Design issues for lucy

Bruce R. Lewis

December 6, 1990

Contents

1 Overview **2**

2 User interface for asking a question **2**
2.1 Requirements 2
2.2 Xlogin issues 3
2.3 Present design 3

3 Interface for Lucy administrator(s) **4**
3.1 Basic requirements 4
3.2 Examining the queue of questions 4
3.3 Forwarding a question 5
3.4 Answering a question 5
3.5 Present design 5

4 Implementing privacy **6**
4.1 Privacy through public-key encryption 7
4.2 Anonymity through an anonymous server 7

1 Overview

This document is intended to speed the design of the lucy system; it is not intended to be a publishable paper describing a finalized design. The goal here is to give readers a clear idea of what lucy is trying to do, and what issues in the design need to be discussed.

A lot of what we eventually decide is "a good thing" for lucy to have will not be possible to implement by January, when it is intended that lucy come into actual use. But knowing what we want to add later will help lucy be designed in such a way as to facilitate later expansion.

The lucy program is not yet available, but the following paragraphs describe what it will be like:

Lucy is an electronic ombudsman, ready to receive questions of all kinds from the community at large. Any member of the MIT community may ask a question on-line, ranging from "I'm new to the area. Where is a good, moderately priced restaurant suitable for an evening out?" to "I recently stole a small amount of cash from a fellow staff member. No one suspects but I'm feeling horribly guilty. What should I do?"

Communications will be anonymous, unless the sender chooses otherwise. A subset of lucy questions and answers will be available on-line to the community for browsing (with the sender's permission only and with all potential identifiers removed). Starting in late January, anyone wanting to communicate with lucy will be able to submit a question by running an application that will be available as a pre-login option on any Athena workstation. (An Athena account is NOT required.) Even if the user logs in before sending the question, the username will be removed from the message before it is sent to lucy, so that it will be sent anonymously unless a name or other identifier is explicitly added. Questions which include the sender's name may be answered personally, while questions with no identifying information will need to be read in browsing mode.

2 User interface for asking a question

2.1 Requirements

The entire MIT community needs to have anonymous access to lucy. The easiest way to implement this is through mail, but given universal access to dialup machines, other channels might be possible.

The Athena interface needs to allow the user to edit a question easily, and should include an explanation of how to get a private reply. Mail might be used as the underlying mechanism for the Athena interface, or some other mechanism of better reliability and anonymity might

be used. Either way, the interface needn't be any more complicated than sendbug. It should be usable from dialup or as a pre-login application under xlogin.

The pre-login interface should allow an option for browsing.

2.2 Xlogin issues

Tty pre-login applications, like techinfo, come up in an xterm with a very large font. This may not be appropriate for lucy, in that such a large font is easy to read from a distance. Users entering questions under xlogin should be able to be more discreet.

Xlogin prevents users from leaving a workstation idle with a pre-login tty application running by periodically checking the idle time on the tty, and killing the application if it's been idle too long. Doing the same thing with lucy means making users edit questions with `emacs -nw`, which makes sure emacs is connected to a tty, but disables positioning the cursor with the mouse or arrow keys. This would probably be disconcerting for most emacs users.

2.3 Present design

The script `/mit/lucydev/bin/ask` fulfills the minimum requirements, but doesn't address the xlogin issues. The script `/mit/lucydev/.xlogin` presently only tells the user that lucy isn't ready yet, and what it is. Here's a conversation with the current `ask` script:

```
Right now, lucy is being experimented with by Athena staff, so your
please don't send questions of any personal nature.  Only send a
question if you're testing the system.

You will edit your question in emacs.  To exit emacs, first hold down
the Ctrl key, then press the x and c keys in sequence, then release the
Ctrl key and press y.

Press Return or Enter when ready to compose your question.


Please enter a one-line subject for this question.  The subject line
will help Lucy handle your question better, and will help you find your
question if it should appear in the public discuss meeting later.
 -->
Any subject will do (e.g. Perplexed in Peoria).
 --> Forced to Enter Subject line in Cambridge
If you have identified yourself in your message, lucy is able to reply
to you personally.  Would you like lucy to do so?
```

```
 -->
Any answer will do (yes, no, maybe, only if ...).
 --> I haven't identified myself.
If you permit it, lucy will publish your question/answer with any
identifying information removed in a browser viewable by anyone.  If
you chose not to receive a personal reply, this is your only means of
seeing the answer to your question.  Would you like your
question/answer to be published?
 --> no thanks
Your question has been sent to Lucy.
```

