A Secured and Semantically Addressed E-Mail Service

ETENG, Idongesit E. 1, 2 OSOFISAN, Adenike O. 3 AMADI, J.C. 1, 3
1, 3 Department of Mathematics/Statistics & Computer Science University of Calabar, Calabar, Nigeria
2 Department of Computer Science University of Ibadan, Nigeria
1 idongesitessien@yahoo.com, 2 mamoshof@yahoo.co.uk

ABSTRACT

This paper illustrates the use of semantic addresses to send emails in a corporate environment. Semantic Email Addressing (SEA) allows emails to be sent to a semantically specified recipient or group of recipients, which may be dynamically changing over time. In order to describe the concept of SEA and its advantages, it is compared with the concept of mailing lists. Two solutions to send semantically addressed emails are presented, on one hand a client sided solution has been elaborated where the email is directly sent by the user, on the other hand, a server sided solution is also discussed, where the server sends the emails to the recipient. To make semantic addressing possible attributes which define users who want to use semantic addressing have to be collected. These attributes can be introduced by either the administrator or the user. The attributes must be up to date; that means regular updates must be carried out. A novel update technique which uses the concept of aging is described. Also, finger print authentication was introduced to curb the security problem that characterizes SEA. Unified Modeling Language (UML) tools are used for the system design. The Java programing language was used to implement the prototype application developed to illustrate the concepts described in this work and several tests were successfully carried out on the developed application.

Keywords: Semantic Email, database update, biometric authentication.

1. INTRODUCTION

Group communication is becoming more and more important, either for private issues or for business matters. It is used to exchange cogitations, feelings, knowledge and information between at least two persons. Communication in groups happens every day between friends, family members, staff in a corporate office and even among presidents of countries. Friends can discuss about the newest and most affordable services on the Internet, family members may communicate one another about the date for family meetings etc. As seen in the examples, to establish a group communication, the following two points must be fulfilled:

• A communication channel must be defined.
• The recipients must be identified.

Defining a communication channel through which the communication should take place is not as easy as it seems and requires the use of existing channels. One of the most common communication channels is the electronic mail (email). Using this medium, people have the ability to exchange information with one specific person or with a whole group of people. The communication to one person is simple; it is nearly like a phone call to a person. The person who wants to establish the communication via the communication channel has to identify the receiving person. The identification via the communication channel using a telephone is done by entering the phone number. With email, the identification process is done by entering the email address.

In a group where the sending party knows everybody (knows all the email addresses to communicate), it is not a big problem to identify the persons. But if the members in the group change very often, it can happen that the sending person does not establish the communication channel to everybody and then information can be lost or not properly distributed to all relevant recipients.

To simplify group communication via email, the concept of mailing lists was invented.

With this concept, the group size does not play a role, the sending party does not have to know all the email addresses of the recipients, and fluctuation in the group is not a problem. A basic limitation however is that if a user wants to take part in a communication process, he has to know of the existence of the mailing list and has to sign up in it.

Mailing lists are used to send messages to predefined groups of people. However, as there are infinitely many ways to define a set of people, for example “all people in the marketing department whose name starts with the letter ‘A’”, one cannot in general rely on such predefined lists. Instead, one must have the ability to address or send emails to arbitrary groups of people. It has also been noticed that the current tools for communication across highly dynamic groups of people are non-optimal. This is because one should be able to send emails to groups of people matching a particular set of attributes such as all departmental heads in the University of Ibadan, or all
supermarket owners living in Cross River State, or all people in Nigeria who speak both English and French and most current tools lack this capability. The answer to this challenge is semantic email addressing.

Semantic email addressing (SEA) is a simple but novel technology that lets its users address emails to a semantically defined set of entities. The recipients to a semantically addressed email are computed on the fly, based on the semantic definition of the address [1] Imagine being able to send and receive broadcast email messages with an email system that incorporates semantic queries as part of an email client application.

All of these can be achieved through semantic email addressing.

1.1 Problem Definition

The primary supporting means for managing the processes surrounding sending messages to groups of people as mentioned above are mailing lists which are usually manually administered. In particular, the management of the groups and respective lists are overly complex. Identification of people that have to be addressed is time consuming and error-prone. An additional problem that somewhat complicates the issue of email addressing is that of security. Care should be taken to ensure that the senders of the mails are authentic users of the system.

