BitRouter's XML State Machine Technology

For Configurable Graphical User Interfaces

Abstract

Using BitRouter's **patent pending** XSM Technology, television and set top box designers rapidly prototype and deliver new functionality and graphical designs. The underlying software code base remains intact and does not require changes. As a result, design and testing is a much simpler process that can involve creative people who are not necessarily familiar with low level programming languages.

This paper discusses the XML State Machine (XSM) technology and its use for rapid user interface (UI) development. XSM uses generalized hierarchical state machines for creating UIs, where the UIs are defined by an XML configuration file.

TVgui is an implementation of the XSM technology for digital TV devices. This paper asserts the need for the TVgui program, the design approach taken, the XML syntax used to abstract the GUI, steps in creating the graphical assets and XML source code for the GUI, the generalized state machine implementation, and trade-offs of the approach.

Abbreviations

DTD	Data Type Definition
EMS	Electronic Manufacturing Service
GUI	Graphical User Interface
HSM	Hierarchical State Machine
ODM	Original Design Manufacturer
OEM	Original Equipment Manufacturer
TVgui	A digital TV application based on XSM technology
UI	User Interface
UML	Unified Markup Language
UML	Unified Markup Language
XML	eXtended Markup Language
XSM	XML State Machine, BitRouter's patent pending technology

Introduction

Anyone who has been involved in developing user interfaces understands the complexities of that endeavor. First, a human interface development expert analyzes the UI requirements and conceptualizes a user-friendly look and feel. A graphic artist then refines the look of the GUI and creates its media assets. Finally, programmers design and implement the GUI code, integrating the media assets and implementing the feel (behavior) of the GUI. Each step is inherently error prone and time consuming.

In the consumer electronics industry today, design cycles are rapidly shrinking. CE device manufacturers and designers want turnkey solutions that can be customized, tested, manufactured, and delivered to market in a matter of weeks. Frequently, silicon companies and EMS providers deliver reference designs for which OEMs and ODMs build customer user interfaces. These user interfaces need rapid and frequent changes to differentiate products and add unique value to each model. The time required to design and implement the custom user interface is a key bottleneck in the "time to market."

BitRouter wanted a way significantly improve the process and eliminate this custom user interface bottleneck. Using BitRouter's XSM technology, OEMs and ODMs rapidly create and customize GUIs for digital TV and embedded devices, such as TV sets, converter boxes and set-top boxes. Using this technology, a GUI can be defined and configured using an XML file. An executable program parses and renders the defined GUI. This is generically called skinning. **In BitRouter's approach skinning also involves defining the behavior of the GUI, not just its looks.**

The resulting program, called an **XML State Machine (XSM)** engine, is best described as a generalized hierarchical state machine, configurable by an XML file, and targeted at creating UIs. The XSM engine source code is completely reusable. The only difference between different UIs for the same device is the XML code. Coding is simplified to skinning and developing the XML file defining the GUI.

This paper discusses a particular type of XSM engine meant for digital TV receivers, appropriately called **TVgui**. XSM engines can be developed for any device that requires a user interface, such as gas pumps, credit card scanners, mobile phones, medical devices, etc.

XML Syntax Definition

The XML syntax is defined in a **Data Type Definition** file (**DTD**). The DTD abstracts out the state machine and GUI concepts. See Listing 1 (DTD Syntax) for reference.

{

Deleted section, available in full version by contacting BitRouter

}

Steps in Creating a GUI

To create a GUI with this technology, the recommended steps, are, in order:

{

Deleted section, available in full version by contacting BitRouter

}

Listing 2 is an example of a relatively simple skin. The TopState defines the fonts to be used in the skin and its Reaction elements define the overall behavior of the skin. TopState contains the three SubStates mentioned previously: Watching TV, Main Menu and Standby.

{

Deleted section, available in full version by contacting BitRouter

}

Figure 1 shows the main menu when it is first entered. Figure 2 shows a channel scan in progress. When TVgui is started for the very first time, there is no channel map data and the skin automatically starts a channel scan to obtain the data. Figure 3 shows how the GUI looks when the main menu is entered and the UP or DOWN key is pressed, so that the Signal Meter ViewPort is shown.

