

A Comprehensive Attachment Repository for E-Mails

H. Ekanayake
Department of Computation and Intelligent Systems
University of Colombo School of Computing
35, Reid Avenue, Colombo 7, Sri Lanka.

E-mail: hbe@ucsc.cmb.ac.lk

Abstract

Email has become one of the crucial medium of communication for today's information society. As a result email system has improved in various ways to provide more privacy, capacity and accessibility. However the attention that has been gathered for improving the manageability of the content is very less. This has become evident when someone wanted to search for an attachment, sent by the person x, six months ago. The effort becomes worst if that person has already deleted the message to preserve his mail box capacity in a favorable limit. This paper examines the current email framework and proposes a global solution to enhance the easy retrieval of email attachments without introducing any improvements to email client programs. The solution uses the descriptive power of email messages to retrieve keywords to annotate email attachments, and later, one could easily locate a document recalling the memory he has had when the document was an attachment to an email message.

Keywords: Electronic Mail, Email, Attachment, Repository, Search

1. Introduction and Motivation

Today a major part of communication happening over the world has grasped by electronic mail, or email. Lot of factors has determined its demand, such as, its low cost and its speed of communication. Email systems are available mainly in two forms, POP mail and web mail. The latter has now becoming very much popular since it does not require additional client software to be configured in a local computer other than an Internet browser. In addition, email service is provided in many forms, such as,

free email service from Yahoo or Gmail, corporate mail, or packages from email service providers.

Earlier email was suffered as an unreliable communication medium due to its absence of providing a guarantee for delivery or security while in transit. However, over the past few years most of these shortcomings were fixed after introducing lot of features into the email system [1, 2]. A recently addressed drawback of email communication is the guarantee for delivery or the repudiation issue. This problem has been successfully resolved after introducing a notarization service for email system [3, 4, 7]. As a result today emails are used by government agencies and other organizations for their formal communication needs. Another advantage of using emails for such purposes is its demanding accountability. The header part of an email message provides a full description of the message, such as, to whom it has been sent and by whom, the constructed date of the message and a short summary of the subject. Emails also provide a facility to attach files and as a result important documents are exchanged over inter-organizations or intra (within)-organization through emails. This trend has created a situation where people do not tend to delete old messages from their mailboxes, since they think that these messages and their content might be useful in the future. The result has created a big problem; people are running out of their mail box capacities!

2. Potential Solutions

Literature was unable to show any similar system that could provide an easy attachment retrieval facility. However, the web disk facility [5], provided by today's web mail systems, has some potential of providing lot of related services. Web disk is a separate storage of hosting email attachments and documents temporary. Since its storage counts against the capacity of one's web mail

capacity, the documents cannot be saved for longer periods. Furthermore, there is no comprehensive search facility provided for these documents.

Yahoo! Desktop Search [6] is a recent attempt that has been launched by Yahoo! Inc to enable online and/or offline search facility to find items, such as, files, email messages, email attachments, instant messages and contacts related to some user. While it perform the searching, it looks for items in Microsoft Outlook, Outlook Express, Yahoo! Messenger or other personal folders for some pre-defined item types. Then it presents the results in a much organized interface, so that one can easily identify the required item(s). This can be seen as an improvement to traditional search facilities provided by operating systems, where in addition to searching files in the local computer the new search looks into email programs. However none of the solutions are using the power of email messages discussed in this paper.

3. Aims and Objectives

The following aims and objectives have been identified for implementation:

- [O1] Provide a transparent service to the users: This is to prevent additional software installations on user's local computers.
- [O2] Enable the service for wide spectrum of email systems: Anyone who is using POP mail, web mail or any other email system should be able to receive the service.
- [O3] Multiple email accounts capability: A user can has more than one email accounts; however there should be only one attachment repository instance for that user.
- [O4] Capture as much as keywords from an email message to provide a much more elaborative searching facility.
- [O5] Provide a comprehensive results sheet as a response to a search.
- [O6] Provide an alternative way to attach documents to an email message, so that the sender has some control over the attachment even after the message has been sent.
- [O7] Reliable and secure service to the users.