Semantic email addressing has been proposed in this paper as a solution to the outlined problem. Semantic email addresses are logical (declarative) descriptions of recipients rather than static lists of strings. They allow sending emails with the same description to highly dynamic groups of people, and at the same time ensuring that the "right" people, at any specific moment in time are addressed. Semantic email addresses are stable, whereas the group might change very frequently. To tackle the second problem mentioned, the use of fingerprints as a form of authentication has been proposed.

1.2 Objective of Study

The objective of this paper is to demonstrate the concept of writing emails to a person or group of persons, without the knowledge of his or their email address or addresses as the case may be. Instead of using email addresses, semantic attributes are declared and employed. This is achieved by inserting relevant attributes like name, address, or company name that a group of persons are working for (in order to send an email to all members of the company) etc. In this paper, an email addressing interface will be used to build a list of email recipients on the fly, without necessarily needing to know a single name. Instead, all an email user would need to do is fill in the parts of the form using the drop-down boxes to guide their selections along the way. The system would then take care of the rest as will be demonstrated in this paper. The paper will also address the issue of security by building in a fingerprint identification mechanism.

1.3 Research Justification

Email addresses, like telephone numbers, are opaque identifiers. They are often hard to remember, and worse still, people’s email addresses and phone numbers change from time to time.

Email addresses are a means to an end. One’s goal is usually not to send an email to a particular address, but instead to a particular person. Ideally, one should be able to send a message to a person just by entering his name, his position, or some other descriptive attributes. If the email address of a person changes, then the email system should send to the new email address, automatically. If the person matching a description differs over time, the email system should always send to the person currently matching that description. This paper presents a system (semantic email addressing system), that will help its users route email to the correct person or persons without needing to know their email addresses and without the need for preexisting distribution groups. This approach will solve the problem of needing to search for the email address of a person or group of persons before sending emails to them, or receiving an email with the message that the inserted email address does not exist anymore.

1.4 Scope and Limitation

Since semantic email addressing has applications both in the corporate world and on the Internet, the implementation of this research work is based on a corporate world, considering a moderate size firm, where users will have to be enrolled manually in the database of the firm.

2. LITERATURE REVIEW

2.1 Problems of Mailing List

A mailing list offers a closed group of people the possibility to exchange messages concerning issues in which all participants are interested in. They are used in business, in research and other settings where group communication takes place. The sending party, with the exception of the administrator, does not know who is subscribed in the mailing list. He gives his piece of advice or asks questions that will be discussed in the mailing list. He sends his message to an allocator who sends each subscribed person the message. The administrator of the list can limit one’s rights to “only allowed reading”, “only allowed writing” or both rights.
When users are subscribed in many mailing lists, it can be very exhausting to maintain their appearances in the lists. Another disadvantage is that users, who are not interested in the topic, after some time, will have to unsubscribe, to become deleted from the list. The next problem can arise, when users change their email address where they want to receive the messages. They would have to unsubscribe the old email address from the list and subscribe with the new address, or they will have to update their profile in all the mailing lists.

Finally, it is not possible for a user to send a message to a particular subset of a mailing list because, the message will be sent to an allocator who sends the message to all subscribed users. This is another reason to think about the concept of semantic email addressing.

2.2 Semantic Email Addressing

The word Email refers to a message in the form of an electronic letter. Semantic refers to the concept of meaning and discusses how one can deduce simple terms from complex terms, to understand the meaning of the complex construction.

Semantic email addressing (SEA) therefore, can be understood as a mailing list, with the difference that the user does not need to sign in and sign out of this list. With semantic email addressing, the user can avoid the maintenance of his attendance at mailing lists described using his email address. Semantic email technology allows the senders of email to specify groups of people, as the intended recipients. One will be able to send email not just to atomic email recipients as we do today, but to those recipients who have a common set of characteristics.

