Design and Realization of Chatting Tool Based on Web

Shi Yuzhuo[1], Hao Kun[2]
[1]Tianjin College of Commerce, Tianjin, China
[2] School of Computer and Information Engineering, Tianjin ChengJian University, China

Abstract—With the fast development of Internet, more and more people choose network chatting tools for communication. Traditional real-time chatting software is usually desktop application programs based on C/S mode, and specific client programs are needed during application. The browser-based real-time chatting tool does not need any additional client program, and the visual communication could be conveniently realized through browser. The real-time chatting tool realized through HTML5 and WebRTC technologies is described in this paper. The system includes basic information management module, chatting communication module, and space management module. The text communication is realized through server forwarding data, and the data transmission of voice and video chat is realized through point to point connection between browsers. Test results show that this system could be excellently applied to practical network environment, and the system is safe, efficient and easy to maintain and extend.

Keywords- al-time Communication; Chatting Tool; HTML5; WebRTC; WebSocket

I. INTRODUCTION

With the development of network, the communication between people is no longer limited to phone calls or short messages, and chat programs such as QQ and MSN are also frequently used. Not only could carries out real-time communications, these programs also could realize functions such as file transmission, games, music, which further accelerate information transmission and realize information share. Traditional real-time chatting software is usually desktop application programs based on C/S mode, and specific client programs are needed during application. But these specific client programs could not be applied to various operating system platforms, the process of server deployment is relatively complex, and both the client-side and server need to be updated during system update. Hence, the real-time chat system based on B/S mode appears.

HTML5[1][2] is the next generation of HTML, and it has some new tags. More robust reference could be developed by using it, the availability and user experience could be improved, and more multi-media elements (such as video and audio) could be provided for the site. Hence, it will be widely used in games and mobile applications.

Real-time audio and video transmission could be carried out through WebRTC [3][4][5]technology by establishing point to point connection between browsers. Web developer could easily realize browser-based real-time communication applications through simple JavaScript API provided by browser. Core technologies of WebRTC mainly include audio engine, video engine and network transmission three parts. Specifically, functions such as video and audio collection, encoding and decoding, display and network transmission are included. WebRTC could be applied to a variety of operating system platform and mobile platform. The ultimate goal of WebRTC technology is helping Web developer to quickly design colorful real-time application through simple JavaScript API based on browser without downloading or installing any kind of plug-in on client side.

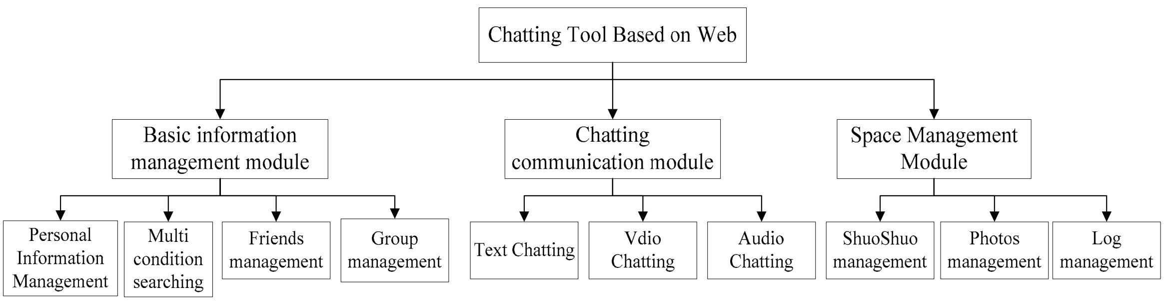

Figure 1. System structure chart

Real-time chatting system using HTML5 and WebRTC technologies based on B/S architecture model is presented in this paper. System includes basic information management module, chatting communication module and space management module. HTML5 and WebRTC should be supported by system client side, and server side should be Tomcat 7. Text chatting is realized through adopting WebSocket based on TCP protocol, and the audio and video chatting is realized through adopting WebRTC based on UDP. WebRTC provides very simple JavaScript API to developer through browser in order to effectively reduce the realization difficulty of audio and video chatting system. Point to point

978-1-4799-2860-6/13/$31.00 ©2013 IEEE

connection between browsers is established through RTCPeerConnection, video and audio collection, encoding and decoding, and network transmission are realized through adopting WebRTC, and the playing of audio and video is realized through adopting HTML5. Finally, text communication is realized through server forwarding data, and data transmission for audio and video chatting is realized through establishing point to point connection between browsers. System organizational chart is shown in Figure 1.

The paper is organized as follows. Section II describes the system realization. Section III presents the key technologies of this system. Section IV concludes this paper.