4. Global Attachment Repository Approach

4.1. Advantages of Email Attachments

The following are some of the advantages that could be gained by preserving the documents as attachments to email messages:

- **It enables availability of the document in global scope:** Since most email services are globally available, the same accessibility applies for the attachments since the attachments are also part of an email message. Therefore if someone sends a document to some other person, by attaching it to an email message, the second person gains the ability to download the document to his computer despite of his location (at his working place or at his home) using the same steps.
- **Attachments are sufficiently described:** Attachments come bounded to an email message. The header part of that message contains information to describe who has been sent the message, who are the recipients, the date the message has been sent, subject line containing a summary of the message and other optional information (priority, path). The body of the message might contain information describing the attachments.
- **Attachments are touched! Therefore one can easily remember them:** Normally no one ignores an incoming email message, unless it is a spam. Since an email message is observed, the observer sees the content of that message including any attachments. This attempt triggers the observer's memory to remember the email message. Later, when he wants to find a document, which has been arrived to him as an email attachment, immediately he could recall the memory to trace the sender, date, subject line ... and would surf his mail box to get the email message and then the document.
- **Reliability:** Locally stored documents are very much intensive to loose, because of unorganized habits, unintentional deletions, modifications and other misconducts. However, the potential to loose an email attachment is very less.

4.2. Interacting with the Global Attachment Repository

This section mainly focuses on the techniques which have been used to fulfill the aims and objectives identified at the beginning of the work.

Figure 1 depicts the technique which has been used to preserve the objectives O1, O2 and O3 (provide a transparent service to the users, enable the service for wide spectrum of email systems and multiple mail account capability).

If a user wants to send an email message to some other party, thinking that the attachment(s) would be useful for him in the future, all he has to do is to mention the GAR server email address (sec@ucsc.cmb.ac.lk) in the BCC address field of that email.

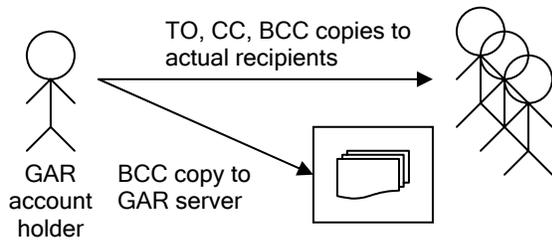


Figure 1: Posting an email

Here BCC address field is chosen, since email addresses mentioned under this field are hidden to the recipients (however there is no restriction to mention the email address under TO or CC address fields). This technique avoids the requirement of additional software to be installed on client computers in advance and as a result users who are using any type of email system can receive this service.

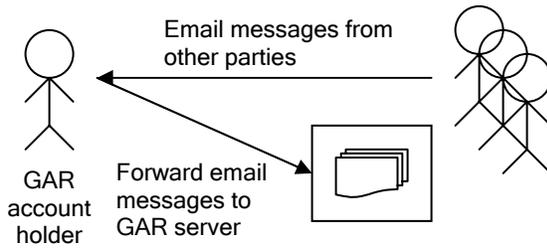


Figure 2: Forward email messages received from other parties

Figure 2 depicts the steps needed on a situation where the user receives email messages from other parties. However the user wishes to store them under the GAR server. The steps in this situation are also simple as forwarding the required email messages by mentioning the GAP server's email address as the recipient.

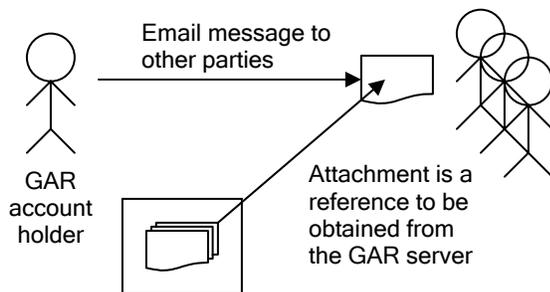


Figure 3: Inline URLs to download attachments

Figure 3 depicts how the objective O6 (Provide an alternative way to include attachments) is fulfilled. The expectation of this objective is to reduce the size of an

email message by offloading the attachments, but provide an alternative way to get those attachments when needed. The solution is to provide inline links (URLs) on the body of the message to download the attachments from a remote location. In this method the sender first sends the documents he wishes to attach to an email message to the GAR server in advance to constructing the actual email. Then the GAR server stores the documents in its repository and responds the sender with an email message containing links (URLs) to those documents. Later, without modifying those links (URLs) the sender fills out the body with his message and posts it to the actual recipient(s). The recipient(s) then follow the provided URLs to download the attachments. This method also adds an extra control over the attachments to the sender, for instance, the sender can modify or change the attached documents without the knowledge of recipient(s).

