TOWARDS THE DEFINITION OF A QUALITY MODEL FOR MAIL SERVERS

Juan Pablo Carvallo, Xavier Franch
Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3 (Campus Nord, C6) E-08034 Barcelona (Catalunya, Spain)
{Carvallo, Franch}@lsi.upc.es

Abstract. The paper presents an approach for building a Mail Server Quality Model, based on the ISO/IEC software quality standard. We start by defining the mail system domain to be used as general framework and the relevant technologies involved. Then a general overview of the ISO/IEC standard is given. The basic steps, the relevant considerations and criteria used to select the appropriated subcharacteristics and quality attributes are also presented. The selected attributes are categorized under the six ISO/IEC quality characteristics conforming the model. Finally some case studies requirements and two commercial mail server tools are used to evaluate the model.

1. INTRODUCTION

Internet applications have changed the way in which modern organizations work. Some of them have base their business model on applications such as the Web and mail systems. Mail services are growing in importance, every day more companies become interested in using them to improve inside and outside communication and coordination. Because of this growing popularity, an overwhelming number of mail related products are currently available in the market and organizations face the problem of choosing among them the ones that best fit their needs. Core components of mail systems are mail servers, therefore, successful mail service deployment depends on their correct selection and configuration. For these reasons, having a good quality model for this domain can be considered especially useful.

The Internet Mail Consortium (IMC) is an international organization that has been constituted primarily by internet mail hardware and software vendors, to cooperatively promote the expansion and use of Internet mail. But even with the existence of this organization (which some of its goals are the selection of protocols and the definition of common standards) it has not been defined a common guide to select mail system components.

It is the purpose of this work to propose a Quality Model that may be used as base for selecting mail server products.

In section 2 of this paper a reference framework for mail systems is defined and some important technologies and components involved are reviewed. In section 3, we explain the main steps that we have followed to build the model as well as some considerations and criteria that we have used to select quality attributes and subattributes. Next selected attributes are categorized under the six ISO/IEC 9126 software quality characteristics. Finally in section 4 two real case studies and two mail server products are used to evaluate the model.

2. THE DOMAIN FRAMEWORK

It has been mentioned that mail systems are one of the most popular internet applications. Some of the Internet basic building blocks are the TCP/IP protocol and its services. Mail systems are not an exception and they may also be defined as a TCP/IP-based client-server architecture.

It is not the purpose of this section (or this report) to deeply explain either the internet infrastructure or the technologies involved. Our purpose in this section is to provide the reader with a general introduction and overview of some of the key concepts involved in the mailing process. Issues such as the basic mailing architecture, protocols and security will only be addressed in an informative way, as parts of a bigger picture, that will be later used as a framework reference.

2.1. The Mailing Architecture.

The Internet Mail Consortium (IMC) [L1] describes the basic client-server mailing architecture (Figure 2.1) as the process of *relaying* mail from an *originator mail user agent* (MUA), to a *recipient mail user agent* through one (or various) *mail transfer agents* (MTA).

The originator MUA submits mail to a MTA who may then relay it to other MTA (or possible many of them). When mail arrives to the destination, the final MTA *delivers* the message to the appropriated mail *message store* (MS), from where can be *accessed* by the recipient MUA.

In practice MTA are software packages installed and running over a single mail server computer or groups of them (*mail server cluster*). Similarly, MUA are software packages known as *mail clients* running over the user local machine.

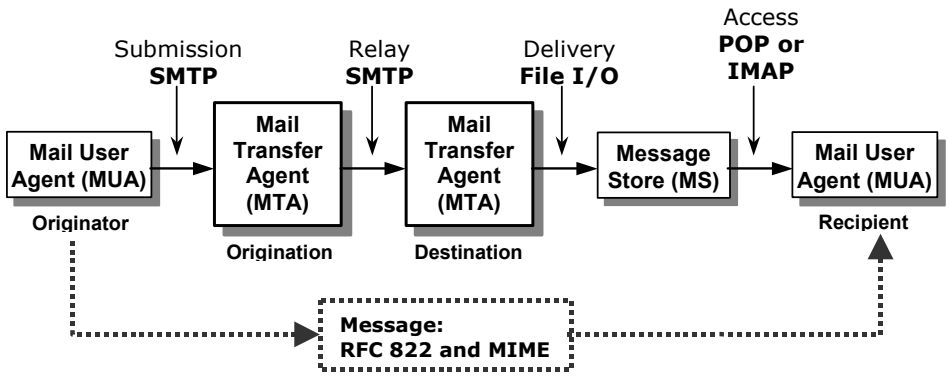

FIGURE 2.1: IMC basic mail architecture and standard protocols.

According to the IMC standard, the submission and relaying processes are achieved by means of the *Simple Mail Transfer Protocol* (SMTP)[1]. The access process on the other hand is accomplished using either the *Post Office Protocol* (POP) or *the Internet Message Access Protocol* (IMAP).

[1] For a more detailed description of TCP/IP related protocols please refer to [1].

2.2. Protocols overview.

SMTP mail messages are composed by a header (or envelope) and a content or body part. The header contains multiple "`<field name> : <field value>`" lines, each one with a specific meaning and use. The most commonly used fields are:

- `TO:` Receiver destination address.
- `CC:` One or many destination addresses to where user wants to send exact copies of the message.
- `From:` Sender information.
- `Reply To:` Mail Box address from where sender wants to recover replied messages.
- `Return Path:` Route back to the sender.
- `Subject:` Short description or summary of message.

SMTP was originally designed to handle text messages in simple and pure 7-bits US-ASCII. Therefore non-text content, such as multimedia files (audio or images) or even messages whose languages required richer character sets, where not supported.

The Multipurpose Internet Mail Extensions (MIME), is an standard that includes mechanisms to solve these problems. In simple words MIME is the encoding scheme used by SMTP-based systems to do attachments of non-7bit textual contents.

MIME defines protocols to include objects (other than ASCII text messages) within message bodies. It also includes additional header fields to specify the MIME version, content type and how objects in the body are encoded. Seven content types where originally defined: Five discrete (text, image, audio, video and application) and two composed (multipart and message). The MIME standard also covers issues such as message fragmentation and reassembly, and external body part subtypes.

Some alternative protocols to SMTP have been proposed, but probably the only one with some degree of acceptance especially in Europe and Canada has been the ISO X.400.

X.400 was designed to become the new standard for electronic information transfer. It included many features not originally supported in SMTP, which were not foreseen when it was conceived.

The X.400 set of standards includes multimedia and multi-language file support, electronic routing control, security, world wide naming structure and integration to X.500 directory services. At the end those last two characteristics turned out to become the source of some of the problems associated to this protocol. Address names were too long and complex for typing because they were though to be stored in a X.500 global name directory, and also organizations did not want their employees' e-mail addresses to be available to the general public.

Mail servers may be accessed either in the local intranet or remotely through the internet. They may also provide service to users in more than one network domain, and these domains can be defined inside or outside an organization. In all those cases they relay messages using SMTP over the TCP/IP protocol. X.400 is build over the OSI layering standard thus is not directly compatible with SMTP. Connectors must be used to bridge communication between them.

The use of POP (currently version 3) or IMAP (currently version 4) protocols, to access and recover messages from message stores (mail boxes), is highly related to the way in which *message access paradigms* [4] operate. The three known message access paradigms, *Online*, *Offline* and *Disconnected* are defined as:

- Offline. The MUA reads the messages from the MS and copies them to the local machine. After this operation messages are deleted from the server, and they can be treated locally.

- Online. All the messages are kept on server and are treated remotely from the Mail Client application.

- Disconnected. In this approach, messages are left on the server, but a copy of them is made in the local machine. After that, they can be locally accessed, even if connection to the server is lost.

Both protocols IMAP and POP support offline operation. But unlike IMAP, POP does not support neither online nor disconnected operation. The IMAP capability to operate in all three modes is its main advantage over POP. Some other IMAP advantages are:

- Capability to manipulate multiple mailboxes.
- Remote folder management (list/create/delete/rename).
- Support for folder hierarchies.
- Support for message status flags.
- Capability for accessing non-email data; e.g.: NetNews, documents.
- Provision for determining message structure without downloading the entire message.
- Selective fetching of individual MIME body parts.
- Server-based searching and selection to minimize data transfer.
- Global availability of messages.

