

Diagnosing Delivery Problems in The White House Information Distribution System

Mark Nahabedian & Howard Shrobe
MIT Artificial Intelligence Laboratory

Abstract:

As part of a collaboration with the White House Office of Media Affairs, members of the MIT Artificial Intelligence Laboratory designed a system, called COMLINK, which distributes a daily stream of documents released by the Office of Media Affairs. Approximately 4000 direct subscribers receive information from this service but more than 100,000 people receive the information through redistribution channels. The information is distributed via Email and the World Wide Web. In such a large scale distribution scheme, there is a constant problem of subscriptions becoming invalid because the user's Email account has terminated. This causes a backwash of hundreds of "bounced mail" messages per day which must be processed by the operators of the COMLINK system. To manage this annoying but necessary task, an expert system named BMES was developed to diagnose the failures of information delivery.

Background

In January 1993, the new Clinton administration committed itself to the use of electronic media such as Email (and later the World Wide Web) for making government information widely available to the public. A collaborative effort between the White House Office of Media Affairs, the MIT Artificial Intelligence Laboratory and others quickly created a workable framework for wide-scale distribution of a stream of daily documents originating in the Executive Office of the President. The document stream includes daily press briefings, speeches by the President and other officials, backgrounders, proclamations, etc. In addition, the stream of released information includes special documents such as the National Performance Review's reports on reinventing government, the proposed health care reform legislation, the yearly budgets, etc.

The Intelligent Information Infrastructure Project at the MIT Artificial Intelligence Laboratory created an information distribution server which functions as the focal point of the distribution chain. Documents are released from the Executive Office of the President through this system; they

are sent from this system to a variety of archiving and retrieval systems around the country, to most on-line services (e.g. CompuServe, America Online), to about 4000 direct subscribers to the MIT server, and to a variety of other servers which further redistribute the documents. A survey of people connected to this distribution chain estimated that more than 100,000 people were receiving information through this medium.

Documents released through this service are coded with descriptive terms taken from two taxonomies: the first taxonomy categorizes the type of document (e.g. Press Release vs. Speech vs. Press Conference); the second taxonomy concerns content (e.g. Foreign Affairs, Domestic Affairs, Economy, Taxes). Subscribers to the service specify a personal profile consisting of combinations of the descriptive terms which characterize their interests; it is the servers job to guarantee that subscribers receive exactly those documents which match their profiles in a timely manner.

Users establish a subscription and modify their profiles by filling out electronic forms (using either Email or the World Wide Web) and submitting them to the server. The ease with which users can manage their profiles is an important measure of the quality of service delivered.

The Problem

The environment just described is open, large scale, and anarchic. The system services thousands of users at hundreds of sites in dozens of countries. Users may establish, modify and terminate subscriptions at any time. User Email addresses registered with the server may become invalid at any time; occasionally users cancel their subscriptions before this happens, but this is comparatively rare. Also, configuration problems at subscribers' sites make their Email addresses temporarily unreachable even though the addresses are valid.

Kinds of Mail Notification

Delivery Failure
Delivery Failure but still Trying
Message Received
Message Opened
Message Deleted
"Vacation" Notice

Causes of Delivery Failure

User Not Found
Host Not Found
Mailer Configuration Problems
Temporary Mailer Resource Problem
DNS Configuration Problem
User Mailbox Full

In either of these cases, "bounced mail" messages are sent to the MIT server informing it of the inability to deliver a message to the invalid Email address. Most Email systems do not consolidate these bounced mail messages; if you send two messages to an invalid Email address, you receive back two bounced mail messages. The White House information stream typically includes as many as a dozen documents a day; with a subscription base of 4000 direct subscribers this leads to a rather large volume of bounced mail traffic each day (more than 100 messages). The failure to handle these messages and to update the subscription database accordingly, leads to a perception by the administrators of the receiving sites that they are being "spammed"¹ by the sending site; given that the sending site in this case is the White House, it is unacceptable to ignore the bounced mail traffic. A second class of problem arises when a user with a valid Email address attempts to terminate or modify a subscription without success; in this case, the perception is that the White House is spamming the subscriber personally, an even more unacceptable situation.

