Formal Specification and Description Language and Message Sequence Chart to Model and Validate Session Initiation Protocol Services

Sa'ed Abed, Mohammad H. Al Shayeji, Ovais Ahmed, Sahel Alouneh

***Abstract*—**Session Initiation Protocol (SIP) is a signaling layer protocol for building, adjusting and ending sessions among participants including Internet conferences, telephone calls and multimedia distribution. SIP facilitates user movement by proxying and forwarding requests to the present location of the user. In this paper, we provide a formal Specification and Description Language (SDL) and Message Sequence Chart (MSC) to model and define the Internet Engineering Task Force (IETF) SIP protocol and its sample services resulted from informal SIP specification. We create an "Abstract User Interface" using case analysis so that can be applied to identify SIP services more explicitly. The issued sample SIP features are then used as case scenarios; they are revised in MSCs format and validated to their corresponding SDL models.

***Keywords*—**Modeling, MSC, SDL, SIP, validating.

I. Introduction

IETF standardized SIP under RFC 2543 as a part of the Internet Multimedia Conferencing Architecture (IMCA). SIP was designed to join other Internet protocols such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Telephony (IP), and others. Today, SIP has become one of the significant Internet telephony signaling protocols. In this paper, we use the SIP as a case study of IP features and services due to its newly visible position. Despite the fact that RFC 2543 provides a comprehensive protocol specification for SIP, there is no formal specification of SIP [1]-[5] exists. The rules that govern the offer behaviors in SIP are defined in several RFCs [6]. The authors in [7] described some techniques to implement telephone services in SIP with less feature interactions, which is close to our work. Hence, the addition of this work is to express and design approaches to specify SIP services formally with least feature interactions. Thus, providing a formal model of SIP services to cover and handle these features is considered a new and interesting way of innovation to identify SIP services more precisely.

In this paper, we follow the specification of the SIP and develop a formal specification in SDL [8], [9]. Then, the model is simulated and validated based on test cases generated using SDL and MSCs [10], [11]. Finally, the results are conducted, summarized and validated using ObjectGEODE

The paper is organized as follows: Section II gives a summary of SIP features and the aims to develop Internet telephony. Section III overviews the SDL and MSC. Section IV introduces a complete explanation of our SDL specification of SIP model. In addition, the section details the formal design approach used in this work by applying use case analysis, structural and behavior definition, and validation. Evaluating and verifying the MSC is the main objective of this section. Finally, Section V concludes the paper and presents some trends for future work.

Sa'ed Abed and Mohammad H. Al Shayeji are with the Department of Computer Engineering at the College of Computing Sciences and Engineering, Kuwait University, Kuwait. (e-mail: s.abed@ku.edu.kw, malshayeji @ ku.edu.kw).

Ovais Ahmed was with Concordia University, Montreal, Canada. (e-mail: ahmad@ece.concordia.ca).

Sahel Alouneh is with the Department of Computer Engineering at the School of Computer Engineering and Information Systems, German-Jordanian University, Jordan (e-mail: sahel.alouneh@gju.edu.jo).

II. Overview of SIP

SIP [1], [2] is an application protocol for beginning and building sessions in an IP network. A session can be two-way telephone call or a multimedia conference session. The facility to create such a session means that many new and interesting features are possible like voice-enriched e-commerce and web page click-to-dial.

The SIP [1], [2] is established to set up, change, and demolish multimedia sessions over the Internet. The IETF Multi-Party Multimedia Session Control Working Group known as MMUSIC originally developed it. Big changes were made to the protocol and published as RFC 2543 in 1999 [1], [6].

The IETF's philosophy [3] is to identify only what is required to specify. SIP protocol is developed as a method to establish sessions without any knowledge of entire details, it just starts, ends and changes sessions. This means that SIP is scalable and fit for several architectures and deployment scenarios. Using SIP, telephone turns out to be another web application and combines with other Internet services. SIP is an easy toolkit in which service providers can use to construct combined voice and multimedia services.

SIP [4], [5] differentiates five parts of beginning, building and ending multimedia communications:

- User location: method to decide and figure out the end-system used for communication;
- User abilities: method to decide and figure out the media and media parameters used;
- User availability: method to decide and figure out the called party ability to start communication;

- Call setup: method to create call limits, guidelines for the called, and the calling party.
- Call handling: method to organize the control of the mid call and third party call as a call transfer after setting up a call and ending calls.

For our work, User location and User capability mechanisms will be out of our interest to simplify the design, instead the other parts will be covered in detail and we will try to validate them using different scenarios.

