Text Is Just a Special Case: How MIME Makes Multimedia Mail Manageable For The Masses

Nathaniel S. Borenstein
<nsb@bellcore.com>

Bellcore (Bell Communications Research)
Morristown, New Jersey, USA

I. Motivation: Why Extend Email?

II. Background: Internet Mail Today

III. MIME: History and Design Rationale

IV. MIME: Technical Overview

V. The MIME Content-types

VI. Non-ASCII Characters In Message Headers

VII. Controversies and Problems

VIII. Status Report

IX. Implications for Email and Beyond

X. Future problems in multimedia data interchange

I. Motivation: Why Extend Email?

Text mail, fax, voice mail very popular.

Text mail deficient for non-English speakers due to ASCII character set.

Modern computer hardware can handle images and audio, too.

Integrated multimedia mail, as in Andrew, Slate, NeXT, etc., has wide appeal.

X.400, which purported to solve the problem, does not.

Why Is Data Interchange Hard?

Hundreds of Incompatible Formats

Standards have failed in two ways

ODA -- re-invents world, still not enough!

X.400

- Got Distracted
- Re-invented the wrong things.
- Complex, Relative little-used in practice
- Incomplete for Multimedia
- Requires software revolution, not evolution

Bottom line: Heterogeneous formats seem here to stay.

II. Background: Internet Mail Today

Internet Mail: the *de facto* standard

Defined by RFC 821/822

Basis for most workstation & PC mail environments

Even more incomplete than X.400

Non-standard multimedia extensions (Andrew, Next, etc.)

The plan: Let email evolve gracefully

Need standardized Internet multimedia extensions

III. MIME: History and Design Rationale

Designed by Internet Engineering Task Force Working Group on Email Extensions

Politics: remarkable alliance between X.400-lovers and X.400-haters

Working Group formed, Fall 1990.

First MIME draft, Spring 1991.

Proposed Internet Standard, Spring 1992.

Primary design goals:

Text mail in any human language.

Non-text mail in any conceivable media type.

Complete compatibility with RFC 821 and RFC 822.

Total robustness over all known real-world email transport systems.

Openness to multiple well-known formats.

Easy Extension to new types and formats.

Formal mechanism for type registration to promote interoperation.

Easy interoperation with X.400.

IV. MIME: Technical Overview

RFC 822 defines a message header (set of structured fields) & plain text body.

From: Nathaniel Borenstein
<nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com
Subject: Plain old email

This is a plain old email message.
It contains ASCII text, nothing more.

MIME: Technical Overview, continued

MIME allows extended bodies, with type information in header fields.

Content-type header field provides type/subtype, optional parameters

> Seven types are defined, many subtypes are expected.

Content-Description gives a textual description of the body data

Content-ID gives unique identifier for body parts.

Content-xxx are the only header fields that matter in body parts.

Content-Transfer-Encoding

Binary and long-line data does not survive email transport

Need to specify encoding algorithm for email transport

Two transfer-encoding algorithms defined

Base64

The densest simple (non-compressed) encoding possible for email.

33% data expansion

Uses only A-Z a-z 0-9 + / =

Content-Transfer-Encoding, continued

Quoted-printable

Maximizes readability of included ASCII

All ASCII characters unchanged except =

Other characters represented as =0A, etc.

Up to 200% expansion.

Encoding Design Rationale

uuencode -- insufficiently robust

Base64 based on PEM algorithm

Simple Examples

From: Nathaniel Borenstein
    <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com
Content-type: text/plain
Content-Transfer-Encoding:
    quoted-printable

This is text with a single non-ASCII character, =FF.

From: Nathaniel Borenstein
    <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com
Content-type: image/gif
Content-Transfer-Encoding: base64

R0lGODdhSgGgAfUAAAENDQ01NTTw8PEVF...

V. The MIME Content-types

MIME defines seven content-types. Most extension via subtypes.

1. text.

Subtypes: plain, richtext. (More expected.)

Critical parameter: charset.

"text/plain; charset=iso-8859-1" permits French email.

Other character sets: ISO-8859-[1-9].

Deferred, but expected: Asian languages, UNICODE, ISO 10646. Politics...

text/richtext is an extremely simple "common denominator" markup language for enriched text.