The above dialogue will produce a question that looks something like this:

```
[0015] daemon@ATHENA.MIT.EDU  Lucy_experimental  12/06/90 10:34 (6 lines)
Subject: Forced to Enter Subject line in Cambridge
Date: Thu, 6 Dec 90 10:33:58 -0500
To: lucyx-mtg@euphrosyne.LOCAL

Like, do you really expect us to believe that this is anonymous?
SEND PERSONAL REPLY:  I haven't identified myself.
PUBLISH QUESTION/ANSWER IN BROWSER:  no thanks
```

3 Interface for Lucy administrator(s)

3.1 Basic requirements

The interface should be usable from dialup, so that Lucy doesn't need to have a workstation in her office or come to a public cluster to work. But Lucy shouldn't need to be especially quick with computers to use the interface.

3.2 Examining the queue of questions

Lucy needs to be able to quickly see

- how many questions are in the queue
- when each question was submitted
- the status of each question
  - new

  - forwarded to an expert
  - tentatively answered (not necessarily published or returned)
- the subject line for each question
- the content of any individual question

3.3 Forwarding a question

Experts may not have athena accounts, so this has to be done by email if it is to be done electronically at all. It should work like `forw` in mh, only instead of inserting a mail message, a question, a space for an answer, and instructions for answering are inserted. Even if forwarding is done with, say, interdepartmental mail, there still needs to be a command for logging the fact that a question has been forwarded.

3.4 Answering a question

Should Lucy choose to answer a question herself, it should be easy to bring up a form with the question and a space for the answer, so that any identification can be removed before the question/answer pair is published.

An answered question should be easy to publish in the browser, at which point it should be removed from the queue and put in the archive.

At the present time, lucy would like to be anonymous. This presents a problem in the current implementation of discuss, because anonymity, although possible, is linked to inauthenticity. However, users who post things for the browser must be authenticated, or anyone could post a bogus question/answer pair.

3.5 Present design

The plan is now to write Lucy's interface with the ss library, the same library used for discuss. That doesn't mean the interface will be exactly like that of discuss; the syntax will be more like that of mh. Perhaps a curses interface would be more appropriate; if so, effort would be best spent expanding the ss library to do curses, rather than writing Yet Another Probably Flakey Curses Interface.

Here's what it looks like now:

```
Available Lucy requests:
```

```
quit, exit, q           end this session
status                  count questions in queue
list, scan              list questions in queue
show                    display a question (and related info if appropriate)
answer, repl            answer a question
forward, forw           forward question to be answered by someone else
publish, post           publish question/answer
help, ?                 redisplay this menu
Lucy:
```

4 Implementing privacy

The only mechanism implemented so far is to provide user anonymity by having `dsmail` strip all the headers from a mail message except the subject and date lines. This mechanism is questionable for the following reasons:

1. Email is not considered completely private, especially if for some reason a message should become a dead letter.

2. Sendmail by default logs the senders of incoming messages before the messages are piped to `dsmail`.

3. There is no provision for someone who wants an anonymous personal reply.

4. How does Lucy remain anonymous while authenticating herself to discuss? Does she want to remain anonymous when dealing with experts? If so, where do their email replies go?

[We'll have a group discussion about the relative seriousness of these problems at the design review.]

The rest of this section introduces some possible solutions, none of which are set in stone. In fact, it may be too difficult to implement any of them by January. But it's good to discuss them now so that whatever solutions are eventually implemented can be integrated smoothly into what exists.

4.1 Privacy through public-key encryption

If the question-asking mechanism encrypts the question with lucy's public key before sending the question across the network, lucy can then use a private key to decrypt it. The implementation of such a scheme is completely open to discussion.

4.2 Anonymity through an anonymous server

Suppose a user doesn't want to identify himself to lucy, and also doesn't want his question published in the browser. The present scheme doesn't allow for that, as the sender isn't stored anyplace where the system can use it.

Say, for example, mail was used. A server could be set up so that you could send mail to lucy@anonymous.LOCAL, and your question would be forwarded as if it were from, say, sender90125@anonymous.LOCAL. It could be replied to without Lucy ever learning who the sender was.

[This still leaves open the dead letter problem, but if mail is unacceptable, we can come up with other means.]