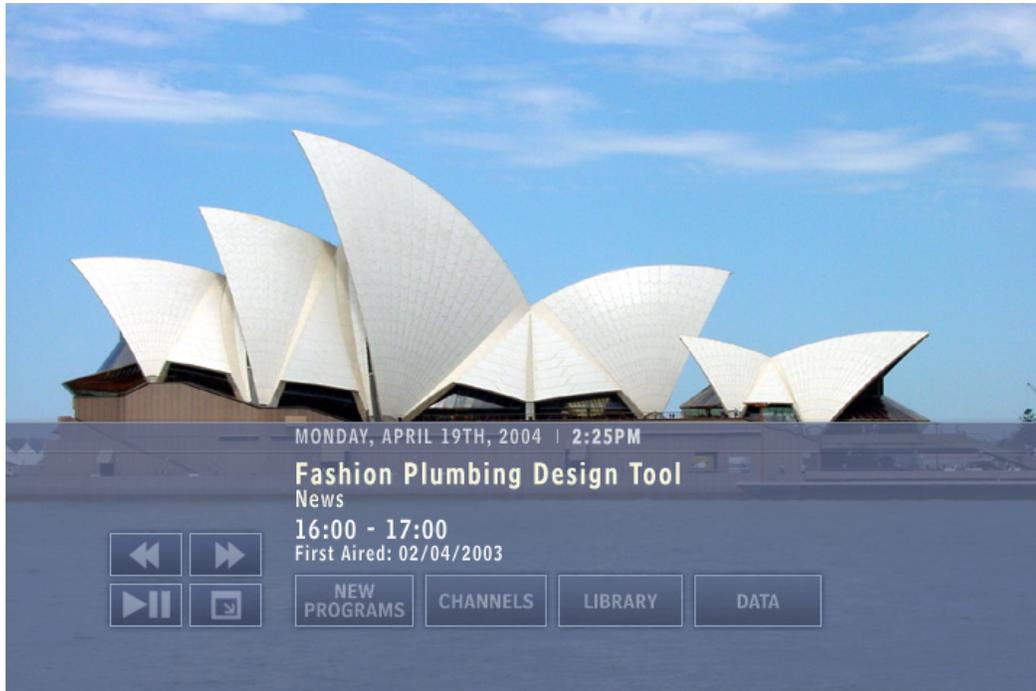


On Screen Display



Linux
Qt/C++
PostgreSQL

Problem: Our client had a store-and-forward satellite receiver with only a web interface for video play out control. Some of their customers did not have access to a web browser, and wanted to control the unit with TIVO like functionality on the monitor.

Solution: On Screen Display (OSD) Application using a remote control to allow operation of the unit like typical consumer video devices.

When the unit is first turned on, and the OSD option is enabled, a fullscreen display presents the user with options to view live channels, or items from a stored library of video. A “hot topics” priority listing of just the newest items (also in the library) is also presented. In addition to video selection, some configuration items are also available, including system status, IP address configuration, on-screen help, and a library search feature.

While video playback is in progress, the OSD is hidden, unless a button on the remote is pressed. In this case, a partial overlay is displayed, with a selectable opacity on the lower portion of the screen. Options presented on this overlay include typical VCR-like controls, FF, PAUSE, STOP, etc. as well as the option to return to the fullscreen menus. If no other buttons are pressed in a short period of time, the display hides itself again until video playback has ended.

DigitalGlue

A GUI was created that uses the video card overlay framebuffer so that the UI does not interfere with normal video playback. Visibility of the UI is controlled in software.

The OSD also controls the placement of a logo overlay (video bug) during playback, if desired. The position, size, and opacity of the overlay are configurable.

An RF remote was selected, using a USB receiver. This allowed the existing hardware to be used without modification (such as adding an IR port). Some remote control driver modifications were made to map custom keys on the remote control to be used for specific actions in the application.

The OSD application was designed to support a customized and flexible configuration.

By defining the visual look of the OSD in a configuration file, it is easy to rebrand (change logos), alter color schemes, and other graphical elements (buttons, backgrounds). This “skinning” feature was used by the OEM to offer professional services for those customers wishing a custom look.

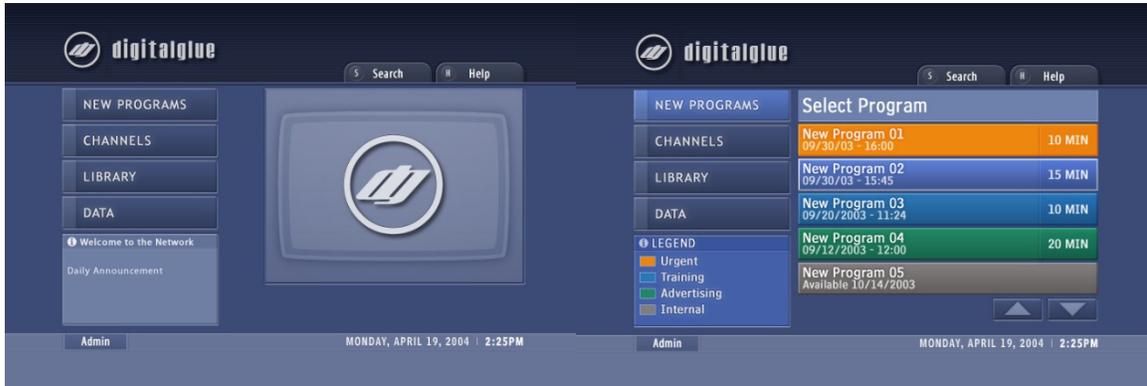
The OSD was implemented using the C++ Qt/Embedded framework. During development, an X-Windows emulation version was developed for ease of debugging, before the final embedded target application was delivered.

A PostgreSQL database was integrated with the platform. This was utilized for the OSD, to store metadata about the local content available, and other program guide information.

Where possible PostgreSQL stored procedures were used, rather than direct database manipulation in C++ code.

DigitalGlue

Sample Screens



Main Screen

New Programs



Program Information

Channel Information



Channel Guide

Search with Onscreen Keyboard