Design and Implementation of Instant Communication Systems Based on Ajax

Liu Tian and Wu Jun

Faculty of Information Engineering
Jiangxi University of Science and Technology, Ganzhou 341000, China

Abstract. The instant communication tool is a hot topic in today's web applications. The method of interaction for polling refreshing every a period of time not only waste network traffics, but also led to the issue of white screen. To address these issues, in the paper, we take the chat system of college friend website as an example to introduce the design of instant communication systems based on Ajax in details.

Keywords: Chat, Ajax, Web, B/S construction.

1 Introduction

With the rapid development of Internet, people are increasingly inseparable from the Internet. Among the numerous network services, online chatting is very hot. Comparatively popular instant communication systems are the Tencent QQ, foreign ICQ, AIM, MSN and Yahoo Mes2senger etc [1]. These important communication systems used the traditional C/S construction, and the user must install the client software, which causes great inconvenience to the user; secondly, in the C/S construction, software upgrading and maintenance are rather difficult, because if the system version updates [2], the clients have to install the new version software. Therefore, currently more and more instant communication systems tend to use B/S construction. Generally, the traditional B/S interactive chat system is: Client-side page is refreshed timingly, reading the news of server database. During this period, each time when the server returns a new Html page to the client-side, the client needs to refresh the whole page once[3]. When the server processes request, the client can only wait idly, resulting in a tremendous waste of resources and the impact of real-time effects and page stable appearance, leaving users great inconvenience. Taking advantage of Ajax without refresh technology to exploit new chat system can solve this problem in a good way. Ajax (Asynchronous JavaScript and XML) technology is very popular in the field of Web development technology, which is like a bridge between a Web client script and the server language so that it can operate together, communicate with each other. The use of Ajax technology development with no page brush chat system is improved both in performance and user experience. This paper takes the friends-making network instant communication system as an example[4], designs and realizes instant communication system without refresh based on B/S structure, using Ajax technology.

Z. Du (Ed.): Proceedings of the 2012 International Conference of MCSA, AISC 191, pp. 687–693.
springerlink.com

2 Overview of AJAX Technology

In the traditional Web model, due to the adoption of synchronized information transmission mechanism, when the client sends a request, the client can only waiting for server to process the request in a static way, unable to conduct any other operation, only when the server finishes processing the request, before sending the data back to the client, the client cannot carry out the next operation. The traditional web request response mode client-side needs a long time, resulting in poor interaction in the systems, prolonged time for system evaluation response, system performance and other problems, not suitable for those chatting systems of strong instantaneity, strong interaction, limited transmission data but with large amount of information. In the traditional mode, even if little date is delivered, the server also needs to return to a complete Web page, which causes redundant transmission of resource and network bandwidth occupied.

AJAX(asynchronous JavaScript and XML Jesse JamesGarrett) is a Web development model proposed by Jesse JamesGarrett[5], which is not a new technology, but the combination of four techniques: CSS, DOM, JavaScript and XMLHttp. JavaScript is the core technology, defining workflow and business logic. Using JavaScript to operate DOM to change the user interface, reorganizing data displayed to the user, and processing the users' interaction based on the mouse and keyboard. CSS provides a reusable visualization style definition method for Web page element. The XMLHttpRequest object is used to have an asynchronous communication with the server.

3 System Structure

Instant messaging system generally has two modes: one is client/server mode, that is, the sending user and the receiving user must communicate through the server; the other is the user/user mode, which means the server establishes a TCP channel to each users to, and their communication is based on the TCP, without going through the server. For the HTTP protocol, the second model cannot be applied on top of it. Therefore, this paper uses the client/server mode. In this paper, the practical application of the system is instant communication system version V2.0 of making-friends network (hereinafter called the instant communication system). The system uses Microsoft NET technology, realizing one-to-many chatting mode. Different from the traditional C/S instant communication system, users can chat with a number of friends at the same time via this system.

3.1 Key Technology

The system adopts B/S mode, the traditional three layers construction: the presentation layer, business layer and data layer. The structure diagram is shown in figure 1. Wherein, presentation layer enables client-side to use the client browser and server to interact information, using Web services component to gain server information via Ajax engine, manipulating the DOM tree page update with the JavaScript scripting language. In the business layer, instant communication engine is a core component of the system, whose components are as follows:

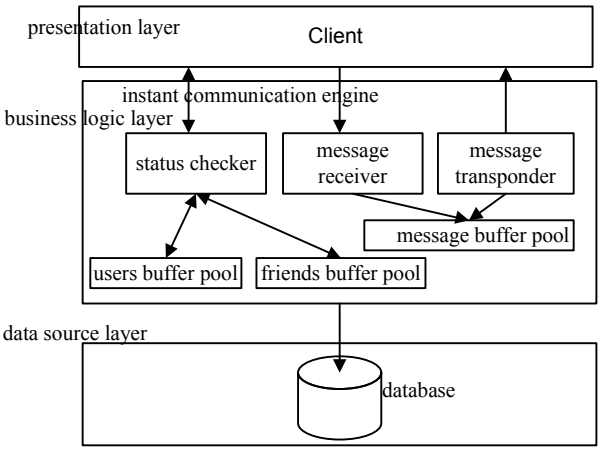

Fig. 1. System framework

(1)Status checker: Being responsible for detecting if user is online, whether to have new friend's request. The judgment is based on the user buffer pool and a buffer pool.