II. SYSTEM REALIZATION

System adopts browser/server structure. Server side is a PC and a PC with digital camera, headphone and microphone is needed on client side. The chatting contents are forwarded through browsers, and point to point connection is directly established between browsers during audio and video chatting. The network topology[6] is shown in Figure 2.

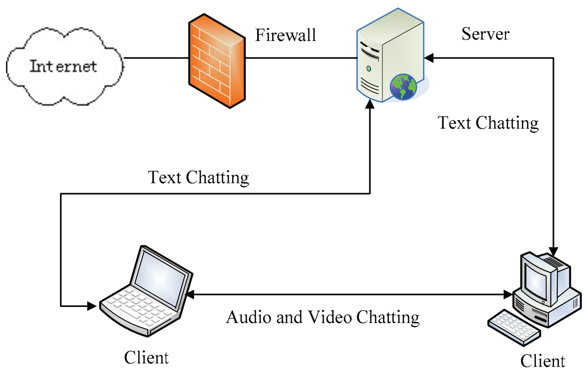

Figure 2. Network topology structure

A. Basic Information Module

Information required for registration is very simple, only including nickname and password, and more information could be gradually improved. After registering and logging in, user could searcher and add friends through multi-condition, and could further group or remark for them.

B. Chatting Communication Module

Text chatting, audio chatting and video chatting could be realized through HTML5+WebRTC technology. In chatting information module, Tomcat 7 is adopted on server side for realizing WebSocket. Server side monitors client side connection and process client side information.

1) Text chatting is realized through WebSocket technology based on HTML5. The flow chart of text chatting is shown in Figure 3.

The process of communication between browser and server through WebSocket is shown as follow:

(a) A monitor activated by server side is needed for monitoring connection requirement sent by browser. Different from the traditional web application based on Socket, server used for realizing WebSocket needs to analyze the WebSocket handshake information sent by browser, and generates corresponding answer information in accordance with WebSocket specifications, which has been realized by WebSocketServlet in Tomcat 7.

(b) The connection with server is established through calling constructor of WebSocket.

(c) After receiving connection from client side, server side creates MessageInbound in order to serving for this client side.

(d) If the above-mentioned steps are completed, WebSocket is on OPEN state, and the messages could be sent to server by calling the send function of WebSocket.

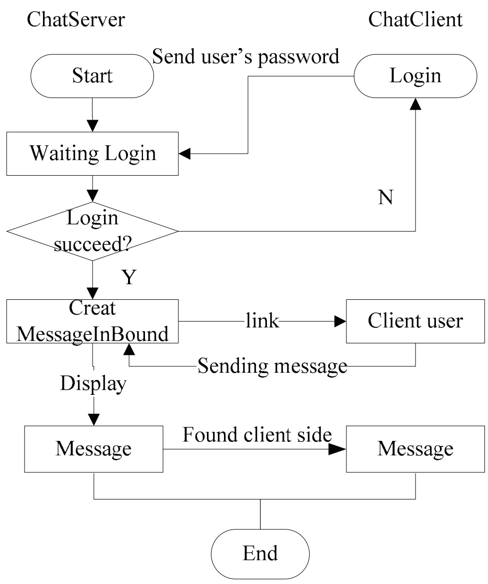

Figure 3. The flow chart of text chatting

2) Audio and Video Chatting

The collection, sending and display of AV (audio/video) on client side is realized through adopting WebRTC technology, which could traverse NAT realizing the interconnection between browsers. The flow chart of audio and video chatting is shown in Figure 4.

Using the programming interface provided by WebRTC, the transmission of AV could be realized through the following steps.

(a) Camera and microphone could be activated through calling webkitGetUserMedia interface, and binary data stream of AV could be further acquired.

(b) The URL pointing to binary data stream acquired in the previous step could be obtained through calling webkitURL.createObjectURL function.

(c) The obtained AV data could be played through setting properties of video tag provided by HTML5.

(d) IceCandidate could be acquired through establishing RTCPeerConnection.

(e) Set callback function for onicecandidate of RTCPeerConnection, and send the IceCandidate message obtained in step 4 to browser on opposite side through server.

(f) Active connection side calls createOffer function of RTCPeerConnection.

(g)Passive connection side calls createAnser function of RTCPeerConnection.

(h) Both sides call addStream function of RTCPeerConnection respectively, and AV data begins to transmit.