But there are also some advantages of POP over IMAP that should be mentioned:

- POP protocol is much simpler.
- POP has more software available.
- POP requires a minimal time of connection (just enough to recover messages from server).
- POP minimizes the use of server resources.

2.3. Directory servers

Messages contained in MSs, as well as contact information and resources such as printers an peripherals, must be organized in some way in order to be easily accessed by MUAs. The X.500 standard defines an specification for a rich, world-wide, distributed directory, based on hierarchically-named information objects, that users can browse and search using arbitrary fields.

X.500 specifies that communication between the client and the server directory uses the *Directory Access Protocol* DAP. However, DAP requires the OSI protocol stack to operate, making it directly incompatible with TCP/IP based protocols like SMTP, IMAP y POP. Other problem faced is that supporting the entire OSI protocol stack requires more resources than those available in some environments.

The *Lightweight Directory Access Protocol* (LDAP) was developed as a lighter alternative to DAP. Based on the TCP/IP protocol stack, LDAP provides on interface to X.500 directory servers and simplifies some of its operations.

LDAP defines only a communication protocol to access data into an X.500 directory. It does not defines the directory service itself (Figure 2.2). The LDAP client and the X.500 directory use different communication protocols (TCP/IP vs. OSI). For this reason the LDAP actually

communicates to a gateway process known as an LDAP server, that translates from TCP/IP to OSI and the opposite way.

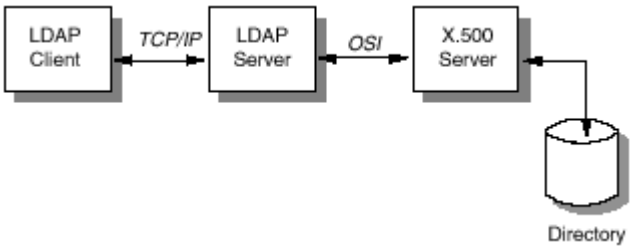

Figure 2.2: LDAP access to X.500 Directory service.

It was later introduced and implemented, the idea of LDAP servers providing access to local directories, supporting the X.500 model, rather than acting as a gateway to them (Figure 2.3). The name LDAP server is used either for servers that implement gateways to X.500 directory servers or for those that access local directories.

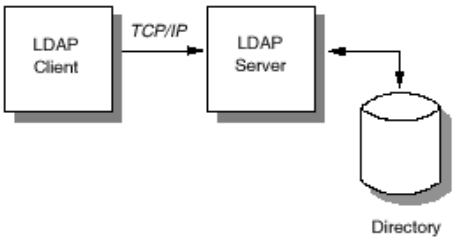

Figure 2.3: LDAP Directory service.

2.4 Mail clients.

As mentioned in 2.1 Mail Clients are software packages, installed and running on local machines. They can be classified in 3 groups:

- E-Mail program clients: Users submitting or accessing mail with specialized mail software applications such as MS Outlook, Eudora or the Lotus Domino Client.

- Web Browser clients: Users submitting or accessing mail by using Web browsers like MS Internet Explorer or Netscape Navigator. Those users are sometimes referred as *web-mail users*.

- Mobile devices clients: Users that access or submit mail, using pieces of software (mostly proprietary), included in devices such as PDA's or WAP phones.

Applications like MS Outlook, Eudora or the Lotus Domino Client are specifically designed and built to be used as mail clients. Therefore they support standard protocols like SMTP, POP or IMAP, and are directly compatible with mail servers. Mail client software applications are rich in tools and resources to manage messages. They can be used to edit messages, maintain address books and perform actions like sending, reading, replying, forwarding, attaching or storing messages among other interesting features.

Web browsers communicate to servers using the *Hyper Text Transfer Protocol* (HTTP). HTTP is the protocol used to transfer *Hyper Text Markup Language* (HTML) documents. *Web Mail Servers* are used to translate user HTTP requests back and forth to SMTP, IMAP or POP protocols so messages in Mail Servers can be accessed.

Web mail servers should also format in HTML messages recovered from messages stores, so they can be displayed through web browsers. In order to do so, they may execute CGI scripts written in Perl, PHP or other programming languages. Web Mail Servers are composed by a HTTP Server, CGIs and a Web Mail Application (used to translate the protocols) (Figure 2.4).

Some modern mail servers include native HTTP support, making it unnecessary to use specialized web mail packages.

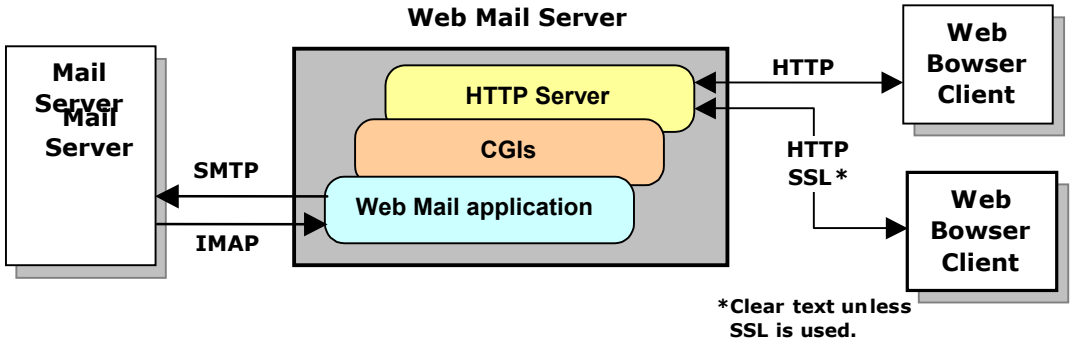

Figure 2.4: Web Mail Server Architecture.

The *Wireless Application Protocol* (WAP) is the facto standard used to provide internet communications and applications to mobile devices. Similarly to Web Mail clients, WAP mobile devices require of gateways (sometimes called *WAP Servers*) to access the mail servers.

2.5. Server Clusters

A group of local Mail Server Computers, working together in a synchronized manner and sharing their resources, is known as a *Mail Server Cluster*.

From the mail server software point of view, mail server clusters may be classified as Active/Active (A/A) or Active/Passive (A/P) [2]. In A/A clusters the mail software (or some of its components) runs in all the servers of the cluster at a time. In A/P clusters, the application runs in only one of the servers. Having a cluster of servers instead of a single server, may be worthy depending on the capabilities and the design limitations of the mail server software. Mail server clusters provide many advantages they may [2] [3]:

- Eliminate single points of failure.
- Reduce downtime for planned outages such as routine maintenance, configurations changes, and hardware or software upgrades.
- Increase disaster recovery capabilities.
- Allow workload balancing.
- Allow replication and synchronization among servers in the cluster.
- Improve scalability.

The use of server clusters also have some disadvantages: They increase the initial deployment cost and add a significant degree of complexity to environment, making configuration and maintenance tasks more difficult.

Some clusters share one common MS for all the servers. In this case some of the advantages such as the elimination of single point of failure, replication and synchronization or reduction of downtime, not longer exist or are dramatically reduced.

2.6. Security

Secure data exchange is one of the goals of communication systems. Cryptographic techniques are used by most of the modern ones, to grant security. The following definition of cryptography[2] was obtained from [5]:

> *"Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication.*
>
> *Cryptography is not the only means of providing information security, but rather one set of techniques."*

The main idea behind cryptographic techniques is to input clear text messages into a cryptographic algorithm (cipher), that returns them as an unreadable output. This process is known as *encryption,* and its inverse as *decryption*.

Security provides confidentiality through encryption, but it must also provide authentication, data integrity and non-repudiation.

- *Authentication* means verifying that the sender of a message is really who claims to be.
- *Data Integrity* makes reference to the process of verifying that a message has not been altered along the communication path.
- *Non-repudiation*[3] is the possibility to prove without chance of denial, that a message has been sent by who claims to be the sender.

It is difficult to keep cryptographic algorithms secret, because they are exposed and used by many people. For this reason current algorithms are "keyed", which means that security relays entirely in a key, and not in the operations that the algorithm performs.

Cryptographic keyed algorithms can be classified as *secret key* or *symmetric* algorithms and *public key* or *asymmetric* algorithms.

In symmetric algorithms the encryption and the decryption key are the same. It means that sender and receiver must agree on the key, before starting a secure communication. They can be classified in *block algorithms* if they operate over blocks of bits of the original message, and *stream algorithms* when they do it over single bits (or bytes).