According to MIT’s Technology Review, the program allows users to select email recipients by creating a search query as opposed to typing in names, addresses, or the name of a mailing list. For example, a user could use SEAmail to send a message to a group - created on the fly - like "all professors who graduated from the University of Calabar since 1980". Being able to pinpoint a recipient in this way would not only be helpful, it could also stem the overflow of email that creates information overload, by making sure that only the exact recipients who need to get the message are contacted. This implies that, because of its targeted nature, SEA could help combat unintentional spam and preserve the privacy of email addresses and even individual identities.

In order for a semantically-based email system to work, there needs to be a rich database that contains relevant information about the people sending email to each other and their interests. There are two ways a user’s attribute can be enrolled in the database. One of such ways is the enrolment by the user himself, and the second is automatic enrolment of the user. In most of the semantically-based email systems, a form which is responsible for the storage of the data in the database is implemented. An important difference to mailing lists is that the user have to enroll once into a single database, storing his personal profile, and not multiple times, when he submits to multiple mailing lists located at different places on the web. For automatic enrollment in the database, any source of information concerning the user has to be found. A possible way to find such information is to search for FOAF (Friend of a Friend) files in the Internet.

FOAF is a project that is used to generate a social network, which a machine can read. An FOAF file is a document where a person can store his information like name, email address, homepage and friends. The file can link to other FOAF files and so a network arises. [2]. That means that a machine can be programmed to read such files and find out vital information of people. Through the links to other files, a machine can crawl over all the files which are linked and store the information that it finds about a person, in the database.

2.3 Writing Semantic Emails

To write an email which will be addressed semantically is not more difficult than writing a “normal” addressed email. The user does not notice a difference when he writes on one hand, a “normal” addressed email and on the other hand, a semantically addressed email.

With the email client, the user can write semantic and “normal” emails without any difference. When he writes the email, he has to decide if he wants to address the email semantically or “normally”. So, there is no difference between both concepts concerning the writing of emails.

Today, the user writes his email message in a window and inserts the email addresses of the email recipients. The next step is that, the user has to send the email. This happens using, in the most popular email clients, the Send button. To make the transmission of semiantically addressed emails as easy as the nowadays email addressing; there should not be a big difference between both email transmission routines. After the user writes the message, as mentioned above, he has to insert the email addresses of the recipients, but with semantic email addressing, the user has to insert a dynamic address to make the transmission possible. Two possible ways to send a semantically addressed email are discoursed below.

a. Sending With The Client-Sided Solution

Popular email clients can be extended to be able to handle the semantically addressed emails as well as support the normal addressed emails. The semantic email client looks like the normal email client with the exception of the semantic send button. To send normally addressed emails, the user has to insert the email address in the To: field. If the
user wants to address another user semantically, he has to input the information which identifies the receiver.

To find out the attributes of the inserted text, search filter can be used. The search filters defines the attributes to be searched, and on the other hand defines what the input string stands for. The user can search for the email address related to his inputted text. After the definition of the attributes, the email address/addresses of the receiving party has/have to be found in a database that stores the users and their specific attributes. When a user with the defined attributes is found in the database, the email address is listed in the To field and the sender can send the email. Of course this solution is very helpful to send emails to people whose email addresses are not known by the sending user. But for security reasons, it would be safer not to send back the receiver’s email addresses to the client, so that the sender does not get to know the real addresses of the recipients and to avoid receiving a lot of spam. To achieve a safer solution, a second possibility is presented below.

b. Sending With the Server-Sided Solution

Another solution which would be more intuitive is the server-sided solution. With this solution, the user can send emails which are addressed emails in the Semantic To: text field. To insert the semantic email addresses, the user has to click into the Semantic To: text field. After the click, the same search filter as in the client-sided solution appears where the user can define the attributes that define the recipients. Unlike the client-sided solution, the server-sided solution receives a possible semantic address email from the email client, which includes a string from the receivers’ attributes and the message of the email. When the server receives the string, it suspends it and searches for the email address of the receiving party in the database. When an email address is found, the server sends the email to this address. If the email was not sent successfully, the server informs the client. In all these possible solutions, the email address of a user has to be found with his describing attributes.

c. Receiving Semantically Addressed Emails

The user starts the receiving routine when he uses the receive button of his email client. After the usage of the receive button, the in-box server of the user becomes synchronized with the local in-box of the client and the emails are listed there. To make semantic addressing suitable for daily use, the email client has to receive semantically addressed emails as well as “normal” addressed emails.

