Developing and Integrating Java Based SIP Client at Srce

Davor Jovanović and Danijel Matek

University Computing Centre, Zagreb, Croatia
Davor.Jovanovic@sorce.hr, Danijel.Matek@sorce.hr

Abstract. In order to provide multimedia services, which would allow users to participate in the modification of forms and contents of the current session in real time, we have begun to create an important component of successful online communication and education - a software based client. This kind of a smart endpoint needs to meet two conditions. The first requires the client to be platform-independent, modular and able to integrate with Web based applications. There are just a few Java based SIP clients (open source and commercial) and they do not meet the second main condition which is the process of authentication in accordance with the recommendations of the AAI@EduHr system. This paper explains the architecture of the client, the extension of authentication concept in AAI@EduHr system, which makes the client unique, as well as how we can benefit from the both.

Keywords. SIP (Session Initiation Protocol), JAIN SIP (Java-standard interface to a SIP signaling stack), JMF (Java Media Framework), AAI@EduHr (Authentication and authorization infrastructure of the scientific and higher education system in the Republic of Croatia)

1. Introduction

Starting from the importance of providing multimedia services for the purpose of allowing new ways of information access in academic community, University Computing Centre, University of Zagreb (Srce) has decided to develop its own software client based on the Session Initiation Protocol (SIP) [7]. Our final goal is to create an application allowing online teaching and learning, access and sharing of classroom-related multimedia, student counseling, collaboration and idea sharing and other services related to the process of studying. All these services are available to students and professors wherever they are provided they pass the authentication process regulated by the authentication and authorization infrastructure of the scientific and higher education system in the Republic of Croatia (AAI@EduHr) [1].

To provide such services it is necessary to have intelligent endpoints and a central infrastructure which can allow establishing, controlling and terminating of a session. It is possible to establish this kind of a model by using the SIP which "only" assists in the provision of advanced services and allows complete freedom when implementing such advanced types of services in endpoints - smart clients. Audio and video conferences, web based meeting rooms, application and desktop sharing, Instant Messaging and rich presence are just some of the services available in this type of infrastructure.

The first step was completed and today we have a functional Java based SIP client, compatible with commercial and open source SIP clients and other elements of SIP infrastructure in accordance with the RFC 3261 and its extensions.

2. Current development and comparison with other Java based SIP clients

The basic idea which has been leading us through the development is a comprehensive and intuitive SIP client, simple to use which will entirely disburden the users in the sense of its configuration and maintenance. The users are able to connect from any computer to a client web site, enter data of their electronic identity as in AAI@EduHr system (username and password) and establish audio/video communication. The users do not need to care whether or not they have the last client version, where from and how to download it, whether they have the client on a CD or some other media.

At the current stage of the development, our SIP client contains what we consider to be the basic features: audio/video communication, authentication process in accordance with...
AAI@EduHr system, it has built-in Network Address Translation (NAT) traversal solutions [8], [9], [11] and runs on Microsoft Windows. More features will be implemented in the near future according to the ongoing development plans.

In short, for the purposes of this paper, we could compare our client with three free (SIP communicator, TudoMais and Peers) and one non-free (Pingtel) Java based SIP clients. All of the mentioned clients currently support more multimedia features than our client but are developed as simple softphone applications therefore not able, without significant changes in their architecture, to be integrated e.g. with our company's services. Our client is designed more as a framework rather than a simple softphone application. This kind of design greatly simplifies the integration process in other Java or Web based applications. One example of this could be the integration with our company directory. Through the client, the user is able to access and search the list of all or some users in the organization. The list of users and their visibility are made according to the company's privacy policy. Next main difference is that the above-mentioned clients do not support the required authentication process. The two examples show advantages of our client when compared to other Java based software clients.

3. Client's architecture

The client is based on SUN Java platform and as such it can be used on all modern computer platforms. The client was developed like an applet and it runs on an end-user's computer providing a rich user interface. One of the key client elements is Java Media Framework (JMF) [4] which enables audio, video and other time-based media to be added to Java applications and applets. Tasks of JMF in the client is to capture, playback, stream, and transcode multiple media formats, and control multimedia devices. By using the Sun Microsystems' Java platform and JMF package as a base, we are developing a web based SIP client which can communicate with other SIP based clients (hardphones or softphones) and other SIP network infrastructure elements.

IETF specifies the SIP in a textual format and gives the developers a freedom to define, in any programming language, their own interface to access defined SIP behavior as described by IETF standard. For the development of a SIP client we needed an interface towards the SIP that would allow us an independent application development. After some testing we decided to go with Java-standard interface to a SIP signaling stack (JAIN SIP) [5]. By using JAIN SIP stack, the application is capable of receiving and sending SIP messages, parsing incoming messages and has the ability to access fields in SIP messages through a standardized interface, message interface, events and event semantics (Transaction and Dialog support). Apart from the mentioned features, the client has a support for functionalities of SIP protocol described in the RFC 3261, Session Description Protocol (SDP) [3], Transport Protocol for Real-Time applications (RTP) [10] as well as SIP extensions such as INFO method, reliability of provisional responses, event notification framework, the UPDATE method, the reason header, the MESSAGE method and the REFER method.