On the surface, it would seem that this problem is amenable to simple automation. However, the open, anarchic character of the Internet makes the problem quite complex: there are dozens of different mail servers, each with a unique "bounced mail" message format. In addition to the variety of Email servers speaking the Internet's native SMTP protocol [RFC821] there are also a large number of other protocol domains bridged to the Internet. These include UUCP, Bitnet, X.400 and a large number of proprietary Email systems (e.g. CC:mail, Microsoft Mail, etc.); bounced mail messages are often reformatted as they cross the bridge between protocol domains, sometimes losing information (and sometimes preserving information which is useless, such as one which directs the recipient to press the F1 key for more information). Within these other mail domains, the format of a mail address might be different from that used in the Internet; bounced mail messages from

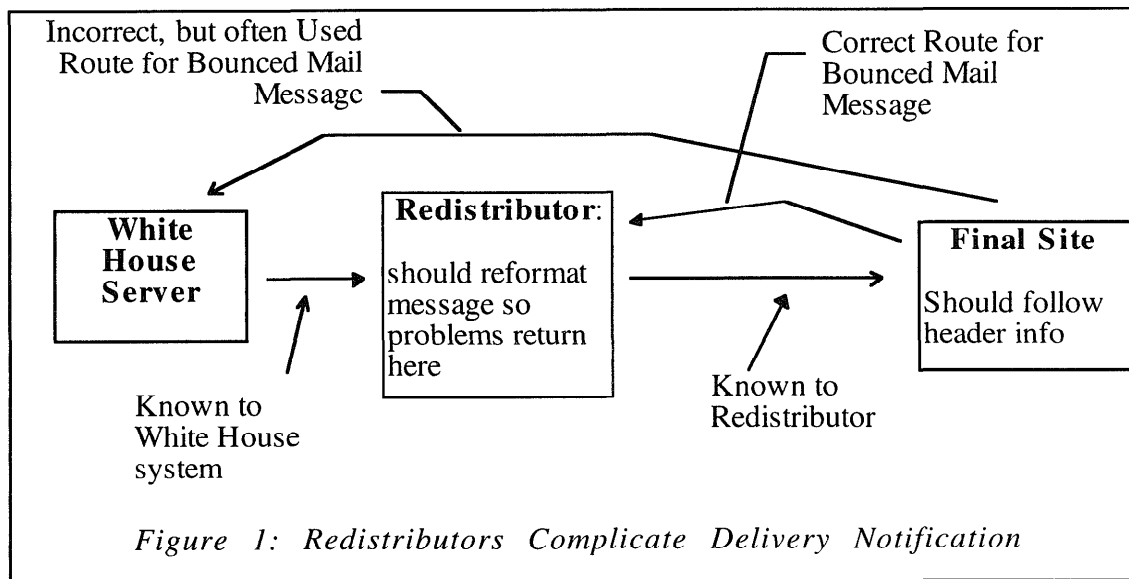
these domains often include their foreign format email address, rather than the Internet format address in our database.

A second set of complications arises from the variability of user's Email addresses. Many people have several Email addresses some of which are forwarded to another. Bounced mail messages in such cases often refer to the "forwarded to address" which isn't in our subscription database. Furthermore, people often subscribe using one address, switch to a second one as their primary address (forwarding the first one to the new address) and then more or less forget about the first address; attempts to modify the subscription using the new, primary address are then unsuccessful, because the system is unaware of the new address. Similarly, if the new address becomes invalid, then a bounced mail message will be sent to the server referring to the new address, which is unknown to the server.

In some mail systems (e.g. UNIX) users may direct their mail streams to shell scripts or other programs for processing. "Vacation programs" are a common example of this, they send back to the sender a message saying that the recipient is away and unlikely to respond soon. This is a courtesy when sent in response to a personal correspondence but when sent back to a bulk distributor like the White House server it shows up as part of the bounced mail stream. In addition, nothing prevents users from writing new mail handling programs, including incorrect ones; when such programs fail, the sender of the message (as opposed to the author of the buggy program) is usually sent a bounced mail message (in principle the postmaster at the receiving site should be sent this message, but principles and reality don't always correspond in this world).