A plain text message in US-ASCII:

From: Nathaniel Borenstein
<nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Plain text mail
Content-type: text/plain; charset=us-ascii

This is plain text mail.

A plain text message in French (using ISO-8859-1, quoted-printable encoding)

From: Nathaniel Borenstein
<nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: French mail
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: quoted-printable

Le courrier =E9lectronique =E0 la fran=E7aise n=E9cessite quelques caract=E8res sp=E9ciaux pour faciliter la t=E2che du lecteur et =E9viter les ambigu=EFt=E9s

Why the Encoding?

Isn’t ’à’ better than ’=E0’?

-- Of course.

-- But SMTP is 7-bit

-- Using so-called "8-bit clean" SMTP causes real problems

-- Can have "8-bit clean" networks with gateways to Internet/7-bit-SMTP

-- Mark internal 8-bit mail with:

```
Content-transfer-encoding: 8bit
```

A Hebrew/English richtext message

From: Nathaniel Borenstein
<nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Hebrew richtext
Content-type: text/richtext;
charset=iso-8859-8
Content-transfer-encoding: quoted-printable

This is <bold> enriched </bold> mail. Note the <italic> dramatic</italic> use of<nl>fonts, even in <italic> =FA=E9=F8=E1=F2 (Hebrew) </italic> mail. <nl><nl>By the way, my Hebrew name is <bold>=ED=E5=EC=F9 =EF=E1 =E9=EC=FA=F4=F0</bold>,<nl>not =EC=F0=FA=F0 as you might suppose.

2. image.

Subtypes GIF, JPEG. Others expected.

3. audio.

Subtype "basic" for single-channel 8Khz u-law. Others expected.

4. video.

Subtype "mpeg". Others plausible.

5. multipart.

Allows multiple body parts of different types, each structured like a mini-message.

Subtypes: mixed, alternative, parallel, digest.

Mixed: simple (serial) combinations.

Alternative: multiple representation of the same data.

Parallel: for parallel presentation *if possible*.

Digest: has special defaults for message digests.

All multipart subtypes share a syntax, allowing future experimentation in structured types, etc.

From: Nathaniel Borenstein
    <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed;
    boundary=CUT_HERE

--CUT_HERE
Content-type: text/plain

Hey, Ned, look at this neat picture:
--CUT_HERE
Content-type: image/gif
Content-Transfer-Encoding: base64

5WVlZ6enqqqqr....
--CUT_HERE
Content-type: text/plain

Wasn’t that neat?
--CUT_HERE--

From: Nathaniel Borenstein <nsb>
To: Ned Freed <ned@innosoft.com>
Subject: An alternative example
Content-Type: multipart/alternative;
    boundary=CUT_HERE

--CUT_HERE
Content-type: text/plain

Hey, Ned, Isn’t MIME great?
--CUT_HERE
Content-type: text/richtext

Hey, <bold> Ned</bold>, isn’t
MIME <italic>great</italic>?
--CUT_HERE
Content-type: application/ODA
Content-Transfer-Encoding: base64

5WVlZ6enqqqqr....
--CUT_HERE--

6. message.

Subtypes: rfc822, partial, external-body

"message/rfc822" allows encapsulated message.

"message/partial" allows automatic fragmentation and reassembly.

"message/external-body" allows data to be passed by reference (hyperlinks).

From: Nathaniel Borenstein
    <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A partial example
Content-Type: message/partial;
    number=1; total=2;
    id="unique-id"

Content-type: image/gif
Content-transfer-encoding: base64

5WVlZ6enqqqqr...

From: Nathaniel Borenstein
    <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A partial example
Content-Type: message/partial;
    number=2; total=2;
    id="unique-id"

Ozs3h4eIKCgo6Ojp...

```
From: Nathaniel Borenstein <nsb>
To: Ned Freed <ned@innosoft.com>
Subject: Some external references
Content-Type: multipart/alternative;
     boundary=42

--42
Content-Type: message/external-body;
     access-type=mail-server
     server="listserv@bogus.bitnet"

Content-type: application/postscript

get rfc-xxxx doc
--42
```

```
Content-Type: message/external-body;
      name="BodyFormats.ps";
      site="thumper.bellcore.com";
      access-type=ANON-FTP;
      directory="pub/nsb"

Content-type: application/postscript
--42
Content-Type: message/external-body;
      name="/u/nsb/BodyFormats.ps";
      site="thumper.bellcore.com";
      access-type=local-file

Content-type: application/postscript
--42--
```

7. application.