A.SIP Architecture

Two elementary modules exist within SIP protocol [1], [4]. SIP user agent that terminates the call system module has a client component called the User Agent Client (UAC). SIP server, which is a network scheme that manipulates the signaling related to several calls and has a server component called the User Agent Server (UAS). The client component starts the calls and the server component replies it. In this case, a peer-to-peer call is permitted using a client-server protocol. The SIP server role is to deliver location and name resolution of the user, since the caller has to forward messages to other servers using next hop routing protocols and to recognize the IP address or host name of the called party.

SIP follows the client/server scheme, which has shown to be effective with the Internet. SIP infrastructure can be delivered by the backbone service provider as part of the IP service offering to other service providers. Therefore, nowadays, it is possible for end-users to write applications similar to the web applications.

B.SIP Methods

In this section, we will go through an overview of SIP methods and concentrate on the methods used for our paper [7]. The methods are defined below where the method tokens are case-sensitive.

Method = "REGISTER" | "INVITE" | "BYE" | "ACK" | "OPTIONS" | "CANCEL"

Note that methods not supported by a UAS or registrar make a response returned whereas the server treats those are not supported by a proxy or redirect server as if they were OPTIONS method and passed consequently.

1.SIP Registration

The user agent is used REGISTER method [7] to inform the network of SIP about its present IP address and URL's for which it would like to accept calls. SIP registration is similar to initialization of mobile phone registration. In order to allow a user agent to use a proxy server for outgoing calls, registration is not a mandatory. It is only required when a user agent needs to register to accept incoming calls from proxies that work for that domain if the location service is not used by some SIP methods to occupy the SIP URLs and IP addresses of user agents. Even the REGISTER request usage is not well defined in the standard, a message body is included in the request.

Fig. 1 shows an example on how the database is accessed by the proxy enclosed Ahmad's current IP address. In this example, Ahmad sends a SIP REGISTER request to the SIP registrar server. The SIP registrar server gets the message and identifies the IP address of Ahmad. The SIP URL address of Ahmad is contained in the REGISTER message and saved in the database to enable the proxy server to locate Ahmad's position. In case a proxy server receives an INVITE request addresses to Ahmad, the request will be proxied to the saved IP address. Later, the registrar server grants the registration by directing a 200 OK response to Ahmad. Thus, registration is accomplished on initialization of a SIP device.

Fig. 1 SIP Registration Example

2.SIP Invitation

To create a media session between user agents, an INVITE method is used. It is identical to setup a message in ISDN or an initial address message in phone. Responses to INVITE is approved with the ACK method. In order to join the session, the INVITE request holds a session explanation, which affords the called party with sufficient data. Its message body holding the media data of the caller. In case an INVITE method does not include the data, the ACK contains the media data of the UAS; otherwise, the media data included in the ACK is rejected. In this case, a BYE message is sent by the called party to terminate the session. When the INVITE, 200 OK, and ACK message have been swapped between the UAC and the UAS, a media session is created. It remains until a BYE message is sent by either party to terminate the session.

3.SIP Bye

To end or terminate an existing session, a BYE method is used. It is similar to a release message in mobile phone. If INVITE received positive response and ACK has been sent, then a media session is established. In this case, a BYE message can be sent only by user agents who contributing in the session, not by proxies or other third parties. This is an end-to-end method, which may not include a message body; hence, the other user agent produces responses. In this case, the caller or the called party in a session can send a BYE message where it increments the CSeq. In addition, a BYE message is sent to pend request cancels the request and the

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:3, 2016

UAS issues the final response for the INVITE.

4.SIP Ack

To acknowledge the final responses to INVITE requests, an ACK method is used in such a way that other requests are never acknowledged. In this case, the CSeq number is never incremented for an ACK, but the CSeq method is altered to ACK so that it can be matched with the number of the corresponding INVITE. The ACK holds an application message body. This is allowed if the initial INVITE did not have an SDP message body. Otherwise, the ACK may not have a message body. A stateful proxy getting an ACK message must decide whether the ACK must be sent to another proxy or user agent or not, i.e. is the ACK a hop-by-hop ACK or an end-to-end ACK. To achieve this, any sent final responses are compared the To, From, CSeq, and Contact headers. If there is a match, the proxy will not forward then the ACK for this hop; otherwise, the ACK is proxied in the direction of the UAS.