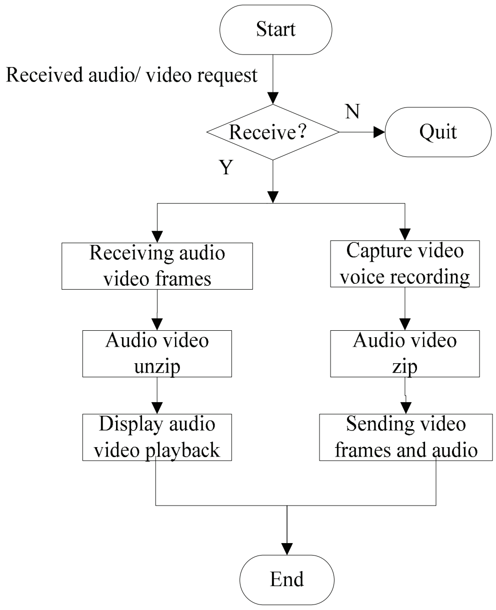

Figure 4. The flow chart of audio and video chatting

C. *Personal Space Management Module*

The dynamic management of user's personal space could be realized by this module, including management for journals, ShuoShuos and photos. During photos uploading, several types of photos are supportable, and several photos could be uploaded at once. Photos could be played in form of slideshow photo browsing.

III. KEY TECHNOLOGIES

A. *WebSocket Communication*

The connection between browser and server is established through WebSocket, and different treatments are carried out by socket.onmessage according to different message types after WebSocket receiving message sent by server. The message might be text message sent by friend, might be friend's AV chatting requirement, and also might be response to your requirement, and its core codes are shown as follow.

```
socket.onmessage = function(event) {
if (typeof event.data === "string") {console.log("Receive"+event.data);
    try { var json = JSON.parse(event.data); if (json.type === "roomLink") {
    var index = json.content.indexOf("room");
    if (index !== -1) {
            json.content = json.content.substring(index);}
  console.log("Receive Reslove roomLink" + json.content);
  $("#" + json.fromqq + "Right iframe").attr("src", json.content);}
   else if (json.type === "doAskOnline") {
  var logContent = "<label style='color:red'>" + json.content
+ "</label>";log(logContent,json.qq);}
  else {
  var logContent = "<label style='color:red'>" + json.fromqq+ " say:" + json.content + "</label>";
```

```
log(logContent, json.fromqq);}}
catch (err) {console.log("Resolve error " + event.data);}}
else if (typeof event.data === "object") {
var target = document.getElementById("target");
varurl= webkitURL.createObjectURL(arrayBuffertoBlob(event.data));
target.onload = function() {
window.webkitURL.revokeObjectURL(url);};
target.src = url;}};
```

B. Point to Point Connection Established Through RTCPeerConnection

Codes for connection establishment, loading AV data, web transmission and display through RTCPeerConnection are very complex, and codes for establishing createPeerConnection is shown.

```
function createPeerConnection() {var pcConfig=null; try {pc = new RTCPeerConnection(pcConfig);pc.onicecandidate = onIceCandidate;
console.log("PASS"+JSON.stringify(pc_config) + "Built RTCPeerConnection ");}
catch (e) {console.log("Establish Fail"+e.message);
try {var stun_server = "";if (pc_config.iceServers.length!==0)
{stun_server = pc_config.iceServers[0].url.replace('stun:', 'STUN ');}
pc = new webkitPeerConnection00(stun_server,onIceCandidate00);
isRTCPeerConnection = false;
console.log("Configure"+stun_server+"Build webkitPeerConnnection00");} catch (e) {console.log("Build Fail"+ e.message);
    return;
}}
```

IV. CONCLUSION

Users could conveniently carry out visual communication by means of browsers through chatting system designed by using HTML5 and WebRTC technologies. Basic information management module, chatting communication module and personal space management module could be realized. Test results show that this chatting system could be applied to various network environments, providing chatting service for different types of people, and the system is safe, efficient and easy to maintain and extend.

V. ACKNOWLEDGMENT

This research was supported by The National Natural Science Fund Project: The research of underwater sensor network dependability safeguard mechanisms and its application of Qinghai Lake, Sanjiang river basin monitoring. (No. 61162003). The Higher Education Science and Technology Planning Project of TianJin (20130419).

REFERENCES

[1] Rob Hawkes, Foundation HTML5 Canvas: For Games and Entertainment[M], friends of ED, 2011

[2] Mark Pilgrim, HTML5: Up and Running: Dive into the Future of Web Development[M], O'Reilly,2010

[3] Chen Tieming, Chen Huibing etc. Designing and Implementing New Dynamic Web Password Login Method, Computer Applications and Software, 2011,28(7):31-34

[4] http://struts.apache.org/

[5] Walter J. Savitch, Java:an introduction to problem solving & programming[M],Beijing University of Post and Telecommunications Press, 2006

[6] W.Richard Stevens, TCP/IP ILLUSTRATED Volume 3:TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols[M], China Machine Press, 2011