Catch-all.

Subtypes: PostScript, ODA.

Most creative extensions expected here.

Possible examples:

EDI

Acknowledgement-request

Interactive-survey

Already registered:

ATOMICMAIL

Andrew-inset

A Complex MIME Message

From: Nathaniel Borenstein
    <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A complex example
Content-Type: multipart/mixed;
    boundary=FOOBAR

This is the preamble area, which is ignored.
--FOOBAR
Content-type: text/plain; charset=US-ASCII

PLAIN TEXT GOES HERE.
--FOOBAR
Content-Type: audio/basic
Content-Transfer-Encoding: base64

BASE64-ENCODED AUDIO DATA
--FOOBAR

Content-Type: image/gif
Content-Transfer-Encoding: Base64

BASE64-ENCODED IMAGE DATA GOES HERE
--FOOBAR
Content-type: text/richtext

This is <italic>richtext</italic>.
<nl><nl> Isn’t it <bigger>cool?</bigger>
--FOOBAR--
This is the "epilogue" area, also ignored.

What the Complex Example Looks like with a standard mail reader

What the Complex Example Looks like with an integrated multimedia mail reader

Minimal MIME-Conformance

Open-ended, needs interoperable core

"MIME-conformance" defined to require:

1. "MIME-Version: 1.0"
2. Content-Transfer-Encoding
3. text/plain in US-ASCII
4. ASCII portions of ISO-8859-*
5. Intelligent treatment of unrecognized types and character sets. (E.g. decode & write to file)
6. message/rfc822
7. multipart (mixed, alternative, *)

VI. Non-ASCII Characters In Message Headers

MIME, as defined by RFC 1341, only addresses message bodies.

The need for non-ASCII header text is addressed by RFC 1342, a companion document to MIME.

Rationale:

More complex

More ugly

Less consensus

Non-ASCII header data is handled as a special case.

Only text (no images, etc.) is permitted

Non-ASCII text is permitted only in certain very special locations

All of this is complicated and made uglier by RFC 822 header syntax.

Old parsers of RFC 822 headers should not fail with these extensions.

The Non-ASCII data are represented as "encoded words", but the details are not given here.

Non-ASCII Data Are Permitted...

Only in fields intended for human reading. (Not automatic processing):

Subject, Comments, Content-Description

Subject: **NON-ASCII**
Comments: **NON-ASCII**
Content-Description: **ETC**

Syntactic comments in any field

From: nsb (**NON-ASCII**)

Route phrase in email addresses

From: **NON-ASCII** <x@y>

Examples of Non-ASCII Header Data

What follows are two examples of header fields containing human names in Hebrew and Japanese.

Due to the use of large fonts in this slide, the encoded word appears on multiple lines, but should not really.

From: nsb@bellcore.com (=?ISO-8859-8?Q?=ED=E5=EC=F9 =EF=E1 =E9=EC=E8=F4=F0?=)

From: nsb@bellcore.com (נתנאל בן שלום)

To: ysato@etl.go.jp (=?ISO-2022-JP?B?GyRAOjRGI0stGyhK?=)

To: ysato@etl.go.jp (佐藤豊)

VII. Controversies and Problems

Banned/punted:

- Nested encodings
- compression
- uuencode
- Lots of top-level types

X.400 interoperation required:

- No use of preamble/epilogue in multipart type.
- Parts are NOT encapsulated messages.

Transport Issues:

- No dependence on line or byte counts.
- Binary or 8-bit transport neither presumed nor prohibited.
- Initial character sets limited to ISO-8859 family.

VIII. Status Report

At least 20 implementations under way.