5.SIP Option

To determine the availability and request about the capabilities of the server or user agent, the OPTIONS method is used. SIP proxy, redirect servers, user agent servers, registrars and clients support this method. Replying to these queries by listing the capabilities of the server or user agent. A server contacts the user agent, when the user registered and became active, can return its capability. On the other hand, a status flag such as (Busy) representing the answer to an invitation is returned by the called user agent. In this case, the server will specify the method by returning an acceptance or (Allow) in the header field. Then, the proxy and redirect servers will pass the query without specifying their capabilities.

6.SIP Cancel

The CANCEL method produced by the user agents or proxy servers is used to end or cancel a pending call. With the same header field values To, From and CSeq, the CANCEL query can terminate a pending request whereas a completed request (when the server returned a final response) will not be affected. Note that, at any time, both the user agent client and proxy client can generate a CANCEL query. In addition, the proxy has the privilege to cancel one or more parallel-search requests if the final response is not yet received. The proxy has to forward the CANCEL requests to all destinations with pending requests.

7.SIP Signaling

SIP is based on the request-response paradigm. The following sequence is a simple example of a call set-up procedure:

1. To initiate a session, the caller (or User Agent Client) sends a request with the SIP URL of the called party.
2. If the client knows the location of the other party, it can send the request directly to their IP address; if not, the client can send it to a locally configured SIP network server.
3. The server will attempt to resolve the called user's location and send the request to them. There are many ways to do this, such as searching the DNS or accessing databases. Alternatively, the server may be a redirect server that may return the called user location to the calling client to try directly. During the course of locating a user, one SIP network server can proxy or redirect the call to additional servers until it arrives at one that definitely knows the IP address where the called user can be found.
4. Once found, the request is sent to the user and then several options arise. In the simplest case, the user's telephony client receives the request, that is, the user's phone rings. If the user takes the call, the client responds to the invitation with the designated capabilities of the client software and a connection is established. If the user declines the call, the session can be redirected to a voice mail server or to another user.

SIP has two additional significant features. The first is a stateful SIP server's ability to split an incoming call so that several extensions can be rung at once. The first extension to answer takes the call. This feature is handy if a user is working between two locations or when someone is ringing both a boss and their secretary.

The second significant feature is SIP's unique ability to return different media types within a single session. For example, a customer could call a travel agent, view video clips of possible holiday destinations, complete an on-line booking form and order currency - all within the same communication session. The commands that SIP uses are called methods shown in Table I.

TABLE I
SUMMARY OF SIP METHODS

SIP Method	Description
INVITE	Invites a user to a call
ACK	Used to facilitate reliable message exchange for **INVITE**s
BYE	Terminates a connection between users or declines a call
CANCEL	Terminates a request, or search, for a user
OPTIONS	Solicits information about a server's capabilities
REGISTER	Registers a user's current location

C.SIP Call with Proxy Server

Interchanging messages can be occurred among SIP-enabled devices such as SIP phones, hand-held or cell phones. In this case, the SIP-enabled devices must be connected to an IP network (Internet). Most of SIP calls are done using a SIP server (proxy server). In Fig. 2, we demonstrate an example of a proxy SIP server work and describes the main features of this session.

In Fig. 3, we illustrate a classic SIP call with a proxy server. Here, the caller Sa'ed calls Ahmad via a SIP proxy server similar to that in HTTP and other Internet protocols. The only difference is that the SIP proxy cannot initiate or end sessions. It is located in the middle of a SIP message exchange i.e. receiving and forwarding messages between the two SIP calls. In this example, one proxy is used for this purpose whereas multiple proxies can be used in a signaling path.