d. Replying To A Semantically Addressed Emails

Generally, the reply to emails is not a big problem. If a user receives an email, he opens it and can reply to the sender of the email. The recipient only has to use the Reply button in his email client. The email client of the receiver will then read the email address of the sender in the email header. After this step, the email address of the user who sent the email is automatically inserted in the To field of the replying email. The subject and the original message are also inserted in the mail with special headers, so that the user who will get the reply will know that the message was replied. If an email was addressed semantically, there will not be a difference about the single reply to the sender of the email.

e. Replying to all recipients of a semantically addressed email

Replying to all users who received the “normal” addressed email is not very difficult. The email client of the recipient has to parse all email addresses which are defined in the To and the Cc field of the email. When the recipient uses the Reply button, all the addresses which were written in these fields will be inserted in the To field of the “replying” email. After this step the replying user inserts his email and sends the email to all users.

Replying to all recipients of a semantically addressed email is a little bit more difficult. If a person receives a semantically addressed email, he also has to know if there are other persons who also received the email, without the knowledge of the real email address. If a user sends a semantically addressed email, he has to define the vital attributes in a search filter. To make it possible for the receiving part to reply the email to all receivers, the “replier” has to know which attributes the sender defined and the type of email addressing (semantic or “normal”) used.

Two possible ways to know the attributes defined and identified in the addressing type used are

1. signature and
2. attachment

Signature:

Today, signatures are very popular and widely used. The user can define within a signature, attributes which he always wants to send with the email. For example, for business purpose, the user inserts his telephone number, fax number, address of the company, and some other for him, vital attributes.

Signatures could also be used to send the attributes which define all the recipients. The email client of the sender adds a unique signature to the message, which the sender does not see. Only the receivers of the email see this signature. If a receiver wants to reply to all other receivers, he has to click the reply button of the received email. The client looks for the signature in the email and parses it to
gain the attributes which the sender also defined in his email. So the replying receiver automatically defines the vital attributes to generate the semantic email address.

Attachment

Another possibility is to send an added attachment with the email to all receivers. In this attachment, the vital attributes of all the recipients could be listed. An advantage of this approach is that the email does not have to be changed; only an attachment has to be added. A big disadvantage is that, some email clients can block attachments and so the user will not have the possibility to reply to the email. Another disadvantage of using attachments is that an attachment is not always trusted because it can contain a Virus, or it can be exchanged through another attachment where the receiver’s attributes are completely deferent from those who actually should receive the email.

2.4 Forwarding Semantically Addressed Emails

The forwarding process of semantically addressed emails is not very different from the nowadays email forwarding routine. If a user wants to forward an email, he has to use the forward button of his email client. Afterwards, the subject field is extended with information that the email is forwarded, for example with an FWD character sequence, followed by the original subject. The original message of the email will also be inserted in the message body. The user only has to insert the email addresses of the recipients to whom the email should be forwarded.

If a user wants to forward a semantically addressed email, the forwarding routine is completely the same but here he to click the Semantic To field, whereby the semantic search filter appears where the user defines the attributes of the forwarding email recipients. Instead of the forwarding button in the most common email clients, the recipient can use the Semantic To field to input the new attributes, which will define the user or group of people who should receive the forwarded email.

2.5 Advantages of Semantic Email Addressing

According to [2], some advantages of semantic email addressing include the fact that no discovery and maintenance are required.

No discovery required: Email addresses and mailing lists can be difficult to discover, even for a person. It is even more difficult or almost impossible for a computer or a web service to discover an email address or a mailing list. However, with Semantic Email Addressing, discovery is not required.

No maintenance required: SEA makes maintaining information about email addresses of different users not just possible, but easy.

As opposed to SEA, traditional mailing lists require manual labor to maintain. A mailing list administrator must go through the process of creating the list. After the list has been created, it must then be maintained by the administrator himself and the individuals who would like to subscribe to the list, unsubscribe to the list, and change their information with regards to the list. This can even be a more complicated issue when several lists are involved. With SEA updates and changes are done easily.