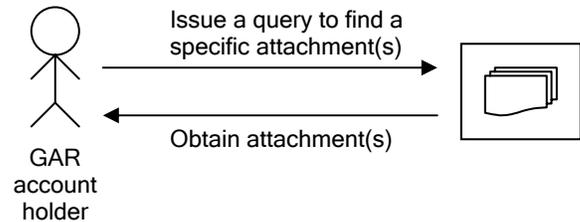


Figure 4: Searching for saved attachments

The most important feature of this architecture is the ability to retrieve a document that was an attachment to an email message, by tracing ones memory associated with it, which is depicted in Figure 4. The user issues a query using the options provided by the GAR server web interface. The server then responds with a nicely organized results-set where the user can easily identify the expected documents and download.

The GAR server captures as much as keywords from the email messages arriving to its mailbox to provide an elaborative search service to its users. The provided information would include the date, recipients, subject, a short summary of the content, details of the attachments and a hierarchical conversation like history if possible.

A SSL or similar secure channel could be used to implement a transport level security during the session of user interacting with the web interface. The GAR server also drops any unsolicited emails that come from unregistered users.

5. Prototype Implementation

The system is designed based on an architecture comprised of modules which is depicted in Figure 5.

POP Connector: This module is responsible for connecting regularly to a preconfigured POP mail server

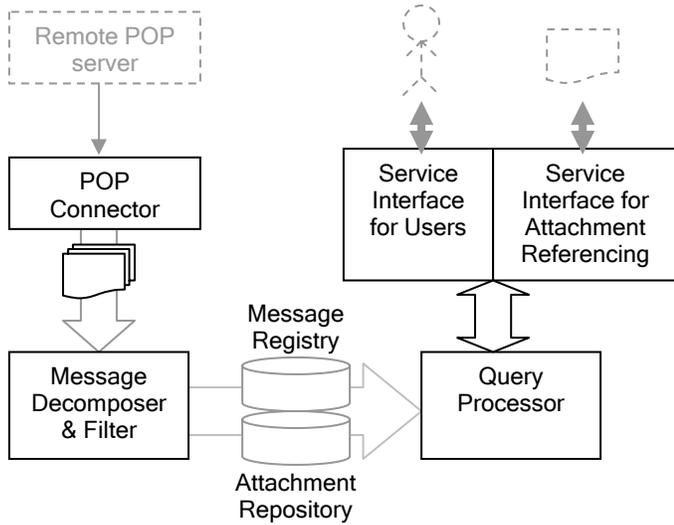


Figure 5: System architecture

to retrieve newly arrived email messages (as a result of copying or forwarding). The downloaded messages are then forwarded to the *Message Decomposer & Filter* module to process.

Message Decomposer & Filter: This module first extract the header part of a message to retrieve the details associated with *From*, *To*, *Cc*, *Date* and *Subject* fields. The address associated with the *From* field is validated against the registered users of the system. The verified messages are further processed to see whether it has any attachments. If the message has attachments those attachments are extracted from the message and placed under a preconfigured attachment folder (*Attachment Repository*). The details of successfully processed messages are saved in the database (*Message Registry*). All the unsuccessful messages are discarded.

Service Interface for Users: Registered users are interacting with the system through the web-based interface provided by this module. When a user wants to search for an attachment(s) he has to describe the email message as much as he can, recalling his memory, in the search window depicted in Figure 6. As a result the user is presented the matches found in the results window depicted in Figure 7. The matches are ordered by its rank. The user can then download the attachment(s) simply by clicking the URL(s) of the document(s).

Service Interface for Attachment Referencing: This module responds to attachments whose URLs are placed in the body of an email message as depicted in Figure 8.

Searching for Attachment(s)

Please describe the email message you want to find from the repository using keyword you remember :

Please note that you can use only **maximum 25 keywords** and when you are using multiple keywords **separate each keyword by a space!** Also just leave a field blank if you can't guess any keyword.