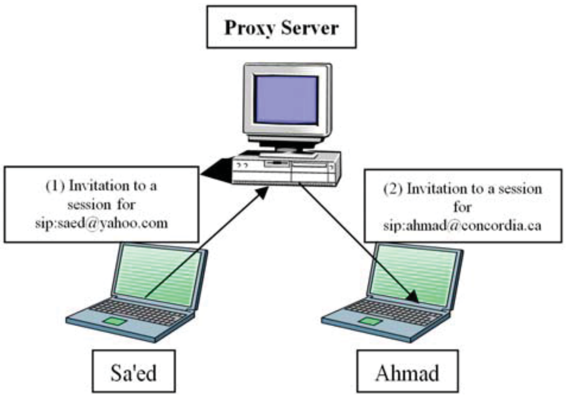

Fig. 2 Proxy SIP Server

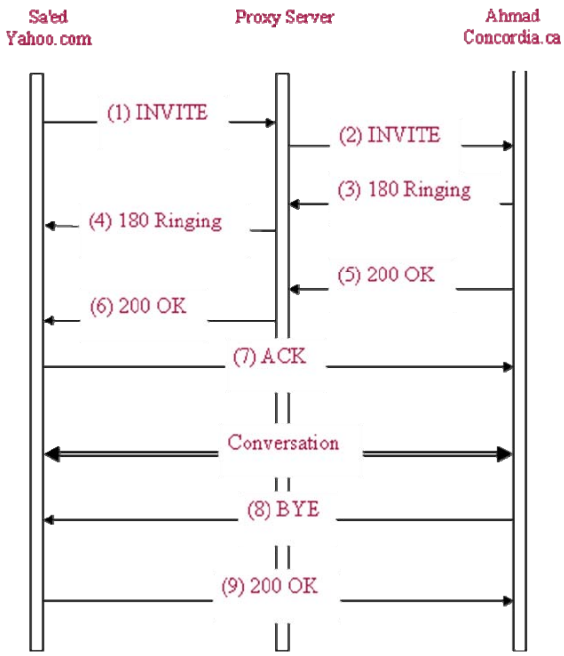

Fig. 3 SIP Call Example with Proxy Server

As the calling party, Sa'ed, does not know exactly where the called party, Ahmad, is currently logged on, a SIP proxy server is used to pass the INVITE message. The DNS captures Ahmad's SIP URL domain name (concordia.ca) and returns the IP address of the proxy server proxy.concordia.ca. The type of the call or session details (audio, video or multimedia session) are included in the INVITE message. Via, To, From, Contact, and CSeq headers characterize the necessary header set details in the SIP message.

III. Overview of SDL and MSC

In this section, we review the required material on formal software specification (SDL and MSC) constructed using the ObjectGEODE tool to afford solution for many problems:

- Deal with modeling complexity using graphical notations.
- Report the features of real-time systems using suitable standards such as MSCs (which provides scenario description of component interaction), SDL (which permits graphical design of reactive systems via Finite State Machines (FSMs)) and TTCN from ITU-T, and UML (which allows the Object Oriented specification of requests and data) form OMG.
- Generate as early as possible a high quality system using modern verification and validation methods.
- Provide automatic approaches appropriate for real-time application and control the development process.
- Maintain both the graphical design and the source code of the applications and allow reusing of the components.

A. SDL Overview

For the past two decades, several techniques were developed in academia and industry to improve the software and hardware based on formal methods.

SDL is a formal specification language developed and used in many real software applications to express the important properties of real time systems. SDL was in such a context used in the telecommunications industry as well as in real time communication systems. SDL defines the interactive environment with their internal structure of these systems. However, confirming the correctness of these systems is guaranteed using Verification and Validation (V&V) techniques.

In our work, SDL was chosen as the modeling language because it is a well-known commercial tool, which provides strong features for verifying and validating designs against MSCs. The contribution of this paper is use the capability of SDL to describe the behavioral specification of SIP protocol existing in chart-like format (informal call flows in IETD drafts) in a graphical MSC and validate the model against these charts.

SDL possesses a lot of strengths and positive qualities. One of SDL’s greatest strength is its ability to describe large, complex, and real-time systems. Some other salubrious qualities are enumerated next. First, SDL is a nonprietary internationally standardized language. Second, SDL is a formal language. As such, the semantic underlying every symbol and concept is clearly, unambiguously and precisely defined with mathematical rigor. This is extremely important for most technical systems –especially mission- critical ones. Third, SDL is graphical and symbol-based. Consequently, it is very intuitive and easy to learn. Even non-users can quickly discern a system’s structure and behavior with little formal training in using SDL. Fourth, SDL is object-oriented (OO), which means that it supports encapsulation, polymorphism, and dynamic binding. Fifth, SDL facilitates maintenance and reuse for large projects as well as permitting information sharing between multiple systems.

B. MSC Overview

MSC is a graphical representation used to describe interactions between entities and communication among processes whereas SDL defines the structure and behavior of the system. It is based on describing the execution trace of the system to show the messages/signals passed among communicating components using charts. Thus, an MSC can capture a small portion of the system behavior and use one case scenario of the system. We have chosen MSC to depict case scenarios due to the following reasons:

1) Using Case Diagram is considered an important user requirement notation.
2) MSC is used for capturing execution traces.
3) It is easy to derive execution traces (scenarios) in MSCs format from the use case diagrams defined for SIP.
4) MSCs integrate well with the SDL modeling language.