One of the problems commonly associated with symmetric key algorithms, is that a secure channel must be provided in order to exchange the key (figure 2.5).

---

[2] For a introduction to cryptographic techniques and TCP/IP security please refer to [1]; more detailed references to cryptography may be found in [5].

[3] Defined in the mailing domain.

Well-Known symmetric block algorithms are the *Data Encryption Standard* (DES), *Triple-Data Encryption Standard* (3DES) or the *International Data Encryption Algorithm* (IDEA), which is used in the *Pretty Good Privacy* (PGP) certification system. An example of stream algorithm is A5. A5 is used in the mobile telephony standard GSM.

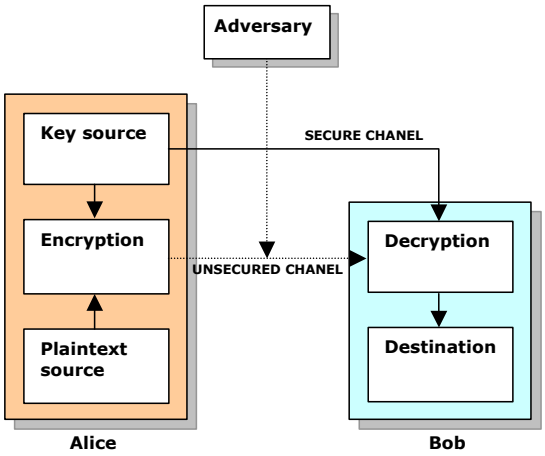

Figure 2.5: Symmetric key schema[4].

Asymmetric algorithms (figure 2.6) use two keys, one public and one private. The private key cannot be obtained from the public one. Clear text encrypted with the public key can only be decrypted with the corresponding private key. Similarly, clear text encrypted with the private key can only be decrypted with its corresponding public key. Users recover each other public key before starting the communication, and then exchange packages encrypted with them.

It is obvious that those systems also provide authentication and non-repudiation. If a public key can decrypt a message, then the message must be originated by the owner of the corresponding private key, he or she gets authenticated and may not denied.

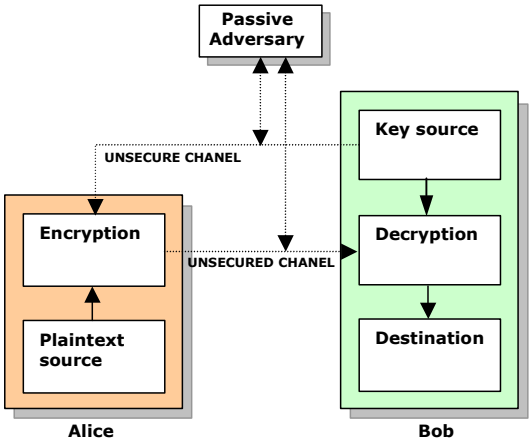

Figure 2.6: Asymmetric key schema.

[4] *Alice* and *Bob* are commonly used in cryptographic literature, to denote the participants of a communication protocol. The *Adversary* represent a security tread.

One example of public key algorithm is the Ron Rivest, Adi Shamir and Leonard Adleman *RSA algorithm* which relies in the difficulty of factoring to large numbers. The public and private keys are a function of two large prime numbers. Other example is the *Diffie-Hellma*n algorithm.

Asymmetric key algorithms face the *man-in-the-middle* attack (Figure 2.7). An adversary (C) sends a public key that Alice (A) assumes is the public key of Bob (B). *A* encrypts messages with that key. *C* intercepts messages from *A* to *B* decrypts them with its own private key, re-encrypts them with *B* public key and sent them to *B*. In this case nether *A* nor *B* are aware that *C* is accessing their messages.

For this reason public keys must be validated for authentication and no repudiation. This validation is obtained using *digital certificates* which are files that bind an identity to the associated public key. The digital certificate is signed with the private key of the a *certification authority* (CA), so it can be authenticated. The ISO X.509 is the international standard for digital certificates.

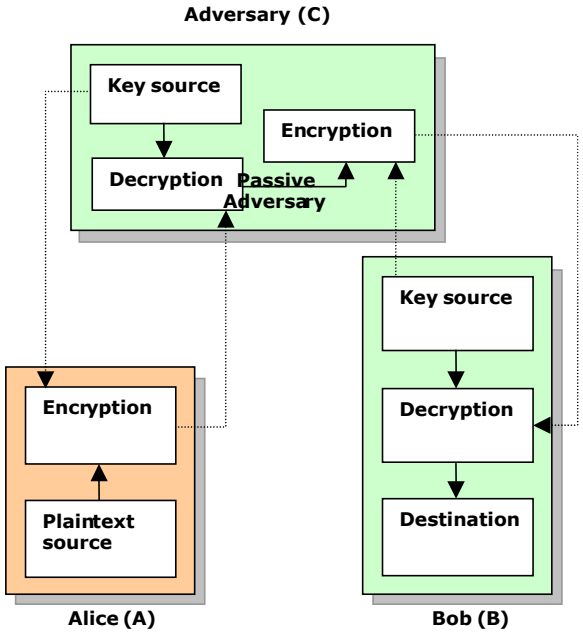

Figure 2.7: Impersonating (man in the middle) attack.

Additionally to key algorithms, cryptography uses *hash functions* to assure integrity and authentication. A *hash function* takes variable-length input data and produces fixed length output data (the *hash value*), which can be seen as the "fingerprint" of the input. That is, if the hash values of two messages match, it is highly probable that the messages are the same.

A hash function that takes a key as a second input parameter, so its output depends on both the message and the key, is called a *message authentication code (MAC)*. If a secret key is added to a message and their concatenation is input to a hash function, the result is a MAC.

The encryption of a hash value with the private key is called a *digital signature*. The encryption of a private key with a public key is call *digital envelope*. Digital envelops are used to distribute secrete keys for symmetric algorithms.

Two of the standard protocols implemented for secure internet communications are SSL and S/MIME. The *Secure Sockets Layer Protocol* (SSL) is a protocol that provides a secure alternative to the standard TCP/IP socket. SSL is composed of two layers with protocols that support a variety of encryption algorithms, and protocols to support initial authentications and transfer of encryption keys. The main purpose of SSL is to provide privacy, integrity and authentication. SSL is not intended to be exclusive to TCP/IP but it is widely implemented for those connections.

Secure Multipurpose Internet Mail Extension (S-MIME) is a SSL-like protocol with uses limited to mail protection. It uses X.509 certificates to validate end-point entities at application level.

2.7. Application binding

Some applications that are usually related to mail servers[5] are *network news* and *mailing or discussion lists*. Network News, is based on the *Network News Transfer Protocol* (NNTP). Users can view categorized news covering different topics, contribute to the news groups that maintain them, or receive periodic mails with complete or abbreviated news, including the appropriated self reference links.

In discussion lists, mails submitted to the list home address are stored in shared mail boxes. From there they can be accessed by all the members of the list. Users may subscribe or unsubscribe to mailing lists, either on demand, by sending a mail to the list administrator, or automatically, sending messages containing single words like “subscribe” or “unsubscribe” to an specific list address. If automatic subscription is not supported by the mail server, additional list server software (for example *Majordomo*) must be used.

As the number of internet applications increases, mail servers tend to bind some of them. Even if applications are not directly related to mail, they may become target of this practice. Real time applications such as white boarding, chat, instant messaging, voice or video conferencing and application and file sharing are fully supported by some mail server products. New protocols like the *Real-Time Transport Protocol* (RTP) and *RTP Control Protocol* (RTPC) have been developed to support those application.

2.8. The general picture

Most of the topics that have been introduced in this section are represented on figure 2.8. Some possible mail scenarios are described in a graphical way, including the main components and the protocols involved.

Organizations may have single or multiple mail servers, or mail servers clusters. Servers may be accessed by members of one or many network domains, defined in the local intranet or remotely through internet. Home users access mail servers usually through Dial-Up telephone connections, despite the availability of new mechanisms, such as cable or satellite connections.

As mentioned in 2.4 mail users have a choice of client applications. Web browsers and mail client applications, may be used alternatively or simultaneously, to access mail accounts. Other users may use mobile devices, ranging from notebook computers to PDA and WAP devices, to access mail. In all the cases messages may include text and multimedia files as well.

There can be local and remote News and List groups, to which, home and organization users, may subscribe.