A final complication, shown in Figure 1, arises because of the presence of redistributors. Redistributors are people or programs which receive the original message stream and then relay it to a set of subscribers known to the redistributor but not to the primary White House distribution server. Virtually any subscriber may independently decide to act as a redistributor of the document stream (for example, by establishing a mailing list). If an Email address on a redistributor's list becomes invalid, the redistributor should be notified; however, often the original source of the message (us) is notified instead. To get the behavior we

¹ *Spamming*: A colloquial term, now common in discussions about the Internet, which refers to the practice of filling up somebody's electronic mailbox with unwanted material, often advertisements, complaints or flames. Origin unknown.



desire, the redistributor should arrange for the headers of the redistributed message to identify it as the destination for problem reports; however, some redistributors fail to configure their mailer appropriately to achieve this behavior. This can also be caused by bugs in either the redistributor's software or in the Email server at the ultimate destination (e.g. failure to correctly interpret the headers) or both. COMLINK tries to distinguish redistributors from normal users by having different types of database entries for each; redistributor entries include an Email address for the administrator of the redistribution list. However, we rely on people to subscribe to COMLINK correctly and often people don't.

The problem of managing a large Email based distribution system in this environment has been recognized for some time [RFC1211]; however, to date the problem has been handled using one of two approaches:

Don't worry, be happy: In this approach, bounced mail messages are ignored. The sender builds up a rather large file of bounced mail messages which is periodically deleted. The destination sites receive many messages which are bounced, but this happens automatically. All told, a lot of resources are wasted, but nobody really cares because it's largely invisible. To be fair, most maintainers do from time to time examine a sampling of the bounced mail traffic and attempt to address the problems.

Big bag of tools to aid the administrator: A number of ad hoc tools are built to aid the system administrator in making sense of the bounced mail traffic [RFC1211].

These help the conscientious list administrator to solve difficult problems, but much of the work remains manual.

Given the high visibility of the White House distribution system and its role as an early experiment in using the Internet to improve government services, neither of these approaches was acceptable. Instead we decided to implement an expert system to aid in the handling of bounced mail and to help in managing other problems such as a user's inability to terminate or modify a subscription.

Structure of the System

The Bounced Mail Expert System (BMES) is a component of a larger system, called COMLINK, which is a substrate for building information distribution and group collaboration systems using Email, the World Wide Web and other Internet based transport protocols. At the core of COMLINK is an object oriented database which includes the following information:

Subscribers: Email address, personal name and subscriptions, date subscription started and date (if any) subscription turned off, whether this user is a redistributor.

Network Hosts: Subscribers at this host, upward and downward links in the domain name hierarchy, mail server type.

Documents: descriptive terms, release dates, subject, etc.

Queued Tasks: Time to execute the task, task type and arguments

BMES draws upon this information to help diagnose delivery failures.

BMES is a rule based diagnostic system driven by a file of bounced mail messages. Each message is a symptom of a failure in the delivery system. The user of BMES is the "postmaster" maintaining the White House COMLINK system. BMES's task is to discover, if possible, the reason why a mail message was bounced and if diagnosis is not possible to present meaningful information to the user and help in gathering more information. If diagnosis is successful, then the system rectifies the problem, usually by suspending a user's subscription.

For each message processed the system follows a standard pattern of processing:

Classification of the mailer which sent the message

Abstraction of the message to hide the syntactic differences between bounced mail messages.

Diagnosis of the cause of the delivery failure, including:

Heuristic generation of hypotheses

Interaction with administrators at remote sites.

The first task is Classification during which BMES matches features of the message against required features in the taxonomy of mailer types. In practice, the classification is done by a rather ad hoc set of rules which search for specific features in the headers and the first part of the body of the message. These features include characteristic substrings within particular headers or in specific locations within the body (usually the first several lines) of the message. These rules were determined based on the authors' observations of the bounce mail messages.

The system currently distinguishes 23 different types of mailers; these need not necessarily correspond to distinct pieces of mailer software, rather they correspond to the variety of distinct formats of bounced mail messages which we've observed. Some mailers have a rather broad range of configurability including the format of the bounced mail message to generate. We have no special knowledge of how the remote sites are being managed and so if two distinct hosts generate bounced mail messages which look different, we treat these as having been generated by distinct mailers even if this isn't necessarily the case.

New mailer types pop up occasionally, but this now happens rarely.

The second stage of processing is to Abstract the message, hiding the syntactic variability between the different formats of bounced mail messages but preserving their semantic commonality. For example, bounced mail messages typically contain a "transcript" which includes email ad-

resses to which it was impossible to make a delivery, and an indicator of the cause of delivery failure. Similarly, most bounced mail messages contain a copy of the original message that couldn't be delivered. The original message includes a set of "**received**" headers [RFC822], each of which corresponds to a mail server in the chain of delivery; the header identifies the host which handled the message, the time of handing, and in some cases the user to whom the message was intended to be delivered. (Note that this is different than the destination in the "to" header [RFC822] of the message, which is typically a generic address such as "**Clinton-distribution**").

