Asian Journal of Information Technology 5(8): 890-892 , 2006

Proposal of Extensions to Electronic Mail Client Applications

[1]Mohammad A. Al-Rababah, [2]Mohammad A. AlMaaitah and [3]Ahmad A. Al-Rababah
[1]Philadelphia University, Jordan
[2]Jerash Private University, Jordan
[3]Faculty of IT, Al-Ahliyyah Amman University, P.O.Box 252, Amman-19328, Jordan

Abstract: In this study we present the proposal of functional extensions to email Client (and alternatively also transmission Protocols). Changes shown are supposed to help people working with email programs in different places, trying to get the same emails at home, work and other locations .we also propose methods for reducing the number of emails sent without the intended attachments.

Key words: IMAP, POP3, SMTP, email, clients, extensions, attachments

INTRODUCTION

Popular emails became practically inseparable part of every modern human life. Their growing success caused dynamic evolutions of service and client applications bearing with exchange, editions and presentations of electronic message but there are still many things to done in this field to allow users work effectively in different scenarios[1].

REDUCING NO-ATTACHMENT PROBLEM

The no-attachment problem touches almost every user of email program. No matter whether you are beginner or experienced user it probably happened that you send important email without intended attachment. Currently there are no solutions, which lets reduce this problem. We would like present two solutions, which can be used together. Figure 1 presents diagram showing the idea of the first one. The standard procedure is: press New message button, edit your email .press send button and then program sends email via SMTP protocol to the server. We propose modifications to this procedure. When user press send button the client applications look for some characteristic key-word are: Attached, attachment, sending, included. There should be possibility to extend and localize the list .if any key-word is found then attachment existence is being checked .if it exists the message is sent. Otherwise the user is asked whether he really wants to send the message without attachments, as there are words in the text, which allow guessing he is not.

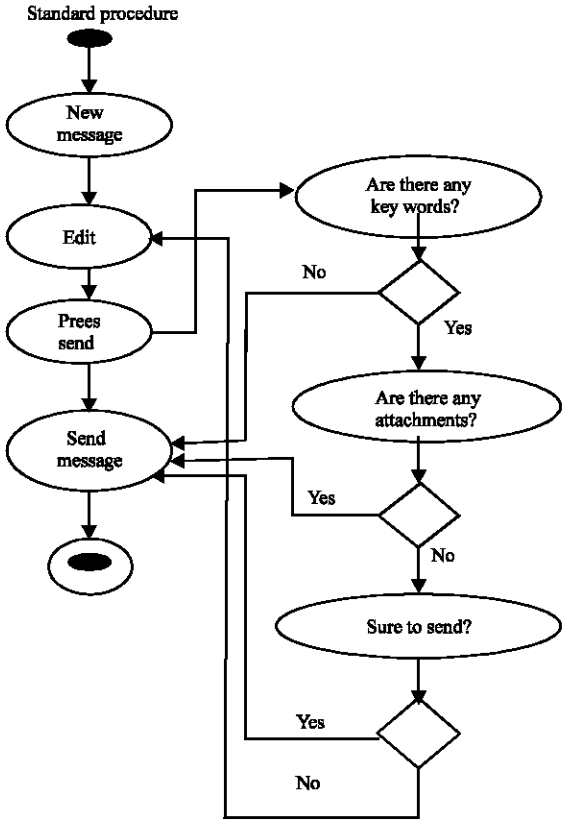

Fig. 1: Sending email procedure-extension

The second solutions are much simpler. The diagram is presented on Fig. 2. it is based on addition of a new

Corresponding Author: Ahmad A. Al-Rababah, Faculty of IT, Al-Ahliyyah Amman University, P.O.Box 252, Amman-19328, Jordan