---

5 For a more detailed description of TCP/IP related applications refer to [1]

Servers and clients may exchange mail by using SMTP, IMAP, POP, HTTP or X.400 among other protocols, depending on there location, network configuration and applications used.

Message stores may be accessed directly or trough LDAP or X.500 directory servers.

Finally some organizations or individual users may need to interact using secure connections based in X.509, SSL or S/MIME.

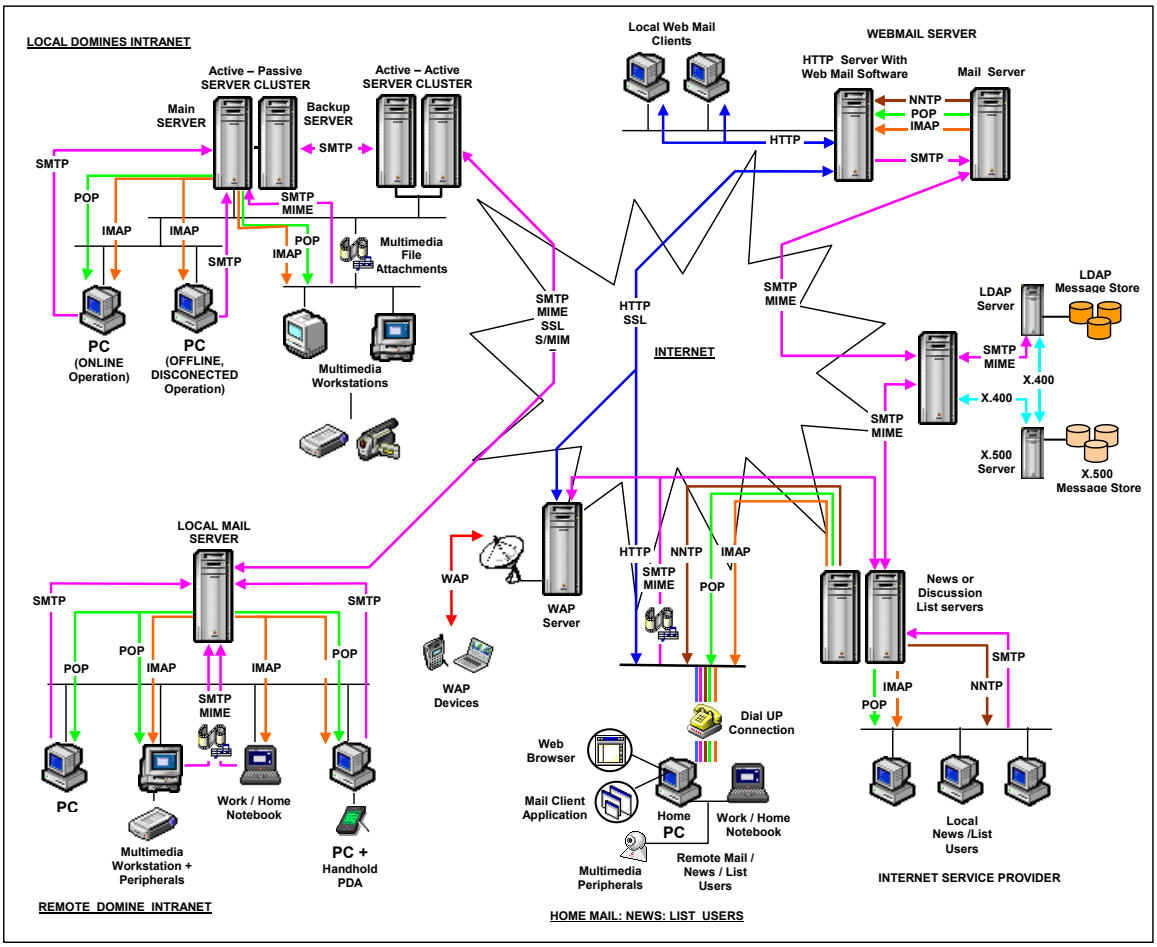

Figure 2.8: The mailing scenario.

3. THE QUALITY MODEL

3.1. The ISO/IEC 9126 Quality Model Standard.

One of the objectives of this work is to identify and propose a set of attributes that may be used to match companies mail requirements with mail servers characteristics. The attributes must be outlined in some sort of hierarchy to simplify this process. *Software Quality Models* have been proposed to help with attributes classification and to deal with this kind of characteristic-requirement matching problem.

The ISO/IEC 9126 [6] set of standards defines a Software Quality Model applicable to every kind of software. It does not define the attributes (because they are specific to each context), but defines a set of six software quality characteristics, under which they can be classified. The characteristics are defined as:

- *Functionality: A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.*
- *Reliability is the set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.*
- *Usability is the set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.*
- *Efficiency is the set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.*
- *Maintainability is the set of attributes that bear on the effort needed to make specified modifications.*
- *Portability is the set of attributes that bear on the ability of software to be transferred from one environment.*

The standard also suggest a group of non-mandatory subcharacteristics. User may eliminate some of them or add new ones, depending of their specific contextual needs.

We will use the ISO/IEC quality model as a framework because it is a publicly-available international open standard, that give us enough flexibility to choose the specific subcharacteristic and attributes to suite our needs.

The ISO/IEC subcharacteristics and their definitions are:

CHARACTERISTICS	SUBCHARACTERISTICS	DEFINITIONS
FUNCTIONALITY	SUITABILITY	Attributes of software that bear on the presence and appropriateness of a set of functions for specified tasks.
	ACCURATENESS	Attributes of software that bear on the provision of right or agreed results or effects.
	INTEROPERABILITY	Attributes of software that bear on its ability to interact with specified systems.
	COMPLIANCE	Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.
	SECURITY	Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs or data.

RELIABILITY	**MATURITY**	Attributes of software that bear on the frequency of failure by faults in the software.
	FAULT TOLERANCE	Attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface.
	RECOVERABILITY	Attributes of software that bear on the capability to reestablish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.
USABILITY	**UNDERSTANDABILITY**	Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.
	LEARNABILITY	Attributes of software that bear on the users effort for learning its application.
	OPERABILITY	Attributes of software that bear on the users effort for operation and operation control.
EFFICIENCY	**TIME BEHAVIOUR**	Attributes of software that bear on response and processing times and on throughput rates in performances its function.
	RESOURCE BEHAVIOR	Attributes of software that bear on the amount of resource used and the duration of such use in performing its function.
MAINTAINABILITY	**ANALYZABILITY**	Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.
	CHANGEABILITY	Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.
	STABILITY	Attributes of software that bear on the risk of unexpected effect of modifications.
	TESTABILITY	Attributes of software that bear on the effort needed for validating the modified software.
PORTABILITY	**ADAPTABILITY**	Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.
	INSTALLABILITY	Attributes of software that bear on the effort needed to install the software in a specified environment.
	CONFORMANCE	Attributes of software that make the software adhere to standards or conventions relating to portability.
	REPLACEABILITY	Attributes of software that bear on opportunity and effort using it in the place of specified other software in the environment of that software.

3.2. Applying the ISO/IEC 9126 to the mail server domain

The initial quality subcharacteristics

As it was mention the ISO/IEC standard suggest a set of subcharacteristics for each of the characteristic of the model. They are not mandatory but we feel that they are reasonable enough to be used as an stating point in our study and so we adopted them without modifications.

Refining the hierarchy of subcharacteristics

The initial set of subcharacteristics may require some refinement once the domain is analyzed with some detainment. For example in the mail server domaine, the ISO/IEC *suitability* subcharacteristic, has been divided into *Mail Server Suitability* and *Additional Suitability*. This decision was taken because many of the commercial mail severs (as it was mentioned in 2.7) tend to bind applications that were not originally related to them. Those applications are not usually shipped within the original packages. They are offered separately, as extensions of the original one. But we found that in many cases, they are referenced as a constitutive part of the functionality of a single product. Many companies may be interested in using them, and so we though that it was important to list them as attributes of a functionality subcharacteristic.

The attributes categorized under the *Usability* / O*perability* subcharacteristic may be seen from two different points of view: general users and administrators. For this reason at the beginning we where tempted to divide this subcharacteristic into two. At the end, we decided that general user operability on mail servers depends on the mail client and the privileges given by the administrator. We were not able to clearly see attributes related to clients that were independent of those related to administrators, and so we decided to keep only one subcharacteristic.

Selecting the quality attributes

The abstract subcharacteristics should be further decomposed into a set of more concrete quality attributes, which may lead to a more particular evaluation of the observable features of a software package in the domain.