Abstraction is effected using the object oriented programming techniques of CLOS [CLOS]. Once the classification stage has identified the mailer type, BMES constructs a CLOS object whose class corresponds to the type of the mailer. This object mediates the abstraction phase. We established a class hierarchy corresponding to the mailer types and an object-oriented protocol² that all mail messages must obey; the protocol consists of about a dozen methods. Each method in the protocol reflects an aspect of the common semantic content that any bounced mail message must contain. There is one method in the protocol which finds the transcript in the bounce message and a second one which maps over its failure descriptions, calling an action routine with the email address and a canonicalized version of the failure code. There are also protocol methods to locate the message text and then to map over the "**received**" headers [RFC822] contained in it. We use the class hierarchy to capture commonalities of message structuring. For example, the location within the bounced mail message and the encoding of the transcript and original message are idiosyncratic to each mailer, however several different mailers share the idea of partitioning the message body using the MIME standards [RFC1341] for structuring mail messages; however they may differ as to what fields they include. Therefore different classes implement the protocol methods differently, but where there is commonality this is captured by CLOS inheritance. All mailers which use MIME encoding, for example, are represented as subclasses of the common MimeStructured message class

² Here we use the term "protocol" in the same sense as in the "Meta-Object Protocol" [MOP] or the Joshua Protocol of Inference [Joshua], not in the sense of an Internet protocol such as SMTP [RFC822]. Fortunately, the object-model used here doesn't use the "message passing" metaphor or we would also have confusion between mail messages and messages being sent to objects.

The power of this approach is that it abstracts away the syntactic variability exhibited by the variety of bounced mail message formats, while highlighting their semantic commonality. Higher levels of the system can expect any mail message to contain standardized information and to behave in standard ways, without to be concerned with the underlying syntactic variability.

The next stage of processing is Diagnosis which involves deciding whether the failure is permanent and whether the recipient is actually known to the COMLINK system. If the address in the mail message is found explicitly in the COMLINK database, if the failure is due to the user's account being closed out (as opposed to a transient error) and if the user has an active subscription, then BMES cancels the subscription.

deliver a failure message only after this elapsed time. Because of this long latency, bounced mail messages can continue to arrive for several days after a user's subscription has been canceled. If a bounced mail message refers to an Email address whose subscription has already been canceled, then the user of BMES is not bothered since the problem has already been handled; the message is presumed to have arisen during the period between the time when the Email address became invalid and when COMLINK was informed of this. This requires COMLINK to maintain an entry for users whose subscriptions have been canceled for a period of time after the cancellation; when BMES cancels a subscription, COMLINK creates a queued task entry in its database with a firing time of one month in the future. When this queued task runs it completely removes the user's account from COMLINK's database. However, dur-

Let the name in the reported address be ?name-1
Let the host in the reported address be ?host-1
For each child ?child-host of ?host-1
If ?name-1@?child-host is the email address of an active subscriber ?sub-1
Then suggest that ?sub-1 is a possible cause of the delivery failure

Rule 1: Probable-User-is-Child-Host

However, sometimes the bounced mail message reports an invalid address which is not present in the COMLINK database. At this point, the Heuristic Generation phase is entered. A small collection of heuristic candidate generation rules is used to suggest candidate addresses which are in the database and which might have led to mail being sent to the address reported in the message. For example the message might report a problem with "foo@ai.mit.edu"; in this case if "foo@w.ai.mit.edu" or "foo@mit.edu" are in the database, they would be good candidates for possible causes of the failure. A rule called Possible-User-at-Child-Host suggests the first. A second rule called Possible-User-at-Parent-Host suggest the second. An English paraphrase of the first rule is shown in Rule 1.

Such candidate generation rules work by traversing COMLINK's map of the portions of the Internet domain name space for which it has subscribers. There are rules which suggest the superior domain (e.g. "mit.edu" is the superior of "ai.mit.edu"), any inferior domains (e.g. "w.ai.mit.edu" is an inferior of "ai.mit.edu"), and any sibling domains (e.g. "lcs.mit.edu" is a sibling to "ai.mit.edu") which the system knows about.

Most mailers attempt to deliver a message for several days when possibly transient problems are encountered; they

ing the intervening period, BMES can tell that it knows about this account and that it knows that it has already canceled the account's subscription.