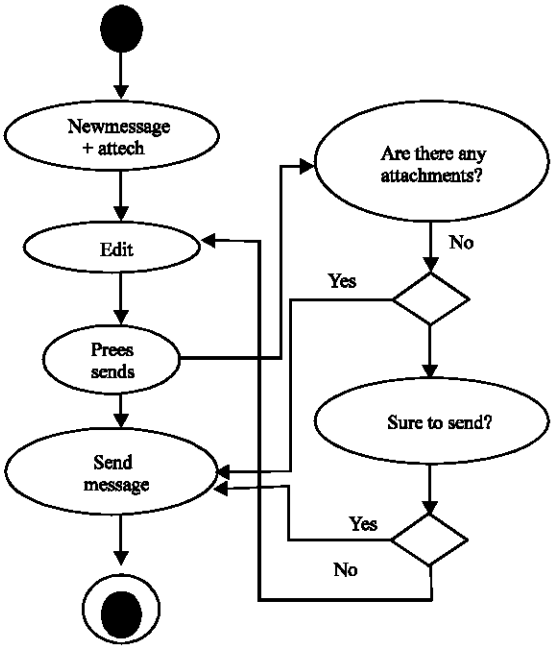

Fig. 2: Sending E-mail procedure-scenario 2

button New message + attachment (or similarly named). User by pressing this button state that he is going to create a message with attachment. if there is no attachment when send[2].

MULTI-LOCATION PROBLEM

There is a group of people whose work makes them deal with emails in many emails at their work, home and university. To do that some of them need the history of sent/received message in every locations. Such people usually cannot number of message grows rapidly and the quota start being a great's limitation in a short time. Moreover such repository is connections to the server .so there is the requirement to get information from the server and store it locally on client computers.There exists solution for such users. By using IMAP protocol instead of POP3 clients can better control sent and received message .unfortunately it has some disadvantage. The most important is that it is not supported by all ISPs (the statistics based on some analyses of offers found on internet allows to state that this may be even majority of ISPs).it is also inaccessible in many institutions because the security politics disallows passing IMAP packets through firewalls as not being required (the method of reducing Rick).IMAP also less popular as it is said to be more difficult. Ironically, because there is such opinion, the better method is less popular and as a result sometime not a possible option[3].

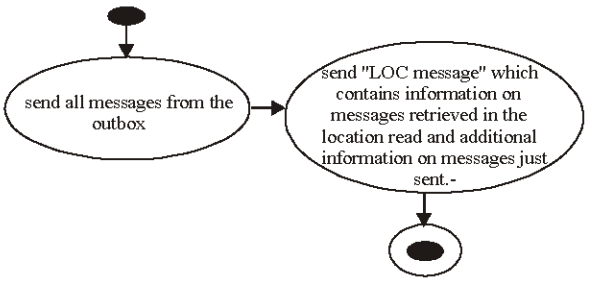

Fig. 3: Sending e-mail procedure-extended

Because of the above analysis we decided to find solution for people who con not or are not going to use IMAP and need extensions which would improve their work. We can propose tow alternative solutions. The first one needs changes only in the client applications. The second requires extensions to POP3 and STMP protocols(what implies changes in the server). The first one is rather workaround and the second is more professional final solutions.

Both methods require the user to define all the locations there he is going to access his mail server and current locality information in every client application he uses {this article does not touch web based methods}[1].

The user is give whole the interface to use additions functionality (rather sophisticated-allowing for example to decide to (or not to) retrieve email being sent himself in particular location} but he is not forced to utilize it. All he is to do as to put mentioned information (definition of localities) everything else is being covered by proposed algorithms). This way there is almost no need fir getting additional skills.

Option 1-no server changes: In this option all necessary charges bleed to be made only to the client application Fig. 3 presents the process of landing messages (SMTP protocol). Fig. 4 shows the way messages are retrieved from the server (POP3 protocol).

The message we call LOC-message is an additional message created automatically by client application. Its body part is empty. All the information is placed in the header. We defined a set of fields (optional fields as defined in RFC 822), which start with LOC- prefix. Their role is to inform client applications in other locations of retrieved /read/not to be retrieved messages.

Moreover, LOC, SUID, field is added to all message send the client. it is defined as : LOC-SUID:<SUID> where <SUID>is a unique string hard to guess(can be created on a base of date and time).client reading header analyzing this field decide whether it created the message if so-not retrieve such message (client can not guess what UID will be given by the server, so it needs to add the information to the header of messages it sends)[2].

