2010 2nd International Conference on Information and Multimedia Technology (ICIMT 2010)
IPCSIT vol. 42 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V42.24

The Design and Implement of Fleet F77 Station's Simulator

Xu JinJun

Naval Aeronautical Engineering Institute Qingdao Branch, Qingdao,Shandong,China,266041
Jinjun20002002@yahoo.com.cn

Abstract- In this paper, we introduced the main technology and method to develop marine F77 station's simulator with Flash software,Flash Media Server(FMS) and XML(eXtensible Markup Language) technology.Through this paper ,we can find that Flash,FMS and XML technology are ideally matched and is effective in developing marine F77 station's simulator.

Keywords: Flash,FMS,XML,GMDSS Simulator,F77 Station

1. Introduction

Along with the GMDSS(Global Maritime Distress and Safety System) entered into fully force on 1 February 1999,the GMDSS teaching and training based on simulator is becoming more and more important and simulator's developing are hotspots throughout the world ,because of the high training cost of GMDSS and equipment's false alert[1].

In this paper,we introduce the main technology used in developing Fleet F77 station's simulator.

Figure 1. The Applications of Fleet 77

2. The main applications of Marine F77 station

Because Fleet F77 station is IP compatible, it supports an extensive range of commercially available off the shelf software, as well as specialized user applications. This makes it an ideal solution for: e-mail,instant messaging,data file transfer, including FTP and digital images,online electronic chart updates,real time weather information,videoconferencing,store-and-forward video,telemedicine,high quality digital voice,crew calling,e-learning and so on,as shown in Figure 1[2][3].

Fleet F77 is a single, integrated solution that delivers versatility and choice through voice, fax and Mobile ISDN, Mobile Packet Data Service (MPDS) communications.

3. Flash Software, ActionScript and XML

3.1 Flash Software

Adobe Flash software is the industry standard for interactive authoring and delivery of immersive experiences that present consistently across personal computers, mobile devices, and screens of virtually any size and resolution.

Flash provides everything you need to create and deliver rich web content and powerful applications. Whether you’re designing motion graphics or building data-driven applications, Flash has the tools to produce great results and deliver the best user experience across multiple platforms and devices.

We create Fleet F77 station simulator’s interface with Flash , which runs in Flash Player,as shown Figure 2. The F77 simulator provides the application’s user interface, fulfill voice and video operation steps of F77 station and also contains ActionScript for connecting to, and managing interactions with, Flash Media Server.

Figure 2. Fleet F77 Station Simulator Developed With Flash

Flash is an authoring tool that lets designers and developers create presentations, applications, and other content that enables user interaction.Flash projects can include simple animations, video content, complex presentations, applications , and everything in between. In general, individual pieces of content made with Flash are called applications, even though they might only be a basic animation. You can make media-rich Flash applications by including pictures, sound, video, and special effects. Figure 3 is Flash integrated development environment.