Most messages are handled by the simple processing described above; however, there is usually a residual of harder problems left over. One cause of the residual problems is that many of the mailers provide minimally useful information in their bounced mail messages. In other cases, there is information provided but the bounced mail message refers to an Email address which isn't in the COMLINK database and none of the heuristics above lead to a known address either.

In almost all cases, this situation arises when the failing address is reached through an "indirection": Either the address is on the mailing list of a redistributor, or it is the target of a forwarding entry for some other Email address, or there is an MX³ record [RFC974] involved. In these cases completely automatic processing isn't possible; there isn't enough information available to BMES to form a full diagnosis of the problem. Some of the required informa-

³ MX records are part of the Internet Domain Name System; the MX record for a host specifies which machine should actually receive mail addressed to the original host.

tion is at a remote site and can be obtained only by communicating with an appropriate person at the remote site. It is a further complication that we don't actually know what remote site does have the information we need.

BMES can help make an educated guess: If it can find the original message included in the bounced mail message and if there are **received-from** headers in the original message, then the server mentioned in the header might have relevant information. In particular, any user at this server who is marked as a redistributor in the COMLINK database is a particularly useful candidate. Redistributor entries contain an Email address for the administrator of the redistribution list; BMES formats the first draft of a standard Email message to the maintainer asking if the failing address is known to the administrator of the list and, if not, requesting help in figuring out what else might be going wrong (the user is then offered the option of further editing the text of this message).

Another heuristic is to look for Email addresses similar to the failing one at each of the sites mentioned by the "received-from" headers and to then send to the postmaster at each of these sites a message explaining the problem and asking for help.

There are some techniques which we employ manually today which are subject to automation. One is used when there are a small number of users at the site which bounced the mail but when it still isn't possible to make a definitive identification of the invalid address (either because the bounced message doesn't contain an address or it contains one which doesn't match any entry in our database). In this case, we generate one message for each user in our database known at the site; this message explains that we are having delivery problems and asks for the user's help if possible. There are two useful outcomes: 1) One of the users knows what's going on and helps us fix it 2) One of these messages bounces, but since the bounced message has the specific user's address in it (which our normal messages lack since they are sent to the whole subscription list) we are now able to determine which address is invalid. This technique is analogous to techniques used in model-based trouble shooting where a new and maximally informative test is generated.

Application Payoff

This application is not a commercial venture and so payback in monetary terms is not a relevant metric for evaluation. BMES was created as a support tool within a collaboration between a research group at MIT and a line or-

ganization at the White House Office of Media Affairs. Each partner in this collaboration had their own goals: The participants from the Executive Office of the President wanted to make information routinely and reliably available to the public and to demonstrate the viability of the Internet as a model for the future National Information Infrastructure. The research group at MIT wanted to explore issues in computer supported collaborative work and in intelligent management of information. For both groups, management of the bounced mail problem is a necessary supportive task but one which cannot be allowed to consume valuable resources; in particular, neither group has substantial manpower to devote to the task. Therefore the relevant metric for evaluating the payback of the investment is in terms of the reduction of manpower contributions from the two groups. This in turn directly translates into the effectiveness of the system at handling bounced mail messages.

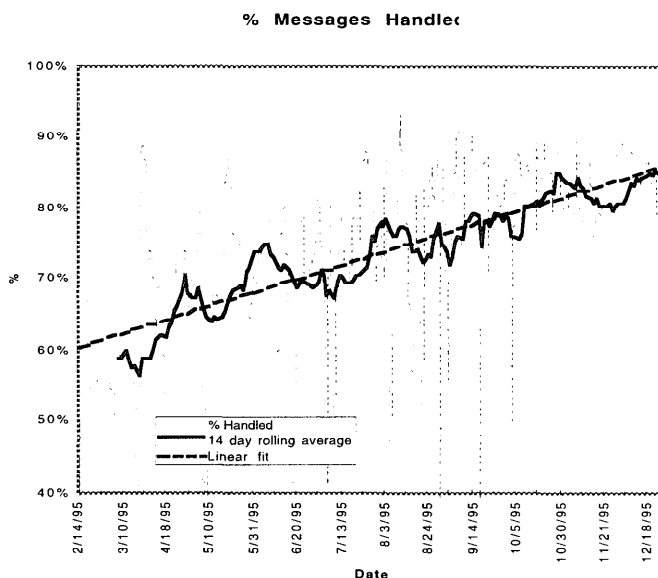


Figure 2: Effectiveness of BMES vs. Time

We have been collecting data on the effectiveness of BMES since early in its lifetime. Figure 2 shows this data for the bulk of calendar year 1995. During this period, 63,091 bounced mail messages were received. BMES was capable of automatically processing 48,031 of these message or 76% of the total. As can be seen from Chart 1, there is a great deal of temporal variability in the system's performance. It simply seems to be the case that some weeks we run into problems with sites whose mail servers provide less information; these weeks have lower overall perform-