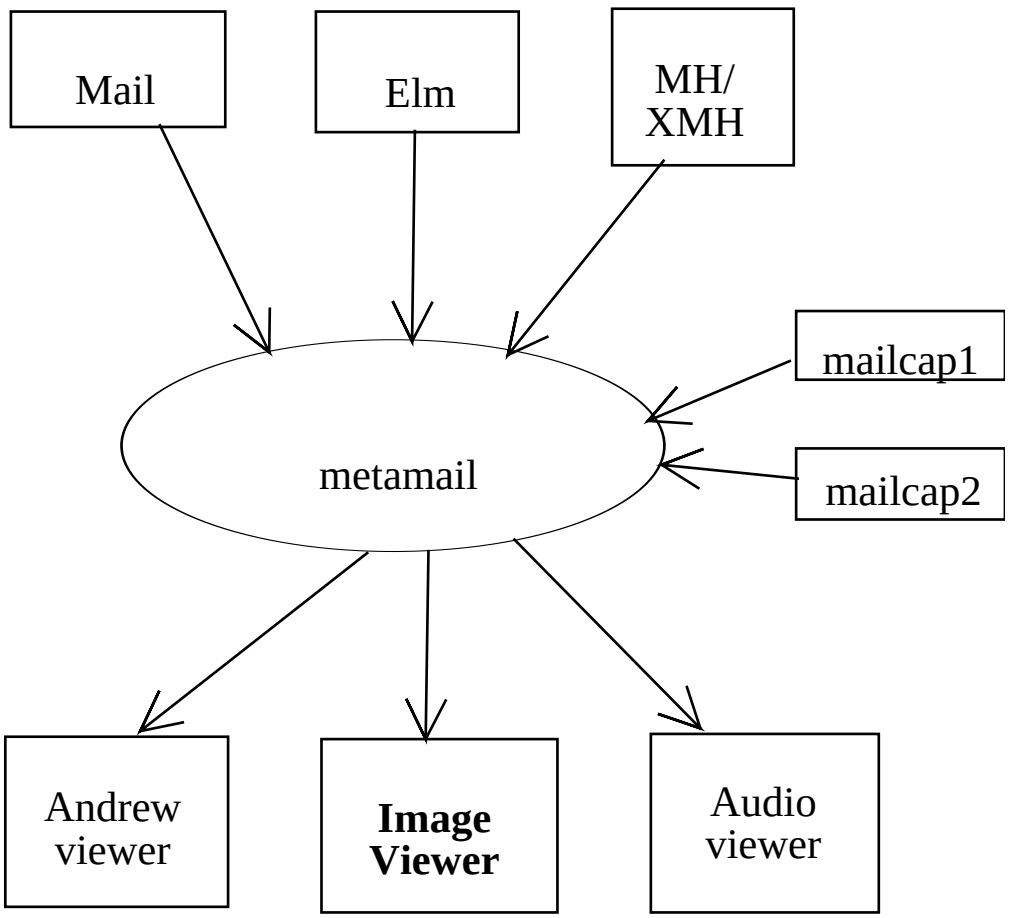
At least 3 public domain implementations available now.

One of these, metamail, includes patches for over a dozen popular UNIX & DOS mail readers. After two months, in use at hundreds of sites.

MIME -> Proposed Internet Standard

The Metamail Software

Easily Patches dozens of mail readers on UNIX, DOS, and Amiga to make them MIME-capable, facilitating MIME transition.



IX. Implications for Email and Other Multimedia Applications

Whither X.400? Choose one:

MIME is the death of X.400

MIME will help X.400 a lot.

Whither Fax?

Probably no effect.

Might facilitate future merging with email

Whither voice mail?

May help with interoperation.

MIME in non-mail applications

Paranoid email robustness can’t hurt elsewhere.

It’s nice to know your data format is mind-bogglingly robust.

Open architecture invites re-use.

Metamail software already used in non-email applications. (Superbook, Cruiser, experimental systems)

De Facto standards happen from the bottom up.

X. Future problems in multimedia data interchange

Constraining and defining the set of formats in use.

Standard formats for tight coupling of separate objects.

Standardized control structures for interactive applications.

New media types (e.g. smell, virtual worlds)

All of this seems to be MIME-able.

Access to MIME

Easy way: all available via mailserver.

Mail to mailserver@thumper.bellcore.com.

Subject: autosend: mime-txt

Subject: autosend: metamail-info

Or anonymous ftp, thumper.bellcore.com, pub/nsb/{BodyFormats.*,mm.tar.Z, etc.}

For more information:

Nathaniel S. Borenstein
<nsb@thumper.bellcore.com>

Bellcore
Bell Communications Research