IV.SIP Modeling and Implementation

This section provides a description of a behavioral and organization of the SIP model. First, we illustrate our design methodology, and then describe the use case analysis, which consists of case scenarios. Then, we clarify the structural and the behavioral of the service model. Lastly, we discuss different scenarios to cover the main design features to verify and validate our model.

A.The Structural Model Definition

In this subsection, we describe the SIP entities structural definitions. The SDL specification consists of the system and its environment. It explains the reaction of the system to signals and messages with the environment. All interactions and process instances have equal rights and form the behavior of a system. The process instance can issue other processes. SDL demonstrates the static connection between entities, interfaces, and aspects that represent the structural definition of SIP entities. The communication between user agents and proxies is identified by different instances (signals or SIP messages) conducted among them. Note that, SIP entity (user agent and proxy) is represented as block and process.

Client and Proxy are the main SIP entity components extracted from the RFC. We tried to extract the minimal architecture to model the SIP protocol. Moreover, we tried to follow the specification mentioned in the SIP as can as possible starting from receiving a message, parsing and extracting the fields and the headers and then send it as a one-block message. Fig. 4 shows the system diagram of proxy and user agents (clients).

As shown in Fig. 4, since we build the main components, it will be easy to add other users and proxies. We select four clients and two proxies to represent different cases of scenarios that can be generated from the design.

A SIP client (user agent) acts as a UAC or UAS in a SIP model and a block type, which includes a process instance to define the entity real behavior, models an instance of SIP entity (user agent or proxy). Each entity has variables (permanent and temporary) for its tasks. The process file reports all behaviors to where the SIP entity has contributed.

A proxy contains both client and server. It interrupts incoming messages, adds messages, adjusts forwarding messages and passes messages between user agents. A proxy is considered as an entity with organized features. It is called a stateful proxy if a proxy tracks a feature of a session; else, stateless proxy [12].

In our work, the SDL is used to characterize various interactions between SIP entities and to simulate different scenarios. The SDL model includes client and proxy block initialized with one process instance. A client replies with the corresponding response messages based on a SIP request (as well as acknowledgements).

B.The Behavioral Model Definition

The communication relation between user agent and proxy processes forms a SIP feature or service where every process instance has a specific task in the service instance. The process instance includes state transitions to model and describe the instance behaviors in the service instance.

In this work, we represent the SIP entity behavior using SDL process sort. Then, we represent the functionality of the basic service of SIP signaling. In the case of proxy process, it exchanges message among user agents or proxies. The proxy entity is deployed as in telephone service. In this case, the proxy executes specific tasks according to the information included in the message header fields. The SIP services and states model are constructed from MSC charts where features and services as time-out, sending messages, etc. are designed and modeled in SDL.

C.Core System Features

There are many SIP features, we will try to cover some of the most important features and present the MSC's scenarios for these features.

Call Screening

Call screening permits the proxy server to block and reject calls from a specific list of numbers. Let us take as an example where a screened number calls the user and the caller gets an unauthorized signal. With this feature, the number and the name of the user will not be shown on the display equipment of the called party. Fig. 5 illustrates the screening call scenario:

- User client1 calls User client3.
- The proxy server blocks the call.

Time-Out

Call timeout allows the user to wait for a period if no answer, and then return the call back with a timeout message. Fig. 6 demonstrates the time-out scenario:

- User client1 calls User client2.
- User client2 No answer.
- Return Timeout to User client1.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:3, 2016