Recipients Computed on the Fly: With Semantic Email Addressing (SEA), the sender of the email does not need to store the email accounts of the recipients, this of course minimizes memory overheads as email addresses are computed on the fly based on the needs of the recipients.

2.6 Problems of Semantic Email Addressing

Semantic Email Addressing raises some important issues, both for its use in a corporate environment and on the open Internet. Some of these issues according to [2] include:

a. Security and Privacy

SEA is ideal for targeted email addressing but, it can also be abused. An individual or company can easily use SEA to send exceedingly untargeted emails. With a small sized community, it is easier to deal with members that violate privacy rules. Within larger communities like the Internet, the communication links are more and thus rules can be easily broken. Care therefore must be taken by the administrator to ensure that mails are sent by the right people to the right people too. In this work, a combination of passwords and fingerprint authentication are used to ensure that mails are sent by authentic users.

b. Dealing with Errors

One of SEA’s most powerful aspects is that the set of recipients changes over time as the people and properties in the database change. Ideally, the database is always up to date, reflecting the world’s changing realities. In practice, information in the database is not always 100 percent accurate. Inaccurate information can lead to people receiving email when they shouldn’t and not receiving email when they should. System users should be able to ensure that their information is correct, either by editing it themselves or by having an authorized administrator change the information for them [2]. In addition to this, the administrator should maintain a log of all information that should be changed and the degree of urgency of the change. See section 3.2 for more details on the handling of updates.
c. Standardization

SEA does not require that all implementations use the same addressing scheme. The scheme can vary from organization to organization or even from server to server within an organization. However, it is expected that one or more standards will emerge over time. With standardization, email clients could provide better built-in support for SEA, including SEA address editing, analysis, and search [2].

2.7 Authentication Technologies

Authentication is the process of verifying the identity of a user or a computing device in a networked environment [3]. It follows that authentication technologies refer to the application of science in authentication. Authentication methods are divided into three main groups according to [4] which include tokens, passwords and biometrics.

a. Tokens (what you have)

When using tokens for identification, the user must possess a physical and portable device which contains the user's identity. Examples of such devices include bank cards, drivers' license, passports, office ID cards, Students ID cards etc. On the point of identification and authentication, the user must present the required token.

b. Passwords (what you know)

These methods are based on a user's knowledge characterized by its secrecy. Examples include passwords, PINs, username, etc. Care must be taken however to keep written down passwords secret. Also passwords and user names must be changed periodically to guard their secrecy. It should also be noted that over time hackers have broken into passwords, therefore, it is recommended in this work that passwords must be combined with other authentication techniques to in order to make their strength “fireproof”.

c. Biometrics (what you are)

The user submits to the system his physical and/or behavioral characteristics. As a result, the individual is either accepted as a valid user or is rejected.

Furthermore, an automated biometric system may operate in two different modes: recognition and identification. In the former mode, the system authenticates a claimed identity by comparing an input biometric characteristic with its theoretical corresponding template. In the latter, the system receives a claimed identity and searches therefore an entire database for a match.

2.8 Finger-Print Identification

A fingerprint is the pattern of ridges that appear on the surface of the fingertip [4]. It is perhaps the most popular and reliable biometric characteristic used for human authentication. These observations rely basically on two of its properties: individuality and persistence. Individuality refers to the fact that the fingerprint is unique across individuals and across fingers of the same individual. Persistence means that the basic fingerprint characters do not change over time in spite of age or ill health.

The fingerprint structure comprises of two levels of features: Ridges and valleys. These patterns are typically used for fingerprint classification purposes and they do not own any kind of properties for establishing the identity of an individual. In a typical fingerprint image, the dark lines are related to the ridge patterns, whereas the brighter ones are referred to as the valleys. This is the global view. At the local level, the fingerprint features are known as minutiae. They correspond to the points where the ridge patterns either bifurcates (ridge bifurcation) or terminates abruptly (ridge endings). These features possess the discriminating information of establishing the individuality of fingerprints and therefore the identity of an individual.

In this work, fingerprint authentication (coupled with passwords are used in order to ensure that mails are sent by the right people.