(2)Message receiver: All users need to send messages to the server, and the server message call distributor to distribute the message to the corresponding user. That is to say the server has the function to transmit messages. Message receiver puts the received message in the buffer pool.

(3)Message repeater: Taking out all of buddy chat log from users and their friends from the message buffer pool.

(4)Message buffer pool: Storing all users' chatting messages, managed centrally by the cache supervisor.

(5)Buffer pool: Caching all friends' information, managed centrally by the cache supervisor.

(6)User buffer pool: Caching all online user information, this is managed centrally by the cache supervisor.

All business logic operation of the layer makes all processed data packaged into XML format and returned to the Ajax engine. The Ajax engine analyzes the data transmitted by server, with the aid of DOM, DHTML to update page, display the data, and complete request response. Because all actions of the Ajax are performed in the background, which will not block the users' behavior, the users' waiting time can be saved. Meanwhile as the data transmitted and received by Ajax are relatively small, with high refresh speed, users will have the experience of real time response. From the system as a whole, all this shortens users' waiting time, and improves the system performance.

In the data layer, users' online and offline states are written into the database.

(1) The user login and exit

Users' type in instant communication engine is an abstraction of subscriber of users' basic attributes and operation, including a user's ID, all information of friends, as well as the registered observers, that is users' friends posted login or exit event. For each user's login and exit, system engine creates the corresponding online user objects and enter the corresponding registration and cancellation in Communication Engine[6]. In order to reduce degree of coupling between the User type, Friend type and CommunicationEngine type, observer pattern is taken in the design, as shown in figure 2. Among them, thematic roles (class User) are responsible for releasing events to their friends, and the observer (Friend type) subscribe events of interest. In this way, when thematic roles generate events (i.e. the user logged in or out), it can notice all the interested subscriber (Friend), in order to make its subscriber process corresponding operation, in this system, the observer can set whether their friends are online based on the event.

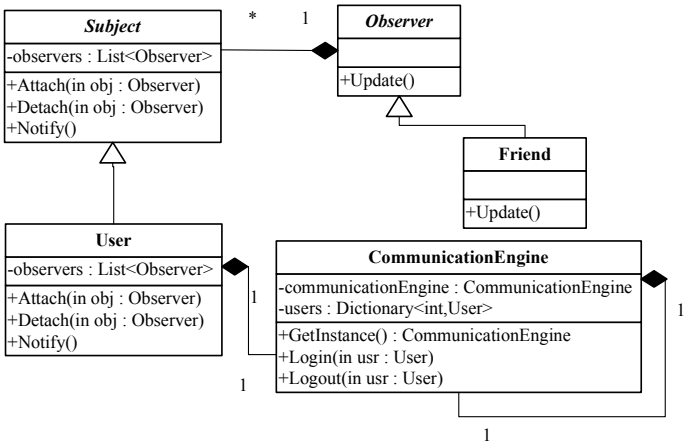

Fig. 2. Users' login and exit based on the observer design model

(2) online user management

After users log in, they need immediately access to current online users and user friend's online lists and other information. Therefore, the server needs to manage these information, and provide inquiry services. In order to realize fast search, the design in this paper uses data type Dictionary < int, User > to cache the online user object, this belongs to c# generic collection type[7]. In the dictionary, every element is a key value pair, time complexity of reading a value through a key is close to O. Because the user's ID is the unique one, system using ID as a key, and the value is the User object. Thus, according to the user ID, user object can be obtained from the user buffer pool. At the same time, in order to avoid security risks brought by the usage of global variables and reduce readability and maintainability of code, this paper adopts single instance mode to realize the online user management.

The CommunicationEngine type contains following properties:

(1) Buffer list for storing user information:
Private Dictionary < int, User > users
(2) Unique instance used to store the CommunicationEngine:
Private static CommunicationEngine communicationEngine
(3) Send and transmit messages

Sending and transmitting messages are managed by MsgStrategy, and this paper adopts the strategy pattern mainly because of the expansibility of the system, so that it can be easily extended to other algorithms for the management of message buffer pool to realize the message receiving and transmitting message. The model changes strategy of receiving and transmitting messages, having no influence on client-side. The process is illustrated in figure 3.