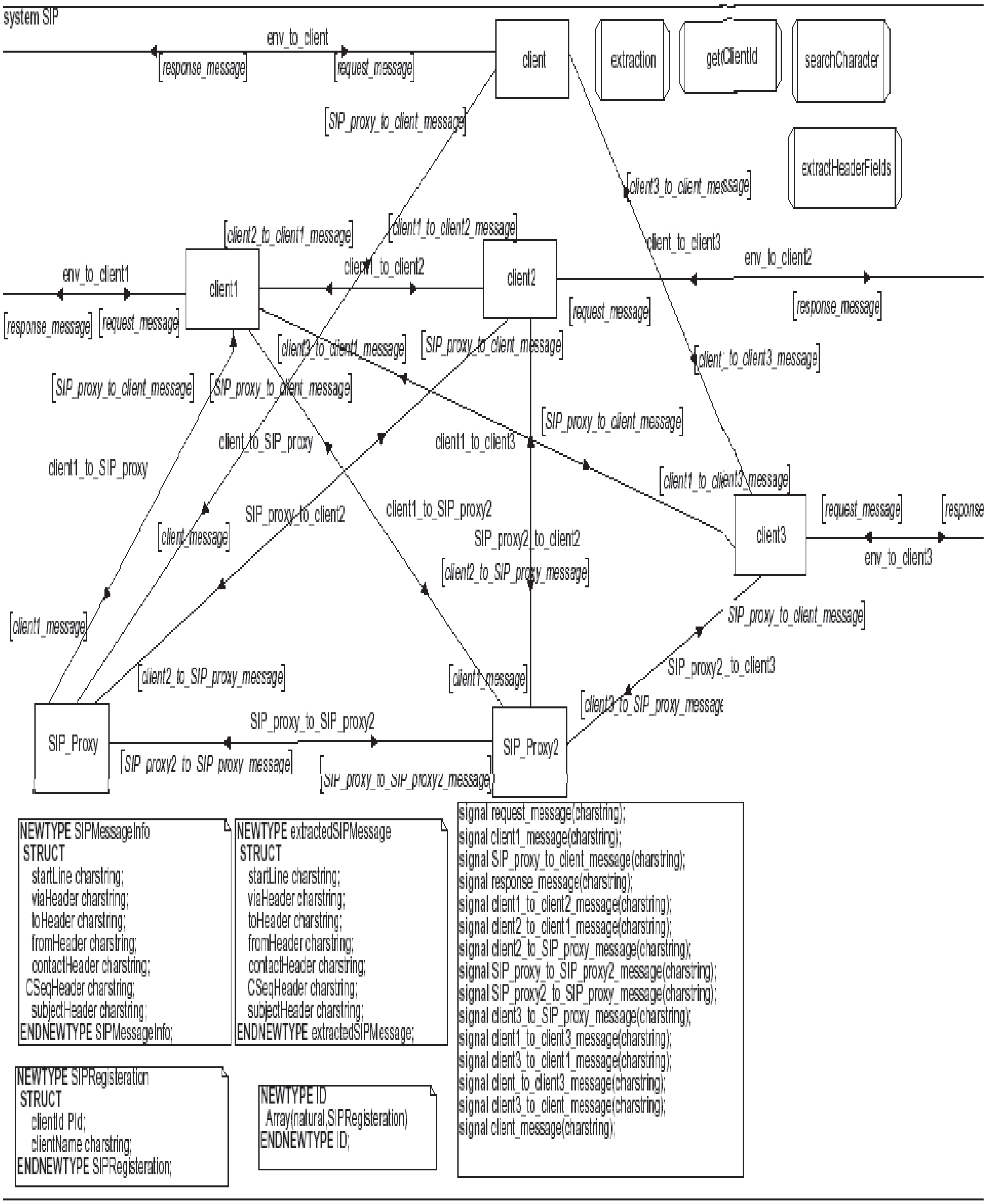

Fig. 4 SIP System Diagram of Proxy and User Agents (Clients)