2.9 Related Work

As summarized by [5], an Electronic mail (Email) is a natural and perhaps inevitable use of networked communication technology that developed along with the evolution of the Internet. Indeed, message exchange in one form or another has existed from the early days of timesharing computers. Network capable email was developed for the ARPANET shortly after its creation, and has now evolved into the powerful email technology that is the most widely used application on the Internet today.

The notion of Semantic Email was originally introduced by [6] to refer to an email message consisting of a structured query coupled with corresponding explanatory text, based on a number of Semantic Email Processes that represent commonly occurring workflows within email. Several researchers have recognized the value of bringing semantics to email. For example, the Information Lens System [7] lets users send semi structured email messages, and filters those messages using production rules. Users can send to a special mailbox called “anyone”, and anyone can choose to receive messages from this mailbox based on production rules. Instead of starting with receiving all emails and writing them down based on filtering rules, the user starts with an empty inbox and pulls in email of interest [7]. More recently, Mails More lets users annotate an email’s
content with Resource Description Framework (RDF) triples, and automatically includes RDF triples based on standard email headers such as the “To,” “From,” “Subject,” and body fields. This can be used for semantic filtering and filing of emails [8]. The Mangrove system takes this idea further. It allows not only structured email content but also semantic email processes [9]. Users can script email clients with declarative workflows that automatically aggregate information obtained from many email responses, automatically resend emails to people who have not responded, or analyzed the semantic content of incoming email messages and respond accordingly.

[10] Lets administrators create query-based distribution groups, which are essentially mailing lists whose recipients are based on a Lightweight Directory Access Protocol (LDAP) query, run when the email is sent. This alleviates much of the administrative work required to maintain a mailing list. However, because only an administrator can create the mailing lists, users cannot send SEA mail, and the information upon which the lists are based is not under users’ control. None of this application’s functionality is available to the users and very little to the administrators. In fact, users cannot see that a distribution group is query based; each query based distribution group has a name, so to an outsider it looks like a regular mailing list.

Our work is similar to that of Kassoff et al (2009) from Stanford University in several ways that will be outlined shortly. In their work, a prototype SEA module is referred to as the Info master Semantic Email Addresser (ISEA) implemented. This runs on top of the Info master information integration engine. The Info master alloys querying multiple data sources on the Internet through a single mediated schema. The system can therefore pull information from many sources not just about people but also useful supporting information about organizations, locations, and so forth. The prototype lets members email people based on their site, group affiliations, name, interests, and other attributes. Information needed are obtained from private databases and publicly available FOAF files. An administrator sets the ontology SEA uses, so it’s centrally controlled but is continually evolving.

In our work, SEA is done through a simple matching of semantic addresses with real addresses stored in the database. Semantic addresses are computed on the fly. A mail sender now clicks the semantic send button and the mails are sent to the right recipients. Security measures are also built in using fingerprint identification.

3. SYSTEM DESIGN METHODOLOGY

This section deals on the system design methodology. It describes the system requirement specification, use case, activity diagram, flowchart, program specification etc. The database design is also presented here.

3.1 System Requirement Specification

This design is based on a moderate – Size Company with several departments. The company has a centralized database containing the following information about its personnel: first_name, last_name, office_number, nickname, department, department_head, email_address, fingerprint and password.

The company also has the following information about its projects: project_id, project_name, project_startdate, project_enddate, project_leader, proj_location. Project team members also have the following information: project_name, project_head, first_name, last_name, email_address. With these information, a user can define precise groups of people such as all developers of a particular project, for example “send mails to all members handling the health project in Ibadan supervised by prof Wale”. Using SEA, a user can send email to a group or a subset of a group. A recipient of a semantically addressed mail can do same or reply the sender of the mail.

To semantically specify a set of mails, a user needs three somewhat separable pieces of functionality in this application:

1. The user must be able to define the group of email addresses of interest. This the user can do by picking characteristics/features that are common to this group of recipients. In the case raised above, project leader = “Prof Wale”, proj_Location = “Ibadan” etc.
2. This definition by the user must somehow be translated to a set of email addresses to which the email is sent. In our work, after a recipient has selected semantic attributes, simple matching techniques are used to filter out the needed recipients’ email addresses from the database
3. The SEA application must facilitate reply in a simple way.