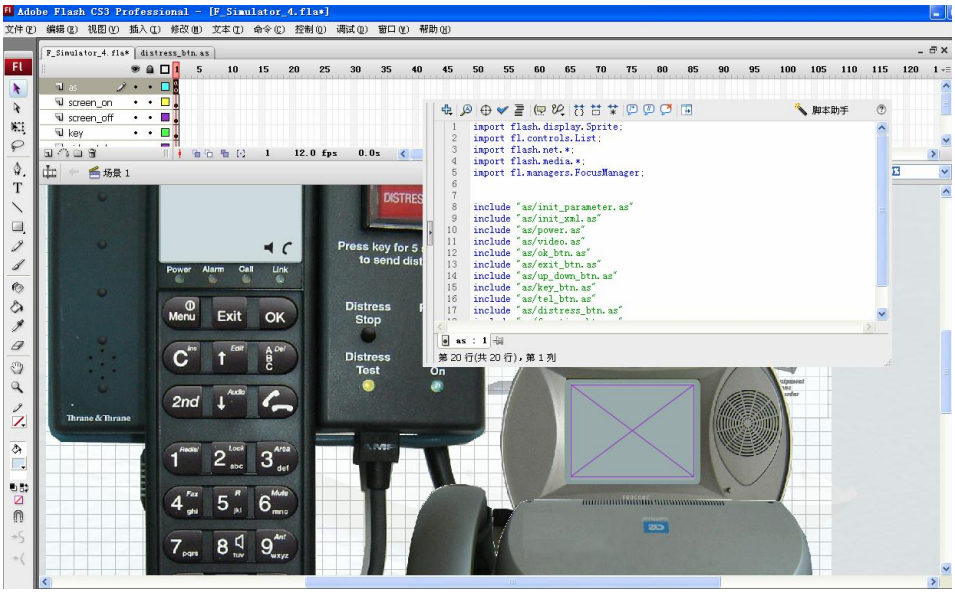

Figure 3. Flash Integrated Development Environment

Flash includes many features that make it powerful but easy to use, such as prebuilt drag-and-drop user interface components, built-in behaviors that let you easily add ActionScript to your document, and special effects that you can add to media objects.

To build an application in Flash, you create graphics with the Flash drawing tools and import additional media elements into your Flash document. Next, you define how and when you want to use each of those

elements to create the application you have in mind. Figure 4 are the pictures before and after click the Distress button.

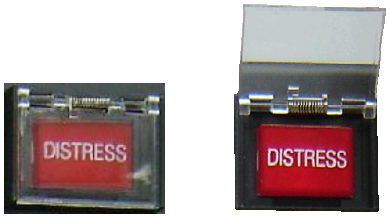

Figure 4. Distress Button Component

3.2 ActionScript

ActionScript code is the programming code you can add to Flash documents to make them respond to user interactions and to more finely control the behavior of your Flash documents. For example, you can add code that causes a button to display a new image when the user clicks it. You can also use ActionScript to add logic to your applications. Logic enables your application to behave in different ways depending on the user’s actions or other conditions. Flash includes two versions of ActionScript, each suited to an author’s specific needs.

ActionScript and JavaScript are both rooted in the ECMA-262 standard, the international standard for the ECMAScript scripting language. For this reason, developers who are familiar with JavaScript should find ActionScript immediately familiar.

3.3 XML Technology

XML(eXtensible Markup Language) is a standard way of representing structured information so that it is easy for computers to work with and reasonably easy for porgrammer to write and understand.

XML offers a standard and convenient way to categorize data, to make it easier to read, access, and manipulate. XML uses a tree structure and tag structure that is similar to HTML. Here is a simple example of XML data.

```
<song>
    <title>What you know?</title>
    <artist>Steve and the flubberblubs</artist>
    <year>1989</year>
    <lastplayed>2006-10-17-08:31</lastplayed>
</song>
```

XML data can also be more complex, with tags nested in other tags as well as attributes and other structural components.

Notice that this XML document contains other complete XML structures within it (such as the song tags with their children). It also demonstrates other XML structures such as attributes (tracknumber and length in the song tags), and tags that contain other tags rather than containing data (such as the tracks tag).

ActionScript 3.0 includes several classes that are used for working with XML-structured information. The two main classes are as follows:

XML: Represents a single XML element, which can be an XML document with multiple children or a single-value element within a document.

XMLList: Represents a set of XML elements. An XMLList object is used when there are multiple XML elements that are "siblings" (at the same level, and contained by the same parent, in the XML document's hierarchy). For instance, an XMLList instance would be the easiest way to work with this set of XML elements (presumably contained in an XML document):

In addition to the built-in classes for working with XML, ActionScript 3.0 also includes several operators that provide specific functionality for accessing and manipulating XML data. This approach to working with

XML using these classes and operators is known as ECMAScript for XML (E4X), as defined by the ECMA-357 edition 2 specification.

3.4 Common XML tasks

When you work with XML in ActionScript, you are likely to do the following tasks:

Constructing XML documents (adding elements and values)

Accessing XML elements, values, and attributes

Filtering (searching in) XML elements

Looping over a set of XML elements

Converting data between XML classes and the String class

Working with XML namespaces

Loading external XML files .