In general, the client architecture follows a so-called "Model-View-Controller" pattern. This refers to the approach where a "model", which actually represents data and operations over them, is connected through a "message controller" i.e. a controller with a part where the interaction with the user is executed. Main task of such approach is a separation of the "view" i.e. user interface from the model. This actually means that we can entirely replace the user interface, modify it and add new interfaces without having to change the model. Several user interfaces are planed to be developed, because a user will sometimes see only a SIP address on an Internet site where user can simply mouse-click on the link and invite a person on that address ("click to dial" feature). A user interface which will work only in ASCII mode has also been foreseen. Figure 1 shows basic components of the client architecture. Main component which connects all other components is the "Main Observer Queue". It represents a central system for receiving and sending messages between components. When it is separated as an object, MOQ is realized in the form of a thread which is blocked and waits for a new message to arrive. When it receives a message, MOQ notifies all "observers" i.e. object observers on the message content. Each of the object observers separately interprets the message and takes adequate measures regarding the attention of the observer. For example, when a message INVITE arrives, "sip stack" object notifies MOQ about it, and it forwards the message to "requestObserver"
object which then based on the message content initiates an adequate action from "sipStackAction" repository, and in this case is receiving of calls. Currently, the client allows audio and video communication like an applet which means that the code of the applet transfers to our computer after successful login from the web server and is run locally by Java Virtual Machine (JVM). Internet browser on our computer embeds the downloaded applet into the web page or it opens a new window displaying a user interface of the applet.

![Figure 1. Client architecture (MVC pattern)](image)

4. Authentication process

The most important advantage of the client, compared with available commercial and open source SIP clients, is compatibility of the authentication process with AAI@EduHr system. Every person in the Croatian academic community has a unique e-identity, which their home institution (faculty) assigns to them at the enrollment. This e-identity, user label or username "uid@realm.hr" and password, is used in the process of authentication and registration, and afterwards also in the process of establishing connection as SIP URI of the user.

Since HTTP Digest authentication [2] used by the SIP is not in accordance with requirements of AAI@EduHr system, we have introduced an interstep before the authentication to the SIP server, and in this process we use a token. The authentication process begins in a way that the users visit a web site through HTTPS, where they enter data of their e-identity, username, e.g. djovanovic@srce.hr and a password. The authentication process itself uses Servlet which through the use of HTML request accepts authentication query from the client and records data on issued tokens into the SIP base. After this step, in the authentication process towards the SIP server, the client uses the token and the users are not aware of the interstep because the process is automatic on the client level. The authentication process is modular and currently uses infrastructure of RADIUS servers in AAI@EduHr system. The process is shown in Figure 2.

![Figure 2. Authentication process](image)
notified through audio visual signalization about an incoming call with possibility to accept or reject it. Furthermore, employees of institutions, except IP-IP communication, are allowed to have audio communication with local and international PSTN destinations through SIP-PSTN gateway. Establishing communication, call-flow with SIP messages and provisional responses, media flow, solutions of the NAT traversal issues and other necessary processes are carried out according to the SIP specifications.

5. Future work

The services that we are planning to develop in the client can be applied in many ways in communication and education processes in the academic community and divided into four main groups: audio conferencing, video conferencing, web based meetings and desktop and application sharing. Additional value of this application in the future will be a possibility to interact, which is often mentioned as a very important component in a successful online education. This does not refer only to participation in real time at some e.g. conference through audio and video interaction, but to providing whiteboard possibilities, sending text messages, displaying presentations, exchanging documentation and all that within the same session and between selected participants of that session. Although such commercial solutions already exist, availability of such technologies in education is rather limited due to economic and technical reasons.

Final goal of the application for the educational purposes is to provide communication between students, professors and a content. Application must provide possibility to the user to decide whether it is going to be one-on-one communication or conference room model, which technologies will be used in that session, and possibility to access and exchange various multimedia contents. There is still a long way in the development of the client and the implementation of the support for multimedia services through the mentioned communication models. The services will first be implemented separately and later available together for use within one session.

Before we start with the development of new services and their provision to the academic community, it will be necessary to completely optimize the core of the client and test the already built-in features. That will be our next step.

6. Implementation challenges

In the process of providing multimedia services, it will be of great importance to observe users' reactions to each service, and based on these inputs we can work on the implementation and integration of the service which proves to be necessary. Offered services need to be simple for use and stable to avoid users' dissatisfaction which may result in the rejection of this kind of communication and education.

Regardless of the type and quantity of implemented features, the client needs to remain intuitive and simple for use also in the future, which means that logic of using similar products, that can be found in the market today, needs to be followed. Behind an easy-to-use interface, there should always be a core which meets the requirements of the SIP and other relevant RFCs, as well as intelligent modular architecture which allows for implementation of new services at any time.

One of the biggest challenges is to ensure full support to all major operating systems and platforms available today. Due to poor multimedia support for Java, this is not an easy task to do. JMF is rather old and it supports just a few modern audio and video codecs. Another problem is that JMF is not distributed along with Java so it requires separate installation. The only way to avoid this is to fully integrate JMF in the client. JMF Windows performance pack is already integrated in the client while other platforms are still not fully supported. The integration process can be very difficult especially if the client is distributed as Java Applet or through Java Web Start (JWS) [6]. The main reason for such difficult integration are native libraries that are part of JMF. There are no circular dependencies between libraries but most libraries do depend on each other. It is almost impossible to load such libraries on Linux and Unix like platforms.

These are only some of the greater challenges and it is almost impossible to predict all of them that could appear in the course of developing the client and for which it will be necessary to find an adequate solution.
7. Conclusion

Media communication using the SIP provides capabilities to architect applications over ubiquitous platforms including the Internet. Although there are multimedia enabled applications, we still need to develop our own application which can be integrated into the infrastructure and services of the academic community. The software client, described in the paper, represents efforts made by Srce to create such application and environment in which it will be possible to offer multimedia services. The paper describes the architecture of a Java based SIP client and its current development status, integration with AAI@EduHr infrastructure, possible services and methods of utilization.

8. Acknowledgments

The SIP client is currently used for audio and video communication for Srce employees.

9. References

[1] Authentication and authorization infrastructure of the scientific and higher education system in the Republic of Croatia. AAI@EduHr: http://www.aaiedu.hr/ [05/2007]