ance. However, it is also noticeable that there is a long term trend of improvement in the system's performance. This is probably due to a combination two factors: 1) Over time, we have confronted most of the mailer types that exist and have built up useful heuristics for dealing with them. 2) Over time, there has probably been a stabilization of technology in the community and a switch to more robust and informative mailer software.

Over the whole lifetime of the project, the time per day put into bounced mail handling has declined from nearly 3 hours per day in calendar year 1993 to about 1/2 hour per day now. We would certainly like to drive this number down further, but the transformation so far has been a qualitative one: The three hours per day required at the start was simply not viable; today the task is annoying but well within scope.

File name	Chars	Lines	Defs
New-db-interface	2,180	69	10
User-rules	10,662	292	21
Zwei-msg	2,798	79	12
Understanding-bounced-mail	41,067	1,106	104
Zmail-commands	25,450	655	20
Mailer-vanilla-unix	10,308	256	11
Mailer-smailer	3,820	101	2
Mailer-compuserve	4,090	95	2
Mailer-mime	6,212	148	9
Mailer-mmdf	7,482	176	14
Mailer-pmdf	4,789	125	4
Mailer-mime-pmdf	5,810	163	6
Mailer-uucp	3,603	89	1
Mailer-uucp-warning	2,814	70	1
Mailer-ibm	3,352	86	3
Mailer-vines	3,130	79	2
Mailer-microsoft	4,894	124	4
Mailer-minus	2,724	71	3
Mailer-local-delivery-agent	5,242	127	6
Mailer-undeliverable	3,139	77	3
Mailer-cc	2,514	63	3
Mailer-aol	3,497	93	6
Mailer-lispm	4,825	122	4
Mailer-mercury	3,266	84	3
Mailer-ctstateu	3,324	80	2
Mailer-smtp	4,049	104	4
Mailer-ksgbbs	3,405	86	4
Bounced-mail-complaint-reply	1,882	51	5
Check-recipient	2,541	61	3
Simple-redirection	5,491	140	5
Relay-zmail-command	25,987	702	65
Total	214,347	5,574	342

Table 1: Code Distribution in BMES

Implementation

Both COMLINK and BMES are implemented within the Symbolics Genera environment, which runs both on Symbolics hardware and on Digital Equipment Corporation Alpha AXP workstations (using the Open Genera emulator software from Symbolics). BMES is integrated with Genera's ZMail⁴ mail client which is built on an extensible substrate for complex mail handling applications. Much of the system relies on this substrate for low level processing such as mail file and header parsing, pattern matching and string searching. BMES itself is implemented in Joshua [Joshua] and makes extensive use of its Protocol of Inference to reason about the contents of the mail messages. BMES itself is invoked as a Zmail command which is applied to the mail file containing the bounced mail messages. When mail messages need to be sent to postmasters or users at remote sites, this is facilitated by use of ZMail's programmatic interface. Table 1 shows the component files in the system, including number of characters and lines of source text and number of definitions (rules, lisp functions, methods etc.).

Deployment and Maintenance History

Work on BMES was begun in the spring of 1993 as an adjunct to a predecessor system to COMLINK (called FORUM) which represented the first collaboration between the MIT AI Lab and the White House Office of Media Affairs. The bulk of BMES was completed by the summer of 1993. As COMLINK's development proceeded, a second version of BMES was developed by modifying the first version to take advantage of the extra information maintained by COMLINK. For a few months, COMLINK and FORUM were run in parallel while users were encouraged to switch their accounts over. During this period, both versions of BMES were run to manage problems from the two streams. The final cutover to COMLINK was completed in early 1995. Since that time, new features have been added to BMES as necessary.

It is interesting to note that BMES was literally developed and deployed simultaneously; it was an experience in evolutionary design of a complex software system. As soon as there was useful functionality, it was deployed and then enhanced during its ongoing operation.