4. Flash Media Server

Flash Media Server(FMS) is a real-time media server that can deliver video on demand, live video, streaming music, video blogging, video messaging, multimedia chat environments, real-time datacasting, and multiuser gaming.

FMS is an open socket server. The key difference between open socket servers and Web servers is that as soon as you receive information from a Web server, the connection is broken. It may look as if you're still connected to the Web server, especially with a Flash page that's animating materials. However, that's not the way it works. If you open a Web page, the Web server sends you the page along with all associated graphics, text, and other media; and your computer sends a message back and the connection closes. With an open socket server, the connection stays open until you quit the application or trigger an event that cuts the connection. Because the connection remains open, you can stream audio, video, text, and any other media available on the Internet, in real time. You just can't do that with a regular Web server because it has an entirely different architecture.

Through FMS and Flash client, applications can send audio and video as well as text messages to the server, and the server streams the data to all connected users. The server can also record media for playback at a later time, such as in a video messaging application.

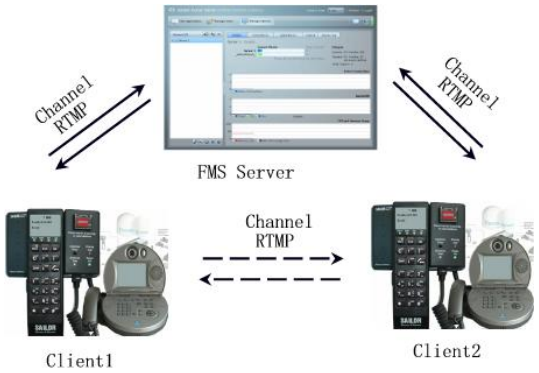

Figure 5. Flash Media Server provides communication channel for clients

The FMS and the Flash client application communicate over a persistent connection using Real-Time Message Protocol (RTMP). In a typical scenario, a web server delivers the Flash client to Flash Player over HTTP. The Flash client then uses RTMP to establish a persistent connection to Flash Media Server, allowing for an uninterrupted data stream to flow between client and server as shown in Figure 5[4]. The RTMP was designed for high-performance transmission of audio, video, and data between Adobe Flash Platform technologies.

Flash Media Server is a hub. Applications connect to the hub using Real-Time Messaging Protocol. The server can send data to and receive data from many connected users. FMS acts as a communication channel between connected users,shown in Figure 6. A user can capture live video or audio using a camera and

microphone attached to a computer running Flash Player and publish it to a server that streams it to thousands of users worldwide. Users worldwide can participate in an online game, with all moves synchronized for all users.

Users connect to the server through a network connection. A connection is similar to a large pipe and can carry many streams of data. Each stream travels in one direction and transports content between one client and the server. Each server can handle many connections concurrently,as shown in Fig4.

In traditional client-server applications, the server is typically used to execute some kind of transaction; the client makes a request, the server performs a database query or some resourcebased calculation, and then returns a result to the client. The connection between the client and server is maintained only long enough to complete the transaction.

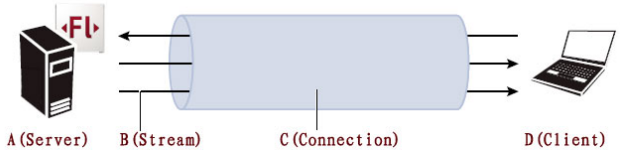

Figure 6. Connection and Stream

While you can use FMS to implement transactions, its core use is handling interactions—coordinating the actions of multiple, connected users, or clients, and transmitting server-side data. FMS provides two communication models that simplify the process of handling user interactions: streams and shared objects.

Streams are a time-based flow of synchronized audio, video, and/or data messages that flow from client to server, or from server to client. Streams use a publish and subscribe model that simplifies development of applications that use streams. A published stream can be played in real time or recorded and played later.

5. Developing and Simulating

To build a Flash application,you typical perform the following basic steps.

Add media elements:create and import media elements,such as images,video,sound,text.

Arrange the elements:arrange the media elements on Stage.