From?	<input type="text"/>
To or Cc?	<input type="text"/>
Date? (MM/DD/YYYY)	<input type="text"/> <input type="radio"/> Before <input type="radio"/> Between <input type="radio"/> After
Subject?	student <input type="text"/>
Content?	<input type="text"/>
File Name or File Type of the attachment(s)?	student doc <input type="text"/>
Comments?	<input type="text"/>
<input type="button" value="Submit"/>	

Figure 6: Searching window

Results of your Search

Searching for Sender: // Recipients: // Subject: *student* Content: // Attachment Info: *student doc*

Please select the attachment you want to download from the list:

Attachment(s)	Message Description
StudentDetails.doc [36.2 kb]	Date: <i>Sep 01, 2005</i> From: <i>testuser1@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Details of 3rd Year Students</i> [More>>]
StudentResults.doc [30 kb]	Date: <i>Sep 01, 2005</i> From: <i>testuser2@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Results of Assignment 1</i> [More>>]
GraphicsPaper.doc [225.6 kb]	Date: <i>Sep 04, 2005</i> From: <i>testuser3@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Results of Graphics Paper</i> [More>>]
Results.xls [340.2 kb]	Date: <i>Sep 05, 2005</i> From: <i>testuser2@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Survey Results</i> [More>>]
Results.xls [569 kb]	Date: <i>Sep 02, 2005</i> From: <i>testuser1@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Tender Evaluation</i> [More>>]
Paper.doc [102 kb]	Date: <i>Sep 02, 2005</i> From: <i>testuser3@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Fw: Philosophy Of Cognitive Science</i> [More>>]
Paper.doc [42 kb]	Date: <i>Sep 01, 2005</i> From: <i>testuser1@ucsc.cmb.ac.lk</i> To: <i>sec@ucsc.cmb.ac.lk</i> Subject: <i>Submission Deadline</i> [More>>]

Figure 7: Results window

The downloading of an attachment is a click on the relevant URL.

Hi,

Enjoy this. We are putting our full efforts into production.. These IT professionals are doing like that.

Click the following link to download the document,

<http://203.143.6.118/download.jsp?docid=20050920152039.M16903>

Regards,

Figure 8: A message with inline URL to attachment

Query Processor: The queries issued by the modules *Service Interface for Users* and *Service Interface for Attachment Referencing* are processed by this module. This module is responsible for analyzing the keywords provided by the user to identify relative importance of them, connecting to both *Message Registry* and *Attachment Repository* to retrieve required information,

filtering them to obtain the most appropriate results set and presenting them in an orderly manner.

6. Conclusion

This paper has presented a simple solution to increase the manageability of attachments by developing a globally accessible attachment repository for emails with searching, organizing and other features. Moreover it has presented an implementation architecture for a prototype system. It can be anticipated that in the future these kinds of supplementary systems would be useful to increase the usability of current email framework.

Some improvements to this proposed architecture is discussed under the future work.

7. Future Works

This attempt is only a first step to a more comprehensive solution. The framework could be improved by adding lot of additional features; some of them are identified as below:

- The current system does not search inside the documents which are stored in the *Attachment Repository*. By enabling a full-text searching functionality this could be incorporated into the system. In addition, the searching functionality could be enhanced by incorporating a good probabilistic searching method to get the most appropriate matches and order them accordingly.
- The alternative attachment referencing functionality could be enhanced to make the documents downloadable only to predefined locations by detecting the IP addresses of the locations. This way one could make a document downloadable only to a corporate network, so no outsider can see the document even the email message has arrived to one's mailbox accidentally.
- When the results of a search are displayed, they could be organized thematically, for instance, group the documents containing only jokes.
- There are situations where an email message represents an official letter. Then it is necessary to treat the body of that email message as a document and store it under the document repository.
- The security by means of encryption, and virus protection to the documents in the *Attachment Repository* could be incorporated to provide a more trusted service to the users.

References:

- [1] "S/MIME Version 3 Message Specification", Request for Comments, [Online] Available at <ftp://ftp.ietf.org/rfc/rfc2633.txt>
- [2] Karin Becker, Simone Nunes Ferreira, "Virtual Folders: Database Support for Electronic Messages Classification" Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications (CODAS), Kyoto, Japan December, 1996
- [3] Hiran Ekanayake, Kasun De Zoysa, Rasika Dayarathna, "A Notarization Authority for the Next Generation of E-Mail Systems", Proceedings of the International Information Technology Conference, December 2003
- [4] "ReadNotify Email Notary and Timestamping Service", [Online] Available at <http://www.readnotify.com>
- [5] "Webmail Tutorial", [Online] Available at <https://ucsc.cmb.ac.lk/data/openwebmail/help/en/index.html>
- [6] Yahoo! Desktop Search, [Online] Available at <http://desktop.yahoo.com>
- [7] H.E.M.H.B. Ekanayake, "Notarization Authority for Emails", B.Sc. Dissertation, University of Colombo School of Computing, May 2003.