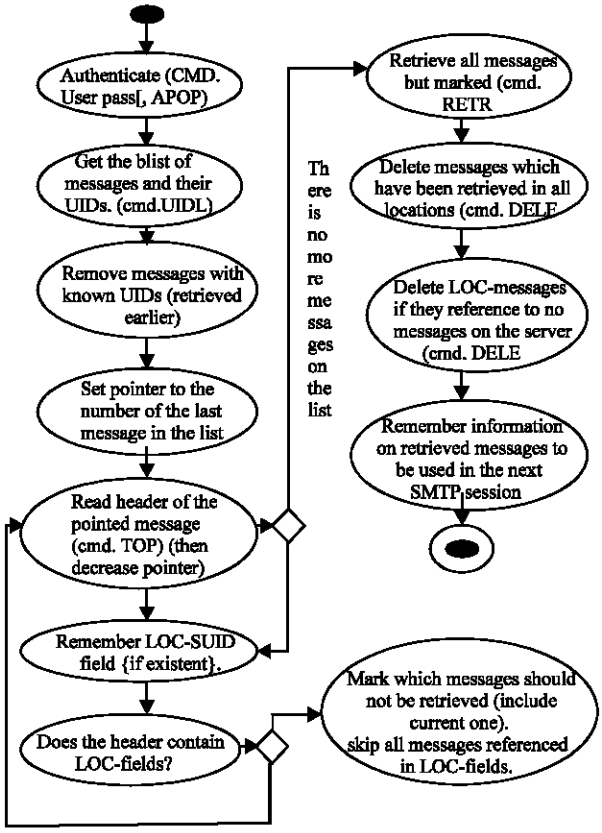

Fig. 4: Receiving e-mail procedure-extended

Client adds source address to BCC field so all sent messages re put also into the inbox and can be downloaded by clients in other locations. When clients applications find such messages put them into sent messages folder (it is not shown on Fig. 4). It allows to replicate sent messages folder. If the user does not want replication, BCC field is left empty.

Next field LOC- RETR: <location><UID>

Informs that the message identified by <UID> (returned by UIDL command -POP3) was retrieved in location <location>. <Location> should be a one word identifier of location, e.g. Home, work, university. Client according to the defined list of locations and information given in LOC-RETR field can decide which messages were retrieved in all locations and remove them from the server

Field LOC-READ: <location> <UID>

Informs that, the message identified by <UID> has already been read and can be marked after retrieval as read.

LOC-SKIP: <location> <SUID>

tells client in location <location> not to retrieve the message identified by <SUID>.

LOC-TO: <location> <SUID>

Informs that the message should be retrieved only in a given location

LOC- SUID2UID: <SUID> <UID>

Lets cache information which message is identified by SUID. It can slightly speed up messages retrieval.

Fields LOC- SKIP, LOC- TO and LOC- SUID2UID will usually appear in messages send by the user himself. He may also change the status of the outer messages.

Option 2- changes in server: The solution presented above required the use of additional messages (LOC-messages). There would be no such need (it would look nicer) if server implemented new algorithms (so SMTP/POP3 commands). All required additional information would be accessible at protocol level (not in headers of retrieved messages). All necessary information would be stored and handled in server's internal structure.

The solution would require changes in both SMTP and POP3 protocols and construction of the server. The analysis of advantages/disadvantages unavoidable difficulties in implementation (long path required to extend standards then implement it in client and server applications and long unpredictable time of migration towards new solutions) made us think it is unnecessary to do it in this way. The cost of getting additional functionality is just too high[3].

CONCLUSION

In this study we presented possible solution to no-attachment problem. It does not eliminate this difficulty but should perfectly reduce it. Unfortunately still does not help in cases when not complete set of files is attached. We also proposed a method, which is awaited by smaller group of users but quite easy to implement (needs few extensions to client applications). Despite rapid development in field of information technology, there is still need to look for workarounds of existing problems, as new and intentionally better technologies are not being implemented fast enough.

REFERENCES

1. RFC, 821, Simple Mail Transfer Protocol
2. RFC. 1939. Post Office Protocol-version 3.
3. RFC, 2822. Internet Message Format.