Use ActionScript to control behavior.

Write ActionScript code to control how the media elements behave,including how the elements respond to user interactions.

Following is part of the simulator’s ActionScript code which implement video in simulator.

To view video data, you must create a Video object and call the Video.attachNetStream() method; audio being streamed with the video, or an FLV file that contains only audio, is played automatically. To stream audio from a microphone, use the NetStream.attachAudio() method and control some aspects of the audio through the Microphone object.

The NetConnection class lets you invoke commands on a remote application server, such as Flash Media Server, and to play streaming Flash Video (FLV) files from either an HTTP address or a local drive. Typically, you use NetConnection objects with NetStream objects.

The NetStream class opens a one-way streaming connection between Flash Player and FMS, or between Flash Player and the local file system through a connection made available by a NetConnection object. A NetStream object is like a channel inside a NetConnection object; this channel can either publish audio and/or video data, using NetStream.publish(), or subscribe to a published stream and receive data, using NetStream.play(). You can publish or play live (real-time) data and previously recorded data. You can also use NetStream objects to send text messages to all subscribed clients.

The following codes is part of the F77 simulator’s code to implement voice and video communication.

```
//creates connection object
var video_nc:NetConnection=new NetConnection();
video_nc.connect("rtmp://127.0.0.1/F_Station")
```

```
//create transmit and receive stream on video_nc
var video_send_ns:NetStream=
new NetStream(video_nc);
var video_receive_ns:NetStream=
new NetStream(video_nc);
//get microphone and camera
var video_mic:Microphone=
Microphone.getMicrophone();
var video_cam:Camera=Camera.getCamera();
//attach microphone and camera object to video_send_ns
video_send_ns.attachAudio(video_mic);
video_send_ns.attachCamera(video_cam);
```

The above code creates the connection between clients and server,attaches microphone and camera to the sending stream.When the user make video phone call ,the voice and video will be sent through the specified channel,as the following code shows:

```
//sends live streaming audio and video to server
video_send_ns.publish("F_Video", "live");
```

The another user can hear and see the published streaming voice and video on specified channel through the following code:

```
// Specifies a video stream to be displayed within video_tel object in the application
var video_tel:Video=new Video();
video_tel.attachNetStream(video_receive_ns);
video_receive_ns.play("F_Video");
this.addChild(video_tel);
```

From the above code,we can see that voice and video communication and broadcast based on simulator can be achieved easily with FlashCS and FMS.This is the feature and advantage of developing F77 simulator with Flash and FMS.

Figure 7, Figure 8 are the displays of telephone call and video telephone separately.

6. Conclusion

The developing of GMDSS simulator and GMDSS training based on simulator is becoming more and more important in maritime college and plays an important role in navigation teaching today. With Fleet F77 station simulator we fulfill imitation of F77 operation, telephony and video communication in GMDSS.

The practical Fleet F77 training has proven that the training cost has been reduced a lot ,GMDSS false alert has been avoided and the training effects have been improved effectively.

Figure 7. Making Telephone Call

Fig.8:Video Communication

Flash has strong multimedia process function,powerful ActionScript and net programming ,with which we can develop GMDSS simulator with high imitation ,friendly interface and high system stability.

FMS is a powerful server platform for creating rich media applications in Macromedia Flash and for streaming audio and video to Flash clients. Implementing the communication in video ,telephony based on F77 simulator with Flash and FMS using Real-Time Messaging Protocal has not been reported by other papers(at least we do not find).

Through this paper,we can find that Flash Software , FMS and XML have the obvious advantages in developing Fleet F77 simulator over other programmming languages.

7. References

[1] "ADMIRALTY LIST OF RADIO SIGNALS VOLUME5" ,PUBLISHED BY THE UNITED KINGDOM HYDROGRAPHIC OFFICE , 2005/6,

[2] "Sailor Fleet55, 77User Manual", http://www.thrane.com

[3] "Inmarsat_F77_technical_information", http://www.inmarsat.com/Services/Maritime

[4] "Developing Media Applications", http://livedocs.macromedia.com/