This email client requires only a small extension over the traditional email client. This involves adding an extra button that lets a user specify a semantic email address. When the user presses the button, a window pops up with an interface for defining a set of recipient. The user is given the opportunity to confirm the mails, displaying the names of the persons satisfying the user’s criteria.

This system has three basic components:

1. A standalone user interface for constructing a semantic email address. This is illustrated in fig 4.1.
2. A database for determining the traditional email addresses of the people matching a query. A view of the database is displayed in figs. 4.7, 4.8, 4.9 and 4.10.
3. An application code to glue the first two together.

Access to this system is heavily restricted. A user will need to be password and fingerprint authenticated to gain access to the system.

### 3.2 Database Update Technique

Two methods for updating the database used for semantic email addressing have been earlier mentioned to include update by the administrator or update by the individual. The novel update method proposed in this paper can be employed for any of the two mentioned approaches.

A special update database is created to store the data that need to be updated. The adjunct database, table 3.1 is linked to the main database. Items for update such as last name (where a female involved in a project is married), email address, proj_leader, proj_location etc. are stored in this special update database. NOTE that this can be decided by the administrator based on the information in the database. The frequency of update is included in another field. For instance, when the duration of projects are known and project start date, project end date are also known. For cases where the frequency of the update is unknown, the figure (date or time) can be set by the administrator. As soon as the update in this case has been done, a reset for subsequent updates should be also programmed. Also, an alert system is built into the system that reminds either the administrator or the user when the update is necessary.

The update database itself needs to be updated, therefore, records with the entry done in the remarks column should be deleted based on age (how long the update task has been carried out).

With this technique, one is sure that the database will always be kept up to date.

**Table 3.1: Database Update Table**

<table>
<thead>
<tr>
<th>Update Data</th>
<th>Frequency</th>
<th>Alert</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proj _Leader</td>
<td>Annually</td>
<td>reminder</td>
<td>done</td>
</tr>
<tr>
<td>Proj _location</td>
<td>When project changes</td>
<td>reminder</td>
<td>Do after change of project</td>
</tr>
<tr>
<td>Project startdate,</td>
<td>quarterly</td>
<td>Send alert</td>
<td>done</td>
</tr>
<tr>
<td>Project enddate</td>
<td>nil</td>
<td>nil</td>
<td>nil</td>
</tr>
</tbody>
</table>

### 3.3 Use Case

The use cases here in fig 3.1 include the sign up for new address, change user information, verify identity, login and send mail use cases.

![Fig 3.1 Use Case Diagram](image-url)

The sign up for new address use case includes the verify identity use case to verify the identity of the user. The collected information is stored in the database. The change user information use case allows the user to change part or all of his/her information. This use case includes the verify identity use case. Since such an action on information is a restricted right of the owner, the user will need to identify him/herself, this will involve a password authentication.
3.4 Activity Diagram

Fig 3.2 shows the activity diagram.

3.5 Flowchart

The flowcharts are shown in figures 3.3 and 3.4.
3.6 Object/Class Diagram

The class diagram is shown in fig 3.5. Each mail belongs to a user and each user can have zero or more mails. Each user belongs to zero or more projects and project consists of one or more users.

3.7 System Control

Access to this system is controlled using password and fingerprint. During sign up, the data of the user are collected via a graphical user interface (GUI). The user’s input is validated to ensure correctness and consistency. If the user’s information passes this test, the user’s mail account is created.

During sign in, the user is requested to enter his/her email address and fingerprint to gain access to the system. If the email address and fingerprint is correct, the user gains access.

When the user is already in the system and needs to perform sensitive operations like deleting of mails and editing of user data/information, he/she is requested to enter a password. This is another security feature.
4. SYSTEM IMPLEMENTATION

4.1 Introduction

In this section, an implementation based on the design presented in section 3 above is illustrated. The reasons for the choice of implementation language along with the system requirement of the target computer system and software maintenance issue are all presented here.

4.2 Features and choice of implementation language

The language of choice here is java. The database server used for storing data is MYSQL. The fingerprint standard development kit (SDK) used in the fingerprint authentication process is Griaule fingerprint SDK from Griaule biometrics. MYSQL connector/J (JDBC) is the connector that serves as the bridge between the interfaces and the database.

a. Java

The graphical user interface (GUI) was designed using Java.