It may not be possible to list all the quality attributes related to mail servers, but it is certainly possible to create a very complete list of the most relevant ones. This is particularly true, if one considers that product manufacturers try to include some characteristics that make their products different to the others. This is also one of the reasons why most of the quality models, such as the ISO/IEC, are open.

There are many commercial mail server packages available in the market, but it is very difficult to find complete, independent and reliable information of them. Manufacturers tend to give just a partial view of their products. Either they put so much emphasis on their product benefits, without mentioning the weakness, or they give a partial look of the truth, making them capable of more features of which they really cover.

On the other hand, there are some third-party reports that look very independent, but they have been strongly refuted for technical departments of parties involved, making them difficult to rely on. Other non-commercial articles compare mail server features, but they base their reports on evaluators knowledge of the tools, and their particular taste, more than in serious technical tests.

Because of these problems, we decided to go for a more abstract approach in determining the quality attributes. We decided to make the selection relying on the concepts that were shown (and there possible benefits), instead of the evaluation that manufacturers (or their competitors) give of them. In other words, we looked for all the great ideas, regardless of the product, the platform or the way and extent, in which features were implemented. At this point we were looking for a qualitative list of attributes in stead of a quantitative one.

Another problem we encountered was related to semantics. Functionalities in different products may have different names, or even if they have the same name, they may perform very different tasks. This makes the identification of characteristics that are common to many products very difficult.

Once we started categorizing attributes, it became obvious that some of them were suited for more than one characteristic. For instance, *Message Tracking and Monitoring* may be seen as a functional attribute that grants accurateness, or else as a analyzability attribute of the maintenance characteristic. another example is the *clustering support* attribute. Depending on the software package, it can be seen as an attribute that enhance efficiency or as one that improves fault tolerance (making the system more recoverable).

Decomposing derived attributes into basic ones

Some quality attributes may not be directly measurable and will require to be further decomposed in order to be evaluated. This was taken in account and it is important to notice that in our classification attributes may contain sub-attributes. For example, the attribute *Functionality / Interoperability / Choice of clients* includes the subatributes *E-mail program clients*, *Web Browser clients* and *Mobile devices clients*, which represent the different kinds of mail clients that were mentioned in 2.4. It was also possible to list them as separated attributes, but we felt that the subattribute approach is more structured, and the ISO/IEC model does not forbids their use.

The following is the complete list of attributes that we have identified in our research:

CHARACTERISTIC : FUNCTIONALITY

Subcharacteristic: MAIL SERVER SUITABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Folders		Attributes related to management of local and remote folders.
	1 Default Folders	Set of default folders provided (usually Inbox, Outbox, Sent Folder, Draft Folder and Trash Folder).
	2 Folders and subfolders management	Management of user defined folders (other than those provided by default).
	3 Integrated access and management of remote folders	Management of remote, default or user defined folders.
2 Message Sending and Receiving		Attributes related to message exchange.
	1 Send and receive plain text messages	From peer to peer plain text message exchange.
	2 Send and receive ASCII attachments	7 Bit ASCII or rich character set file attachments.
	3 Send and receive multimedia attachments	Multimedia attachments such as audio, video or picture files.
	4 Send and receive RTF, HTML formatted mail.	Possibility to format messages in HTML or RTF, and to display messages recovered in those formats.
	5 Send Messages to distribution lists	TO: field including a list destination address in stead of a personal one.
	6 Send and receive Encrypted Messages	Support of encryption algorithms to ensure message confidentiality.
	7 Send and receive Authenticated Messages	Support of mechanisms to authenticate messages originators.
	8 Rules and filters for incoming mail	Possibility to apply rules and filters to incoming messages e.g.: - to move incoming mail to folders depending on sender. - to delete messages with specific addresses or contents. - to deny exchange of messages larger than a predefined size etc.
	9 cc recipient list	Carbon Copy message to addresses listed in field.
	10 bcc recipients	Sent Blind Courtesy Copies of message to addresses listed in field.
3 Message Handling		Attributes related to message management
	1 Message creation and management options	Message edition.
	2 Grammatical tools	Tools to correct messages grammar and spelling.
	3 Reply to messages	Reply to message originators directly from received messages.
	4 Message Forwarding	Sent received messages to address other than return one.
	5 Message redirection	Change destination addresses of messages received in one account.
	6 Automatic message redirection	Same than message redirection but automatically
	7 Status marks	Possibility to apply status flags to messages. Used to inform of things such as priority of messages, if messages have been read or redirected, etc.
	8 Message Sorting	Classify and sort messages according to different attributes, e.g. sender, date of arrive, subject, etc.

4 Address Book		Attributes related to Address Book management
	1 Electronic address and contact information	Management of contact directories that include contact information such as electronic addresses, telephone numbers etc.
	2 Distribution Lists Management	Possibility to create, modify or delete, lists of destination addresses to be treaded as a single one. Messages sent to this address are forwarded to all the directions associated to the list.
	3 Personal distribution lists	Same than distribution list bud managed for each individual user.
	4 Nested distribution lists	Management of hierarchical distribution lists (lists including other lists or individual users).
	5 Definition of groups and friends	Management and administration of contacts as categorized groups.
5 Calendaring & Scheduling		Attributes related to Calendaring and scheduling information
	1 Appointments and remainders registration	Personal or group online digital scheduler.
	2 Check appointments for groups and friends	Possibility to interact with other users or group schedulers. (See if the have planed appointments).
	3 Meetings scheduling and invitation	Possibility to interact with other users or group schedulers. (add appointments).
	4 Task and Notes	To do tasks registration and management
6 News		Support for Usenet news users, news groups and news messages.

Subcharacteristic: ADDITIONAL SUITABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Collaborative applications.		Binded applications that allow workgroup collaboration
	1 Integrated document management	Possibility actively manage documents shared with other users.
	2 Workgroup-based project management	Tools to manage the entire life cycle of a project within a single environment.
	3 Web Workflow	Possibility to program mail workflow tracking process, such as document approval, purchase orders, etc.
	4 Discussion databases	Support for database storage of group messages and files.
2 Web based messaging.		Binded applications that allow real time communication.
	1 Chat	Used to build online communities of users interested in discussing similar topics or issues.
	2 Instant Messaging	capability to send an immediate, text-based message to another user on a computer network. Unlike e-mail messages, instant messages are posted immediately to the other user's screen
	3 Voice and Video conferencing	Real time multi-party audio and video conferencing
	4 Data Conferencing	The sharing of a program, such as Microsoft Word, participants have the ability to co-create documents in real-time.
	5 Whiteboarding	A multi-user drawing application that enables users to sketch diagrams or organization charts
	6 File Transfer	The sending of a file, in the background of the conference to another user.

Subcharacteristic: ACCURATENESS	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Message tracking and monitoring	tracking of messages across network domains. Users can check the status of their sent messages.
2 Message delivery notifications	Information automatically provided by server if delivery problems are found. (e.g.: Wrong destination address, servers down etc.)

Subcharacteristic: INTEROPERABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Choice of clients		Different Kinds of clients supported by server.
	1 E-Mail program clients	Users that connect to mail servers using mail client applications such as Lotus Notes, MS Outlook or Eudora.
	2 Web Browser clients	Users that connect to mail servers using web browser such as Internet Explorer or Netscape Navigator.
	3 Mobile devices clients	Users that connect to mail servers using mostly proprietary pieces of software, included in devices such as WAP Phones or PDAs.
2 Open Interfaces and connectors		Software packages, used as gateways between mail servers, and other specialized software components or applications.
	1 To Distributed objects	Software components that allow the interaction with distributed objects repositories. (e.g.: CORBA/IIOP, COM/DCOM, RMI)
	2 To DBMS	Software components that allow the interaction with database systems. (e.g.: ODBC, JDBC, OLE DB)
	3 To other mail servers.	Software components that allow the interaction with other mail servers.
	4 To structured information	Software components that allow the use of structured information. (e.g.: XML, SOAP)
	5 To applications	Software components that allow the interaction with specialized applications such as DreamWeaver, FrontPage, faxing services etc.
	6 Other Interfaces and connectors	Software components that allow the interaction with other systems not contemplated ins subattributes 1 to 5.