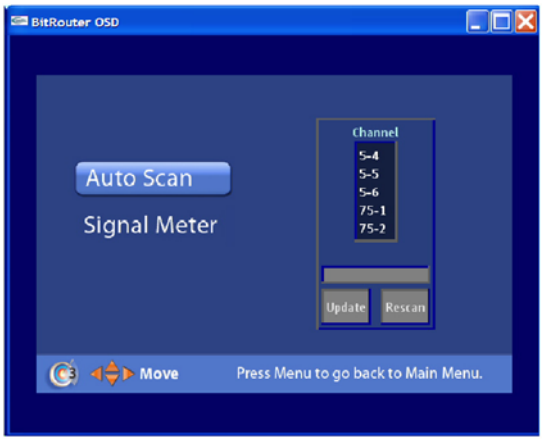

Figure 1 – Main Menu

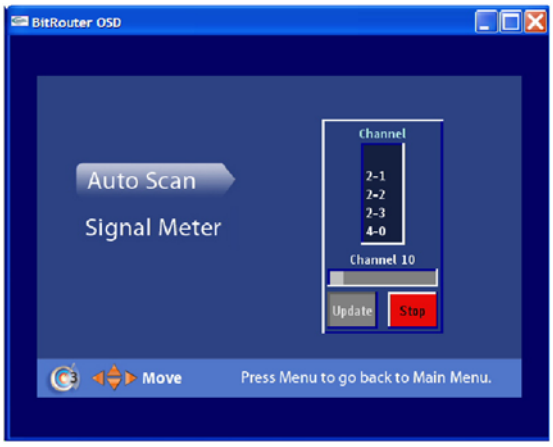

Figure 2 – Channel Scan

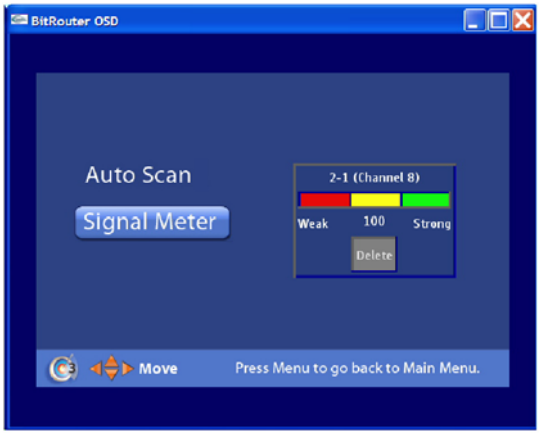

Figure 3 – Signal Meter

Generalized State Machine Implementation

This section discusses TVgui's internal data structures….

{

Deleted section, available in full version by contacting BitRouter

}

ViewPort Implementation and Other Considerations

As previously discussed, ViewPorts are GUI widgets that control and display information from the middleware. In essence, they are predefined state machines…

{

Deleted section, available in full version by contacting BitRouter

}

BitRouter's graphics library supports 8, 16, and 32-bit bitmaps. Its media library supports JPEG, PNG, GIF, TIFF, and run-length encoded bitmap files. Using 8 or 16-bit bitmaps or compressed images helps in reducing the footprint needed for a skin.

Using the same basic TVgui a designer can implement multiple DTV devices, such as an ATSC converter box, an integrated TV, a connected home TV, etc. Once all the underlying hierarchical state machine infrastructure is in place within TVgui, its use for different devices is achieved by manipulating the XML skin. Optionally, the underlying state machines (ViewPorts) that are not required for a particular device may also be compiled out, rather than disabled using XML.

BitRouter has implemented a commercial ATSC converter box using the TVgui application. The complete skin for this product is defined by only 2,200 lines of XML code. This is a key strength of the XSM technology. OEMs and ODMs need only manipulate 2,200 lines of XML to personalize the product for market introduction. A person-year of GUI development and testing has been reduced to a week's worth of XML editing. And, no C/C++ level source needs to be released to the OEM or ODM.

Conclusion

The CE industry faces shrinking design cycles. BitRouter's **patent pending** XSM technology offers an order of magnitude improvement in time to market by providing the following benefits:

- Pre-packaged functionality and UIs
- UI customization by manipulating minimal lines of XML code
- An upward migration of designer skills from C/C++ programming to XML or GUI design tools.
- IP protection by not requiring release of source code or API libraries
- Rapid UI prototyping for testing, demos, and products
- Reusable, generalized hierarchical state machine building blocks
- Devices with multiple user selectable and downloadable skins
- UI authoring and editing using third-party and BitRouter's GUI editing tools

Listing 1 – DTD Syntax

{

Deleted section, source code listing provided in full version, available by contacting BitRouter

}

Listing 2 – Example Skin

{

Deleted section, source code listing provided in full version, available by contacting BitRouter

}