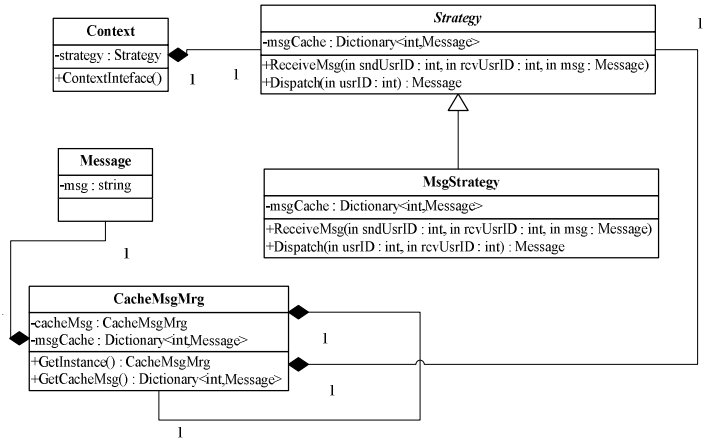

Fig. 3. Sending and transmitting messages based on strategy design model

Due to the fact that the buffer pool is a critical variable, which is mutually exclusive to write operation, this paper adopts the singleton pattern to prevent message buffer pool.

CacheMsgMrg contains properties:

(1) Buffer pool used to hold the chat log:
Private Dictionary < int, Message > msgCache
(2) Only instance used to hold the CacheMsgMrg:
Private static CacheMsgMrg msgCache

Strategy type includes the following operations:

(1) void ReceiveMsg (int sndUsrID, int rcvUsrID, Message MSG)
(2) Message DispatchMsg (int usrID, int friUsrID)

The two operations realize the message receiving and message forwarding, wherein, parameters sndUsrID and recUsrID in operation of ReceiveMsg are the user IDs of sending and receiving message, and the parameter MSG is the message sent, the operation of the user stores the the message sent to the friends into the buffer pool.

Parameters usrID and friUsrID in the operation of DispatchMsg are the user IDs of sending and receiving message, with a Message object returned.

The object contains a chat message. Because the HTTP protocol is a request-response protocol, the client-side cannot have real-time monitoring of server status changes, in order to realize the real-time message distribution, client-side sends update request actively every 5 seconds to the server using Ajax technology. If their friends send a new message, the server sends a new chat message to user for request, otherwise, return to "404" will have no new messages.

3.2 Processing Offline

When the user closes the chat window or exits the chat system, the client sends a cancellation action to the server. Chat engine releases object memory of user, and releases event to its subscribers, making its buddy update user presence. As the network is not stable and breakdown may arise in the client-side system, which makes the service server unable to accept cancellation request, and the user object offline is still in the user buffer pool. If the server is run for a long time, the memory will store a large number of off-line user because it does not receive such cancellation request, which greatly increases the system load, and even causes the system skips collapse, on the other hand, because the cache pool stores too many users, reducing the searching efficiency greatly, to solve this problem, the client-side needs to send online confirmation request to the server at a regular and certain time actively (this system uses 10 minutes). The whole process is in the charge of state inspector. If the server has not received the user's online confirmation request for a long time, the conclusion will be the users have been offline. The server removes the user from the cache, informing its subscribers of the exit event.

4 Conclusion

The realization of Instant Communication system is consisted of two parts: the server-side and client-side.The server requires a timely response to requests issued by the client-side. With the Ajax technology, the operation of chatting messages can be conducted hiding in the background, making up for the disadvantage of the traditional B/S architecture, which has low efficiency in refreshing frequently the entire page. This brings the user good experience of having no refresh, taking full advantages of the application of Ajax. This paper’s next step is how to realize the functions of file transmission, voice and video.

References

[1] Yuanjie, L.: analysis of communication protocol in major instant messenger software. Application and Research of Computers, 243–250 (2005)

[2] Jiang, Y.: Chat system research based on multithreading technology. Computer Engineering and Design, 4046–4096 (2008)

[3] Crane, D., Pascarello, E., James, D.: Ajax in Action. Ajaxcn. Org., translated, pp. 25–50. People’s post and Telecommunications Publishing House, Beijing (2006)

[4] http://www.gxqingyuan.com
[5] Garrett, J.J.: AJAX: A new approach J. to web applications. Adaptive Path (2005)
[6] Gamma, E., Helm, R., et al.: Design Patterns Elements al, of Reusable Object-Oriented Software, pp. 194–201. Publishing house of mechanical industry (2004); translated by Li Yingjun et
[7] Kauffman, J., Sussman, D.: ASP.NET 2.0 classic tutorial – C #. The posts and Telecommunications Press (2007); translated by Meng Xianrui, Yi Lei, et al.