Subcharacteristic: COMPLIANCE		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Mail Transfer Protocols		Protocols used to send and relay mail (e.g.: SMTP, ESMTP, X400).
2 Message Access Protocols		Protocols used by clients to access mail in servers (e.g.: POP3, IMAP4).
3 Message Access Paradigm		How mail client access and interact with mail in servers (e.g.: Online, Offline and Disconnected paradigms).
4 Directory Services		Mechanisms and protocols supported to manage directory services. (e.g.: LDAP, X.500).
5 Web Protocols		Supported Web applications protocols (e.g.: HTTP, NNTP).
6 MIME		Support for Multipurpose Internet Mail Extensions.

Subcharacteristic: SECURITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Secure E-Mail Standard Protocols		Supported protocols for secure mail and mail attachments exchange (e.g.: SSL, S/MIME, APOP).
2 Certification System		Supported Certification Mechanisms
	1 Directory methods	Security standard that uses Public Key encryption and Certification Authorities for authentication (e.g.: X.509 and CA, PKIX)
	2 Referral methods	Shared Key certification standards, where users and key are referred from one user to the others, forming chains of authenticators (e.g.: PGP).
	3 Collaborative methods	Authentication at protocol level, needs to be completed with a higher layer authentication protocol. Uses chains of authenticators at bout ends. (e.g.: : SKIP)
3 Encryption Algorithm		Supported Encryption algorithms (e.g.: RSA, Diffie-Hellman, DES, 3DES, IDEA)
4 Login and password		Login control to accounts with user names and passwords authentication.
5 Execution control lists (ECL)		Lists of executable files allowed to run on server, specially useful to protect against virus executables.
6 Access Control Lists (ACL)		List of access privileges to files. They can be defined at, local user, group or rest of the world levels.
7 Trust Relationships		Network inter-domain level privileges, for interconnection and sharing of resources between different domain users.
8 Spammers Thwarting and Bulk-Junk mail handling		Politics, filters and rules to deal with spammers and unwanted-unauthorized bulk mail.

CHARACTERISTIC : RELIABILITY

Subcharacteristic: MATURITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Time of product on market, versions and updates.	For How long have the product been offered to users, which versions are currently available and how many upgrades of each release have been offered. This attribute may be very important in terms of knowing, up to what level manufacturer is committed with future development of the tool, and also how successful releases have been.
2 Product versions and updates	Number and characteristics of versions, presentations and updates of product.
3 Maturity of OS and Hardware platforms	How strong are the Operating system and hardware platforms, over which mail server is to be installed.
4 Percentage of availability	Percentage of time that mail server is expect to work without interruption. Some planned down time must be considered for maintenance, upgrades and reconfiguration.

Subcharacteristic: FAULT TOLERANCE	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Failover capabilities	Mechanisms that are provided to maintain availability and protect information, in the event of hardware or software failures, or low resources operation (e.g.: no enough space in message stores).
2 Clustering	Fail over capabilities related to advantages provided by clustering. (e.g. no single point of failure, alternative access to message stores etc.)
3 Database Replication	Online replication of message stores, between local or distributed servers, and the possibility to selective access them.

Subcharacteristic: RECOVERABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Full or selective replication and synchronization	Selective replication of entire messages stores and directories (users, folders, profiles, permissions, etc.) or parts of them.
2 Single Mailbox Backup and recovery	Possibility to backup individual mail boxes and to restore them without affecting overall performance.
3 Online Incremental Backup	Possibility to create backup copies without stopping services (maintaining availability).
4 Online Restore	Possibility to restore from backup copies without stopping services (maintaining availability).
5 Dynamic Log Rotation	Possibility to dynamically assign log management operations to servers in cluster.
6 Event Logging	Maintenance of log files with information of all system events triggered during operation.
7 Transaction Logging	Maintenance of log files with information of mail transactions executed during operation.

CHARACTERISTIC : USABILITY

Subcharacteristic: UNDERSTANDABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Interface Standards, and standardization	Standards used in user interface, is it textual o graphical. How well standardized area events and objects associated to them. This may include things such as: integration with operating system environment or reuse of icons and the events associated to them.
2 Well defined architecture	How recognizable and differentiable are application components. How intuitively are related to the set of actions that they perform.
3 Interface language	Languages supported for the interface.

Subcharacteristic: LEARNABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Documentation, user manuals and references.	Relevance of information provided by manufacturer. Is it complete and clear?, does it deeply explain features or only describe them?.
2 Tutorials	Are multimedia curses provided with software package or available online. Is training included in price.
3 On Line Help	Online local help.
4 Predictability	How intuitive are actions to be performed in relation to the options in software component. Is it easy for users to relate the interface icons, colors, dialogs etc. to the actions that they perform.
5 Vendors customers support	Do the company providing the software or their representatives have a customer support department?. If they do, how well prepared in use of the application are their technicians?.

Subcharacteristic: OPERABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Accounts Administration		
	1 Individual Users and groups	Maintenance of users and user groups.
	2 Private and public accounts	Maintenance of public and private accounts, associated to users, groups and distribution lists.
	3 Individual and shared Mailboxes	Maintenance of message stores and individual mailboxes, associated to individual and lists groups users.
	4 User Privileges	Maintenance of individual and group, access control lists.
	5 User Profiles	Definition of groups, privileges and resources, assigned to individual and group users.
	6 Directories and subdirectories	Management of directory services.
2 Resources Administration		
	1 Maximum storage time of mail messages	Limit of time that messages are to be kept in message stores, without downloading them.
	2 Maximum time of life for inactive accounts	Time that unused accounts will remain active, before deleting them.
	3 Mailbox quotes	Storage space assigned to individual mailboxes.
	4 Mail file sizes	Maximum size of messages or message attachments, that users may send.
	5 Management of Groups of Servers as a Single Entity	Logical grouping of mail servers, to manage their resources, as if they where part of a single one.
3 Message Delivery Administration		
	1 Maximum number of delivery retries for outgoing Messages	If delivery problems are found, haw many times the delivery process must be retried before canceling.
	2 Time between delivery retries	Waiting time before server try to resend messages.
	3 Mail delivery Priorities	Definition of priority rules for mail delivery. e.g.: messages to lists, should be sent after mail to individual users.
4 Services Administration		
	1 Distribution lists	Parameters related to distribution lists management.
	2 News groups	Parameters related to news groups administration.
	3 Automatic subscription	Configuration of parameters and rules, for automatic subscription and unsubscription, to distribution list or news groups.
5 Environment and interconnection Administration		
	1 Security Parameters	Configuration of parameters related to security, such as authentication mechanisms, and security protocols to be used.
	2 Protocols	Definition of protocols to be supported by server
	3 Login Mechanisms	Configuration of parameters used to create log files.
	4 Clustering and failover systems	Management of server clusters and related mechanisms such as replication, and A/A A/P components.
	5 Backup and recovery politics and systems	Definition of back up and disaster recovery politics.
	6 Connectors	Configuration of different connectors to be used by server.

6	Web Based Administration	Authorized administrators can perform tasks such as, users and groups management and messages monitoring, from anywhere using a Web browser.
7	Administrative tools and wizards	Set of utilities designed to automate configuration and some commonly performed tasks.

CHARACTERISTIC : EFFICIENCY

Subcharacteristic: TIME BEHAVIOR		
	ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1	Average response time	Amount of time required by server to detect and process new messages.
2	Message through output	Amount of time per unit of size required to send a message.
3	Load Balancing	Support for uniform distribution of workload between servers in cluster.
4	Multiprocess Support	Possibility to perform more than one process at a time. It improves performance for concurrent users.
5	Online Defragmentation and Space Recovery	Possibility to perform administrative tasks, such as message stores defragmentation and space recovery, without stopping services. Response time improves after process is concluded (during process it may be affected).
6	Routing control	Gives flexibility to increase performance and reduce transmission costs, thwart spammers, filter junk e-mail, and easily enforce quotas on message and mail file size
7	Clustering	Helps to increase workload balancing and multiprocess support depending on server capabilities.

Subcharacteristic: RESOURCE BEHAVIOR		
	ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1	Number of mailboxes per server	Maximum number of mailboxes that can be defined in a single server.
2	Number of Concurrent mail users per server	Maximum number of concurrent mail users accessing a single server.
3	Number of active webmail clients	Maximum number of concurrent Web-Mail users accessing a single server.
4	Management of quotas on message and mail file size	Management of storage space assigned to individual mailboxes, and maximum size of messages or message attachments. This makes possible to obtain the number of mailboxes to be supported by a server.
5	Message volume of their target customer	Mail clients differ in the size and volume of messages that they may handle (e.g.: WAP devices may manage short mostly textual messages, while web browsers may handle text, pictures or even video files)
6	Single Copy Store	Possibility to store a single copy of a messages to be recovered by many users. This is very useful in case of news or lists group messages.