Figures 4.1 through 4.5 show the interfaces.

4.3 MySQL

MySQL is used as the database server for the system Figures 4.6 through 4.11 show the tables in the database.
4.4 Griaule Fingerprint SDK

This is a fingerprint SDK from Griaule biometric. It provides the class libraries that can be used in developing fingerprint related application in java and other programming languages.

Griaule fingerprint SDK is easy to use and very compatible with java programming language.

This application supports the following fingerprint scanners;
- Microsoft fingerprint reader
- Microsoft wireless intelligent mouse explorer with fingerprint reader.
- Microsoft optical desktop with fingerprint reader.
- Digital persons U.Arc.U 4000
- Digital persons U.Arc. U 4000B etc.

4.5 My SQL Connector/J

This is a driver that connects applications developed in java to the MYSQL database server.

It provides an easy path for the MYSQL server. This connector is open source and is free.

Programming with this connector is purely with JAVA and SQL.

4.6 System Testing Strategies

Module testing: The different modules were tested individually for performance accuracy. The modules were tested with the expected functions that they are meant to perform. These tests were successful.

Unit testing: Every module was tested in relation to other related modules. The module is supposed to work together, linking each other to achieve the purpose of the application. These tests were successful.
Integration testing: The entire system was then tested to ensure efficiency and correctness. These tests were also successful.

5. SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Introduction

From the forgoing, it is obvious that semantic email addressing is a very useful tool in addressing many people dynamically. This advantage also raises the issue of the applicability of SEA on large communities like the Internet where spamming is recurrent. Everyone has to deal with the daily never ending spam messages sent to mailboxes on a daily basis. With a tool like the semantic email client, spammers would have a very potent tool to over-flood people with spam. Hence, spam filters will have to be implemented in the email client of semantic email users, or users who want to use this technology can define an attribute which allows only known users to address them semantically. All of these have been achieved in this work.

5.2 Summary

In this paper, the importance of communication among several parties has been established. Electronic mails (E-mails have been identified as a popular service for sending mails from users to recipients. This history of emails has been briefly given. The shortcomings of mailing lists which have been used for sending messages to a set of common recipients have been discussed. The concept of Semantic Email Addressing has been proposed over the traditional mailing lists. A simple but novel methodology for semantic email addressing that incorporates fingerprint authentication and password at several levels of the work is proposed. A novel aging technique for updating the database is proposed. Unified Modeling Language (UML) design tools are used for the design of the system. The prototype system was implemented using the java programming language.

5.3 Conclusion

In this paper, the notion of semantic email addressing was introduced. The targeted nature of a semantically addressed email is powerful and can be used to combat unintentional spam. Furthermore, since it can facilitate the contacting of individuals based on their characteristics, it can be used to preserve the privacy of email addresses and even individual identities. SEA is a simple concept, and has to be incorporated into commercial email systems.

5.4 Recommendation

As mentioned earlier in this section, an interesting possibility to help control spam is semantic filtering and filing of emails where one could write semantic email rules based on not just the standard email fields, but also based on the semantic email address.

Currently, a very practical domain of application of semantic email addressing is in business settings. The dynamic concept of this technology can be fully utilized in such a setting because of the fluctuation of employees, the alternation of employees to other departments and so on.

Another area of applicability for SEA is in e-commerce, where e-mails need to be sent to specific users and these users change very often.

Another domain where the application can be used is where customer relationship concepts take part. For example in a hotel, where guests receive newsletters, offers and other mails.

How to generate user interfaces that allow these sorts of filters to be specified, and how to efficiently filter large numbers of emails against a semantic email address are subjects for further research.

5.5 Further Work

SEA is a simple concept that has the potential to be easily implemented, leveraging existing semantics such as FOAF, and significantly improving the functionality of email systems. As part of further work, the concepts of SEA, database updates and fingerprint authentication can be incorporated into commercial email systems.

Standardizations in terms of ontologies for Semantic Emails should also be researched. Additionally, we intend to develop the application developed in this work as a web service not just as an application. This will enable us to fully make comparisons with our existing mailing list web service [11].

REFERENCES