⁴ There are several other products named Zmail which are not related to the one included in the MIT Lisp Machine software systems and its commercial offshoots such as Symbolics' Genera.

BMES is an unusual application: It is a component of the COMLINK system which supports thousands of users but there is only one user of BMES itself. That user is also the developer and maintainer. Currently, the bounced mail processing is done at MIT; however, we anticipate complete hand-off of the COMLINK system in the near future at which time, personnel in the Executive Office of the President will assume responsibility. As with much else about this application a crisp definition of deployment is not easy. A large population has received information from the White House for several years now and the management of Email delivery problems has been substantially automated as part of that task. It is true that the system is still operated by its developers, but that was the anticipated situation at the outset. Routine sustainable operation has been achieved and that has enabled other aspects of the project to proceed without undue drain on scarce personnel resources.

Future Work

Though BMES greatly reduces the effort required to process the mail backwash from a bulk electronic mail distribution, there is room for improvement. The addition of some form of reverse mapping of MX records would help to identify an address on the distribution list based on an address as determined from a bounce message. The domain name system does not provide such a mapping, so one would have to be constructed by iterating over all mail sites in the distribution database and doing a domain MX lookup for each one. Because of changes to the distribution database and to the DNS, this reverse mapping would need to be updated regularly.

As it is currently implemented, BMES is difficult to extend as new mailer types are discovered and as existing ones change. This difficulty is because the work of identifying mailer type is distributed over a number of ad-hoc parsers. As one adds a parser to recognize a new mailer type, one must be careful that this parser does not also recognize the messages of previously implemented mailer types. Perhaps reimplementing the parsers using a rule based parser generator would simplify the definition of mailer types.

The ideal solution to the problem of handling bounced mail would be the universal adoption of standards which specify how mail delivery status information is reported. If delivery failure notifications explicitly stated the reason for failure, and the failing address, as well as any addresses from which it might have been derived, then BMES could be replaced by a much simpler tool. Only one simple parser would be needed to extract the information from the bounce

message. The system would require fewer, simpler rules for identifying the problem subscription. Recognizing the problem of numerous bounce mail formats, the Network Working Group of the Internet Engineering Task Force has recently proposed a set of standards [RFC1891, RFC1892, RFC1893, RFC1894] which specify how mailers should report delivery status. As sites upgrade their mailers to ones that adhere to these standards, there will be fewer and fewer bounce messages that will require a system like BMES to interpret.

References

- [CLOS] "Common Lisp Object System Specification", Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, and David A. Moon
Sigplan Notices, 23(Special Issue), September 1988.
- [RFC821] "SIMPLE MAIL TRANSFER PROTOCOL", Jonathan B. Postel, August 1982, Internet RFC 821.
- [RFC822] "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES", David H. Crocker, August 13, 1982, Internet RFC 822
- [RFC974] "Mail Routing and the Domain System", C. Partridge, January 1, 1986, Internet RFC 974.
- [RFC1211] "Problems with the Maintenance of Large Mailing Lists", A. Westine and J. Postel, March 1991, Internet RFC 1211.
- [RFC1341] "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", N. Borenstein, N. Freed, June 1992, Internet RFC 1341.
- [RFC1891] "SMTP Service Extension for Delivery Status Notifications", K. Moore, January 1996.
- [RFC1892] "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", G. Vaudreuil, January 1996.
- [RFC1893] "Enhanced Mail System Status Codes", G. Vaudreuil, January 1996.
- [RFC1894] "An Extensible Message Format for Delivery Status Notifications",] K. Moore, January 1996.

[COMLINK] "The Open Meeting: A Web-Based System for Conferencing and Collaboration," Hurwitz, Roger & John C. Mallery, Proceedings of The Fourth International Conference on The World-Wide Web, Boston: MIT, December 12, 1995.

[GENERA] **Genera Reference Manual**, Symbolics Inc.

[MOP] **The Art of the Metaobject Protocol**, Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow, MIT Press, 1991.

[Joshua] "Joshua:Uniform Access to Heterogeneous Knowledge Structures (or Why Joshing is better than Coniving or Planning), S. Rowley, H. Shrobe, R. Cassels, W. Hamscher, AAAI National Conference on Artificial Intelligence, 1987, Pages48-52.

[ZMAIL] **Editing and Mail Manual**, Symbolics Inc.