CHARACTERISTIC : MAINTAINABILITY

Subcharacteristic: ANALYZABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Message tracking and monitoring	Tracking of messages across network domains. Users can check the status of their send messages.
2 Automated mail server usage reporting	Manage the messaging environment via direct statistical analysis of server performance and connectivity. For example, track the number of mail users versus HTTP users connected to a server.
3 Expert Analysis Tools	Analyze server functions over time, for performance tuning, capacity planning and trend prediction. Set and track service level agreements, correlate performance statistics and more.
4 Billing services.	Track, report and analyze system usage for billing, charge-back and capacity planning purposes.

Subcharacteristic: CHANGEABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Cross Domain Administration	Centrally administer all servers in the organization network domains. Change user names, add/delete users, move users, upgrade servers, etc.
2 Scalability as resources are increased	How possible is to add new users or mailboxes, and redefine parameters such as mailbox quotes, as hardware resources (e.g.: primary and secondary storage) are increased.
3 Automatic mail redirection for moved accounts	Allows for messages to be automatically redirected, when users mailboxes are moved between servers.

Subcharacteristic: TESTABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Message tracking and monitoring	tracking of messages across network domains. Users can check the status of their send messages.
2 Expert Analysis Tools	Analyze server functions over time, for performance tuning, capacity planning and trend prediction. Set and track service level agreements, correlate performance statistics and more.

CHARACTERISTIC : PORTABILITY

Subcharacteristic: ADAPTABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Supported Operating Systems		Choice of operating systems over which mail server may be installed and run.
2 Supported hardware platforms and architectures		Choice of hardware architectures over which mail servers may be installed and run.
3 Choice of clients		Different Kinds of clients supported by server.
	1 E-Mail program clients	Users that connect to mail servers using mail client applications such as Lotus Notes, MS Outlook or Eudora.
	2 Web Browser clients	Users that connect to mail servers using web browser such as Internet Explorer or Netscape Navigator.
	3 Mobile devices clients	Users that connect to mail servers using mostly proprietary pieces of software included in devices such as WAP Phones or PDAs.

Subcharacteristic: INSTALLABILITY	
ATTRIBUTE	**ADDITIONAL DESCRIPTION**
1 Administrative tools and wizards	Set of utilities to automate installation and configuration process
2 Development tools for collaborative applications	Possibility to programmatically change environment, or add new required mail-related features.
3 Views and forms editors	Possibility to create new mail forms or modify default ones, to enhance there functionality.
4 Documentation, user manuals and references.	Relevance of information provided by manufacturer. Are references complete and clear, do they deeply explore features or only describe them.

Subcharacteristic: CONFORMANCE		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Support for standard Interfaces and connectors		Software packages used as gateways between mail servers and other specialized software components or applications.
	1 To Distributed objects	Software components that allow the interaction with distributed objects repositories. (e.g.: CORBA/IIOP, COM/DCOM, RMI)
	2 To DBMS	Software components that allow the interaction with database systems. (e.g.: ODBC, JDBC, OLE DB)
	3 To other mail servers.	Software components that allow the interaction with other mail servers.
	4 To structured information	Software components that allow the use of structured information. (e.g.: XML, SOAP)
	5 To applications	Software components that allow the interaction with specialized applications such as DreamWeaver, FrontPage, faxing services etc.
	6 Other Interfaces and connectors	Software components that allow the interaction with other systems not contemplated ins subattributes 1 to 5.

Subcharacteristic: REPLACEABILITY		
ATTRIBUTE	**SUBATTRIBUTE**	**ADDITIONAL DESCRIPTION**
1 Server build in accounts and mailboxes migration tools		
	1 To/From other E-Mail Servers	Tools to migrate individual accounts or mailboxes from one mail server to another.
	2 To/From other OS	Tools to migrate individual accounts or mailboxes from one mail server to another mounted in a different operating system.

Determining metrics for attributes

Another problem to be addressed when defining the quality model is selecting the metric for each attribute. Some attributes can be evaluated by simple boolean values, either application complies with them or not. Other may be represented by atomic data types such as integer or float values of a particular unit (e.g.: the average response time in milliseconds, or the maximum account size in megabytes). A number of attributes require a more complex

representation such as fix or open sets (e.g.: the languages of the interface that are supported for a package, or the list of the default folders which it provides).

Metrics for some quality attributes may be very difficult to define. For example the rules and filters for incoming mail are defined by using logical expressions involving the AND, OR and NOT operators as well as string and other functions over several fields of messages. Value of attribute includes logical expressions that may be difficult to evaluate and may differ based on requirements and configuration. An alternative way to be used in those cases is to define the metric as a discrete set of values which are the result of a function that depends of other values that must be evaluated independently of the model.

The metric suggested for each of the identified attributes is shown in appendix A.

Stating relationships between quality attributes.

Some attributes imply the use of others. For example, if one Mail Server uses a *Certification System*, some *Encryption Algorithm* must also be used, because they are needed to grant confidentiality. Another example are the functional *Sent multimedia attachments* and the *MIME* attributes, when the SMTP *Message transfer protocol* is to be used.

This feature is particulary interesting because once relationships among attributes are identified they may be used to automatically extended requirements.

A tabular representation of the direct relationships that we have found is shown in tables 3.1 and 3.2. Attributes in rows contribute to attributes in columns, with either a positive (+) or a negative (-) partial support [6]. It is important to mention that some other direct or indirect relations between attributes may be found, but we are listing only the most relevant ones.

CHARACTERISTICS		Efficiency
	SUBCHARACTERISTICS	Time behaviour
	ATTRIBUTES	Average response time
Reliability	Recoverability	
	Full or selective replication and synchronization	-
	Single mailbox backup and recovery	-
	Online incremental backup	-
	Online restore	-
	Dynamic Log rotation	-
	Event Logging	-
	Transaction Logging	-
Efficiency	Resource behaviour	
	Number of concurrent mail users per server	-
	Number of active webmail clients	-
	Management of quotas on message and mail file size	+
	Message volume of their target customer	-
	Single copy store	+

Table 3.1: Related Efficiency – Reliability attributes.

ATTRIBUTES	Suitability											Interoperability	Compliance	Security	
	1**		2 Message Sending							6	1*	4	1	2
	1 Folders and subfolders creation and management	3 Integrated access and management of remote folders	1 Send and receive plain text messages	2 Send and receive ASCII attachments	3 Send and receive multimedia attachments	4 Send and receive RTF, HTML formatted mail.	5 Send Messages to distribution lists	6 Send Encrypted Messages	7 Send Authenticated Messages	News	2 Web Browser Clients	Message Access Paradigm	Secure E-Mail Standards	Certification System
1 Mail Transfer Protocol			+	+	+		+							
3 Message Access Protocol			+	+								+		
5 Directory Access Protocol	+	+												
6 Web Protocols						+				+	+			
7 MIME				+	+	+								
1 Secure E-Mail Standards								+	+					
2 Certification System								+	+				+	
3 Encryption Algorithm													+	+

Table 3.2: Related functional attributes.

4. CASE STUDIES

In order to evaluate the mail quality model proposed in section 3.2, we have been provided with two real study case requirements. The first case (*case study A*) is a public institution that will provide mail services to about 50000 users. The second one (*case study B*) is a small software consultant and *Internet service provider* (ISP) company. Case study B, manages an Internet portal with several news groups and discussion lists. It also provides mail services for internal and external users.

The complete list of requirements for case A is:

Interconnection related requirements:

1. Support for the commonly use certification standard X.509.
2. Support for access to server from other applications.
3. Support for web access.
4. Support for connection to other mail networks by standard protocols.
5. Support for standard mail clients.

Functionality related requirements:

6. Automatic message redirection.
7. Message prioritization.
8. Permissions management (e.g.: Message size limits, destinations etc.).
9. Protection against viruses and any other risks.
10. Anti-spam filters (by subject, by origin etc.).
11. Individual or organization level distribution lists.
12. Data confidentiality (for messages stored and in transit)
13. Message authentication.
14. Support for any type of attachments.
15. Folders and subfolders at any level.
16. Rich character set.
17. Message sorting by different criteria.
18. Spanish language support.
19. Messages must never get lost.