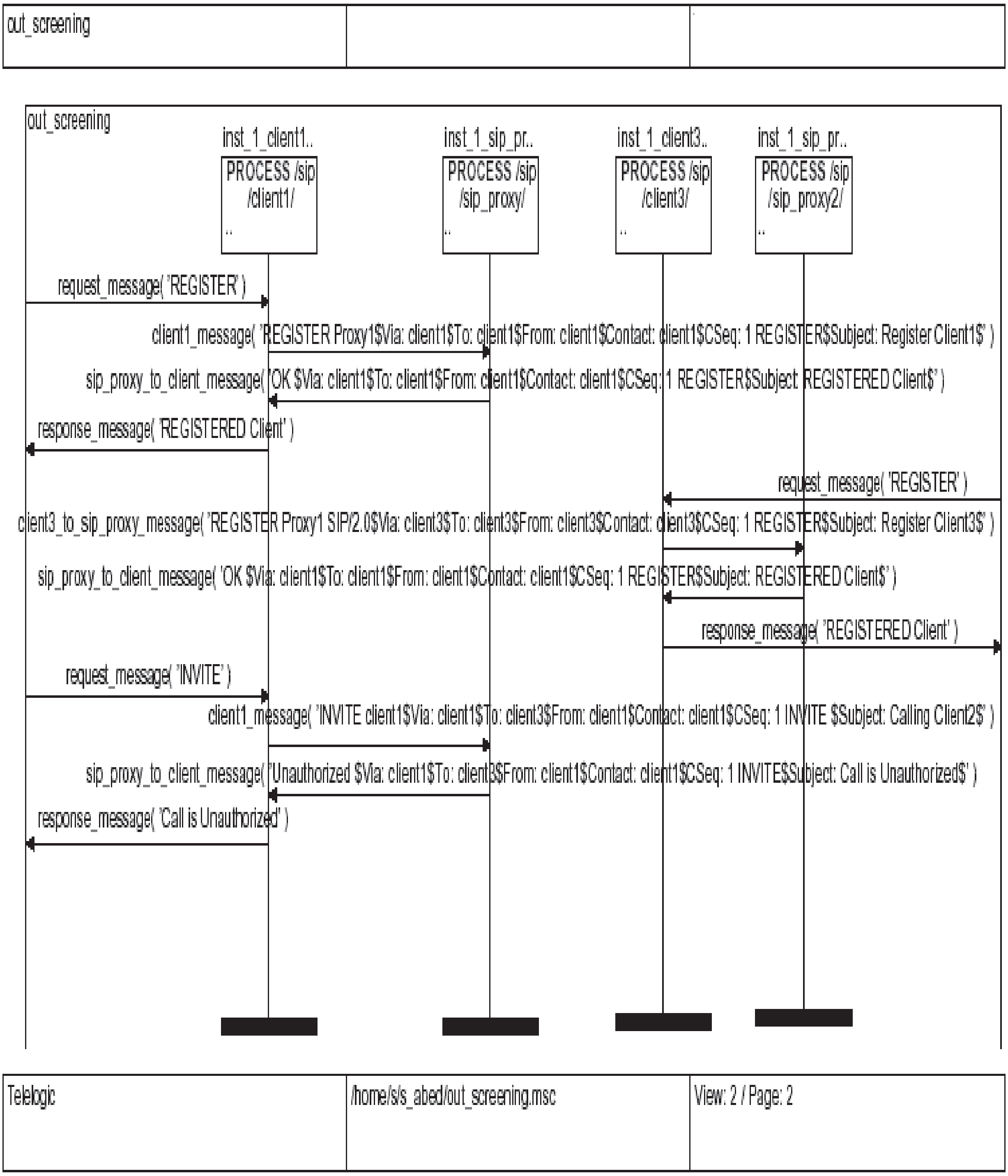

Fig. 5 Call Screening Scenario

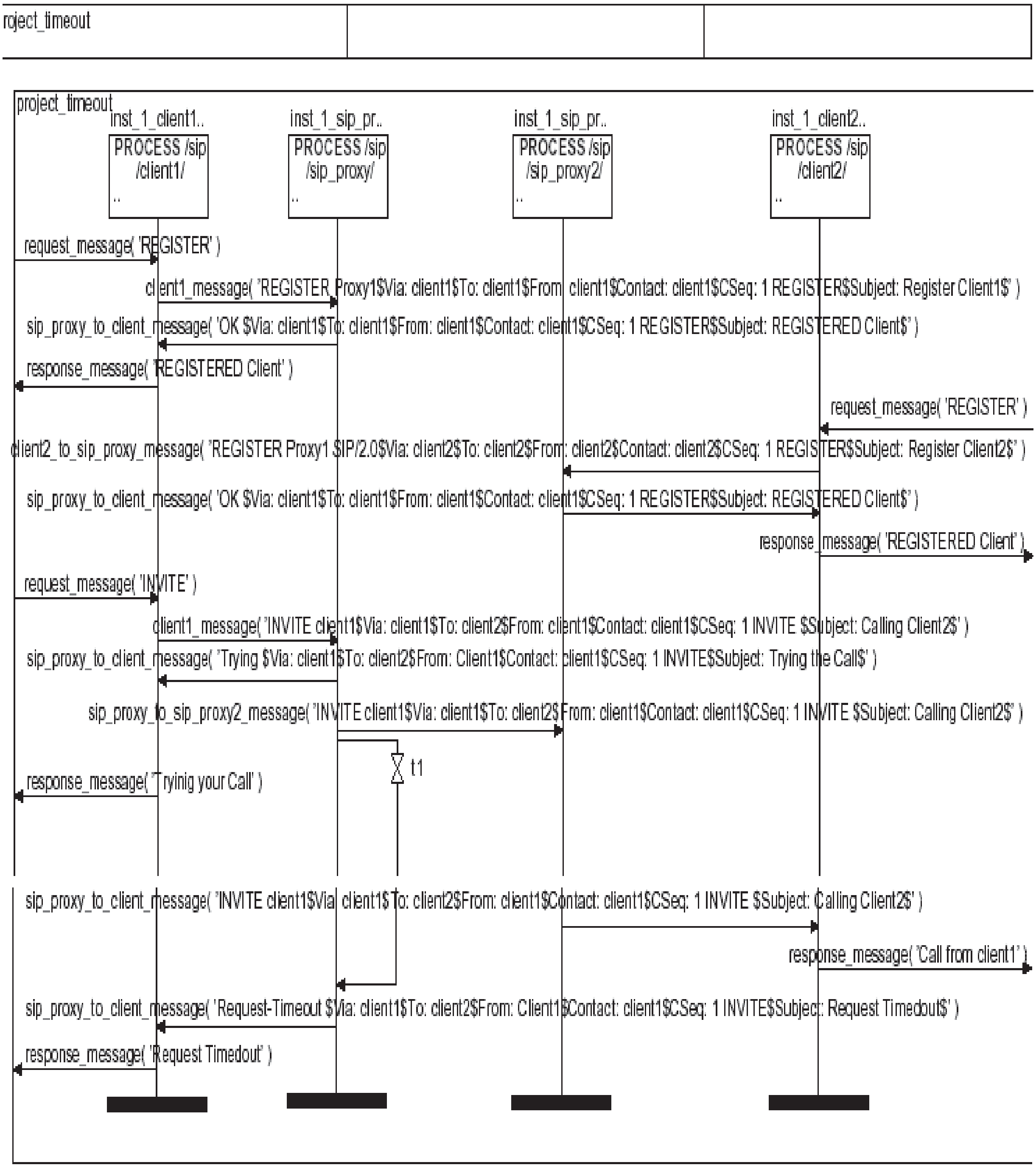

Fig. 6 Time-out Scenario

V. Conclusion and Future Work

In this paper, we provided a formal model of SIP protocol using SDL language to cover the important features and services of SIP. We discussed the importance of using a formal language as SDL and showed its advantages and disadvantages. We provided the usage of SDL to model and IETF applications protocols such as SIP signaling protocol. We used also MSC to characterize some SIP features and to verify the SDL model of SIP.

As a future work, we plan to extend our model to include other features to support SIP IETF standard. Our target is to make our model useful and available to other researchers in academia and industry sectors since the model is easy to adapt and modify to address new standard.

REFERENCES

[1] Alan B. Johnston, *SIP: Understanding the session initiation protocol*, Artech House Telecommunications, 2009.

[2] Henry Sinrreich and Alan B. Johnston, *Internet communications using SIP: Delivering VoIP and multimedia services with session initiation protocol*, Willey Computer Pub., 2006.

[3] A. Johnston, R. Sparks, C. Cunningham, S. Donovan and K. Summers, *SIP service examples*, Internet Draft, Internet Engineering Task Force, pp. 1-167, 2008.

[4] James Wright, “SIP: an introduction,” *Technical White Paper, Konnetic, pp. 1-8,* 2011.

