive wirelessly and present video streams transmitted by one (or several) Visual Tools DVRT. We will be able to find out which DVRT are present in the LAN and request them to stream video from any of its cameras.

Due to commercial reasons, though the main development will, of course, be done on a Linux platform using the OCERA components, we will develop the application in a programming language suitable for an easy port to PocketPC. This porting will lack the benefits of the OCERA components wich are developed in and for a Linux environment.

2.2. Architecture

The development of VTWV will be done on a Compaq iPAQ 3850 with a Symbol 802.11b CompactFlash Wireless LAN card.

2.2.1. Linux Distribution

The target handheld will have the Linux Familiar distribution installed (see http://familiar.handhelds.org)

Instead of using the off-the-shelf kernel in the Familiar distribution, we will use the kernel from the OCERA project, which is a vanilla kernel plus several patches that enhance its soft real-time capabilities.

The Familiar distribution will provide all external software required for the application. In particular, it will provide the different drivers needed for the CF Wireless LAN card, the iPAQ screen, etc... as well as specific software developed for this kind of platform, as the handwriting recognition software, orientation display manager, etc...

2.2.2. Ewe

For the development of VTWV we will use the Ewe programming system (http://www.ewesoft.com).

Ewe allows to write Java programs for desktop computers as well as for mobile handhelds. It is particularly suitable for the development on handhelds platforms and will allow us not only to use it under the Linux kernel for the OCERA project but to port the application (though, unfortunately, making no use of the OCERA components) to the PocketPC OS, which is of particular importance from the commercial point of view.

VTWV have no hard real-time constraints, but in case we add, in future releases, some feature requiring hard real-time capabilities, we can still make use of Ewe since it provides an API for C extensions.

Chapter 3. Components Specification

VTWV, as a multimedia application, will benefit from the OCERA resource reservation components. Next we will present to what extent VTWV will make use of each OCERA resource reservation component.

3.1. Generic Scheduler Patch

This component consist of a small patch that will be applied to the Linux kernel.

This patch provides hooks to the Linux kernel that are to be used by the resource reservation scheduling module, which implements sophisticated real-time scheduling policies that able the CPU reservation.

3.2. Integration Patch

The Integration patch take into account the introduction of the preemtion patch and the high-resolution timers patch.

The inclusion of this patch into the kernel running on the iPAQ, over which VTWV will run, is subject to the resolution of eventual incompatibilities of this patch with the StrongArm architecture.

3.3. Resource Reservation Scheduling Module

This OCERA component consist of a kernel loadable module. This module will provide a **resource reservation scheduler** for soft real-time tasks in user space. It is based on the Constant Bandwidth Server.

This is the core of the resource reservation components and is the most important component VTWV will test. This component will be responsible for the soft real-time character of VTWV. As we have exposed several times, the high cpu-consuming characteristics of the multimedia application together and the wireless transmission media (that can heavily affect performance because of retransmissions caused by interferences) conforms an ideal scenario for the verification and testing of the resource reservation scheduler. Furthermore, VTWV being run in a limited-cpu environment increases the need for resource control to get a smooth behaviour of the video player.

3.4. Quality of Service Manager

The Quality of Service Manager implements a mechanism for identifying the temporal characteristics of a task and to adjust its scheduling parameters so to maximize its quality of service.

This manager will allow us to set a specific quality of service for our application. Even if VTWV characterizes as a soft real-time application and no strict time constraints are needed, we can decide that under certain overload conditions, the cpu that the QoS manager can provide is not enough to give the appropriate performance and/or behaviour and we can inform the user so. This management will be done using the QoS Manager component.

This component is a kernel module plus a daemon which are fed with the appropriate requirement level: hard, soft or best-effort. The module measures the QoS and implements the control algorithm; the daemon will enforce the access policies and will receive requests from processes serving as interface for the module.

3.5. User Library/API

This component is used as a mean to access the resource reservation capabilities of the system. It is implemented as a set of user libraries that offers a POSIX (or similar) API convenient for the access to the resource reservation services. VTWV will use these libraries to setup the resource reservation services according to its needs.

Chapter 4. Other uses of OCERA

In this chapter we will enumerate other uses, apart from Components, of the OCERA output that will be used in the development of VTWV.

4.1. OCERA Linux Kernel

The development of VTWV will be based, naturally, on the OCERA Linux kernel.

We remind that this kernel is a standard Linux kernel with several patches applied in order to improve the soft real-time capabilities and to prepare the field for the development of some of the OCERA soft and hard real-time components.

4.2. OCERA Framework

The development of VTWV will be done using the OCERA framework. This framework permits the development and deployment of embedded and real-time systems based on the OCERA architecture.

Though the OCERA framework allow us to generate the embedded system root filesystem, we will make no use of this feature since we, luckily, have a complete Linux distribution on top of which our application will run.