Utilization Related requirements:

20. The expected amount of users is 50000. From them 3000 are more active (50 messages per day which may include attachments), the remaining users make a more sporadic use of service (average of 5 messages per day with no attachments included). High concurrency of users is expected
21. The service must always be available.
22. Message trough output time, must be inferior to 1 minute for messages with no attachments. For messages with attachments must be inferior to 5 minutes per megabyte.
23. Helpdesk must be included with the application.
24. Service monitoring utilities (e.g.: statistics, configuration, system validation, etc.)
25. Installation time inferior to one month after platform become operative.

The complete list of requirements for case B is:

1. Server must support SMTP, POP3 and IMAP4 protocols and be capable for WEBMAIL.
2. Management of 30 discussion lists and news groups with about 100 users subscribed to each one.

3. Possibility to automate subscription to mail lists.
4. Filters and rules to move messages containing attached files with certain extensions (e.g.: .vbs) to alternative folders, from where they can be later reviewed. This may be useful to exclude possible virus files.
5. Incoming mail monitoring and server access log. Possibility to eliminate mail from incoming and outgoing queues directly from monitoring program.
6. Mail delivery notifications, possibility to configure parameters such as maximum number of delivery retries, and time between them.
7. Management of mail priorities, for example mail sent to lists should have lower priority that mail sent to individual users.
8. Possibility to grant permissions to some users, to send mail from server computer.
9. Possibility to ensure that mail do not get lost if mailboxes run out of space.
10. Multi-domain administration of mail server.

The matching of requirements of cases A and B with attributes and subattributes in quality model is shown in Appendix A. Some requirements (e.g.: case study A requirements 2 and 9) are too general. A more detailed specification must be provided to better classify them. Some others (e.g.: case study B requirements 6 and 9), ether require or imply a mixture of functionalities, which may be supported by selecting several attributes and subattributes of model. Here again we believe that further feedback may be required in order to be better classify them. But even with these inconvenients we were able to easily categorize them. An approval mark followed by a interrogation sign is used to represent the mentioned and other similar cases in appendix A.

We have also tested our model by mounting the characteristics of two of the most popular commercial mail servers available in market: Microsoft Exchange 2000 (Enterprise edition) and Lotus Domino R5 Server. We obtained their basic characteristics form [8][9][15] and [L2]. Furthermore we have also installed them on a Windows 2000 Advanced Server (Service pack 2). The purpose of this experience was not to do performance or other advanced test, but to gain a better understanding of some of their characteristics that are mentioned in literature. We were unable to find information about some of the products functionalities and we are not experienced users of either of the products, so it is clear that the list of characteristic provided in appendix A, may not reflect the complete and real functionality of the tools.

Once incorporated into the model we were able to easily compared requirements with product functionalities. This allowed us for example to detect that requirements 6 and 9 of case study B were not supported (at least not in the extend required) by Lotus Domino Server (R5). Exchange 2000 on the other hand, includes features that support those requirements and may therefore be more eligible for this case. We do not want to explicitly name one of the tools as better that the other for each case (because as mentioned earlier some of their features may not be included in appendix A) but rather to show, that by using our proposed quality model it is possible to identify differences between tools and better evaluate them .

It is our believe after this experience, that once case requirements get refined and full product features get mounted on model, It may be a very effective tool to help in the selection of the products that are best suited for each case. This quality evaluation must be obviously complemented whit other relevant factors, such as cost, political issues, etc.

5. CONCLUSIONS AND FUTURE WORK

In this paper we propose a mail server quality model base in the ISO/IEC software quality standard. The model is composed of attributes and subattributes which have been categorized under the six ISO/IEC quality characteristics. Although we are aware that it may be very difficult to list all the quality attributes that mail servers may provide, we believe that attributes and subattributes selected for our model, represent most of the conceptual functionalities that today mail servers have to offer.

The process used in this paper to build a quality model for the mail server domain, can be extrapolated and used as a methodology to build quality models for other software domains. The process is composed by seven steps, a preliminary one in which the domain frame work is defined, and six more steps which can be used intertwinedly or iteratively at any acceptable extend, until model is completed. These steps which are described in section 3 can be formally listed as:

- Step 0. Defining the domain.
- Step 1. Determining quality subcharacteristics.
- Step 2. Defining a hierarchy of subcharacteristics.
- Step 3. Decomposing subcharacteristics into attributes.
- Step 4. Decomposing derived attributes into basic ones.
- Step 5. Determining metrics for basic attributes.
- Step 6. Stating relationships between quality entities.

Two real case studies have been used to evaluate the model. We found no problems at the time of matching the requirements with attributes in the model. This was fulfilled even though some requirements were very general and in some cases had to be related to several attributes. It was our feeling after this process that the model is complete enough to successfully accommodate a wide range of different real case requirements. We also believe that after requirements are mounted in the model, a feedback process is required to refine and complete them, in order to have a better product selection scenario.

The model has also been tested by evaluating, two popular commercial mail server products. Here again, we found no problems to accommodate their characteristics and it was our feeling that the model is flexible enough to fulfill this task. Some work related to metrics remains to be done, but we have tried to include examples in most of the attributes, to facilitate this work to future users.

We believe that the model can be effectively used in the selection of mail server software. We did not explicitly choose one of the products to be the best for each case studied, because at the time that tests were performed, we did not have some information about products that was relevant to this mission. But once requirements and products characteristics are formulated with respect to the quality model, we were able to easily compare and identify differences among them.

References

[1] *A. Rodríguez, J. Gatrell, J. Karas, R. Peschke*. **"TCP/IP Tutorial and Technical Overview".** IBM Red Books. August 2001.

[2] *Compaq Computer Corporation: Knowledge Management and Messaging solutions*. **"Deploying Microsoft Exchange 2000 clusters"**. Compaq Computer Corporation, May 2000.

[3] *M. Barrios, O. Conradsen, C. Haramoto, D. MarinHigh*. **"Availability and Scalability with Domino Clustering and Partitioning on AIX"** IBM Red Books, Agust 1998.

[4] *T. Gray*. **"Message Access Paradigms and Protocols".** University of Washington, September 1995.

[5] *A. Menezes, P. Van Oorschot, and S. Vanstone*. **"Handbook of Applied Cryptography".** CRC Press, August 1997.

[6] ISO/IEC Standards 9126 (Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for their use, 1991).

[7] *L. Chung, B. Nixon, E. Yu, J.Mylopoulos*. **"Non-Functional Requirements in Software Engeneering"**. Kluwer Academic Publisers, 2000.

[8] *The Microsoft Exchange Server Product Group*. "**Exchange 2000 server a Comparison to Lotus Notes/Domino R5 Enterprise Messaging, Reliability, Manageability and scalability"**. Microsoft press, October 2000.

[9] *Lotus Development Corporation*. **"Inside Notes: The Architecture of Notes and the Domino Server"**. Lotus, July 2000.

[10] *E. Gerck*. **"Overview of Certification Systems: X.509, PKIX, CA, PGP & SKIP"**. The Bell, October 2000.

[11] *Knowledge Management and Messaging Solutions.* **"Deploying Microsoft Exchange 2000 Clusters"**. Compaq Computer Corporation, May 2000.

[12] *Gregory Pfister.* **"In Search of Clusters"**. Prentice Hall, August 1998.

[13] *W. Stallings.* **"Data and Computer Communications". 6th edition**. *Prentice Hall, 2001.*

[14] *A. Tanembaum*. **"Computer Networks" 3rd edition**. Prentice Hall, 1997.

[15] *Creative Networks.* **"A Comparison of Exchange and Domino Development"**. Creative Networks, March 2001.

Links

[L1] The Internet Mail Consortium. **http://www.imc.org/.**

[L2] Microsoft Exchange 2000 online documentation.
http://www.microsoft.com/exchange/en/60/exhelp/default.asp

Appendix A.

Matching of case study requirements and product features, with mail server quality model characteristics.

Symbols Table	
Symbol	**Meaning**
✓	Feature is supported by application.
-	We were not able to find information about the feature, or we may not confirm or deny if is supported by application
1/2	Feature is partially supported
x	Feature is not supported
✓*	Feature is supported with additional software components
✓?	Additional Information from users is required.
	A metric for feature must be defined