[5] Pavel SEGEČ, “SIP services using SIP servlet API–the infoline service,” *Journal of Information, Control and Management Systems, Vol. 8, No. 3, pp. 227-236,* 2010.

[6] S. Okumura, Softfront, T. Sawada, KDDI Corporation and P. Kyzivat, *Session initiation protocol (SIP) usage of the offer/answer model*, *RFC 6337, http://tools.ietf.org/html/rfc6337, August* 2001.

[7] K. Y. Chan and G. v. Bochmann, “Methods for designing SIP services in SDL with fewer feature interactions,” *In D. Amyot and L. Logrippo, editors, Feature Interactions in Telecommunications and Software Systems VII*, IOS Press, Amsterdam, pp. 59-76, 2003.

[8] Roberto Saracco, J. R. W. Smith and Rick Reed, *Telecommunications systems engineering using SDL*, Elsevier Science Inc., 1989.

[9] Laurent Doldi, *Validation of communications systems with SDL: The art of SDL simulation and reachability analysis*, Whily, 2003.

[10] J. Lennox and H. Schulzrinne, “Feature interaction in internet telephony,” *Sixth Feature Interaction in Telecommunications and Software Systems Workshop*, IOS Press, pp. 38-50, May 2000.

[11] Ligang Wang, J. William Atwood and Anjali Agarwal, “Validation of SIP/H.323 interworking using SDL/MSC, *SDL*”, *System Design. Lecture Notes in Computer Science,* Vol. 2708, pp. 335-351, 2003.

[12] D. Seo, H. Lee, E. Nuwere, Sipad, “SIP-VOIP anomaly detection using a stateful rule tree,” *Comput. Commun.* 36 (5), pp. 562–5, 2013.