A Patient Centered Asynchronous Messaging System
a Project Concept Proposal
for
Advanced Software Engineering – W4156
by
Team: Messengers

Davis Bu
db398@columbia.edu

Maryam Kamvar
m kamvar@cs.columbia.edu

Larry McKnight
mcknight@DMI.columbia.edu

Peter Stetson
stetson@DMI.columbia.edu

Sept. 30, 2002
Basic Concept

Our project will be a skeletal system for a hospital based message system that allows providers to send notes about patients to each other. The metaphor for this is an e-mail system, differentiated by the fact that every note will have a patient context, and that messages will be sent to “roles” rather than specific providers. The use of roles can also be conceptualized as a whiteboard in that it is a shared workspace between providers and about a specific patient. The concept of roles is critical since the specific providers involved in a patients care are often changing and it can be difficult to identify the specific provider in a given role. The patient context is critical for security and auditing purposes. See the appendix for a more detailed rational behind the system.

Users

- Physicians
  - Interns (Dr. Newby)
  - Residents (Dr. Checker)
  - Attending (Dr. Guru)
- Nurses (Nrs. Nightengale)
- Patients (Mr. Sikly)
- Lab Alerting System

User Stories

-High Priority: Login, Send Messages, Read Messages

1. Dr. Newby logs in and sees all the notes that are waiting on all his patients. Notes are organized by patient.

2. Nurse Nightengale logs onto the system and sends a note to the intern on patient Mr. Sikly. She doesn’t know who the intern who is covering the patient currently. Dr. Newby logs in as the intern on Mr. Sikly and checks the system for messages. He sees the note from the nurse. He responds to the message. Nurse Nightengale sees the response.

3. Dr. Checker, is looking at Mr. Sikly’s medical record and wants to send a message to Dr. Newby to followup on a lab.

4. Mr. Sikly gets his blood work done in the outpatient lab. The lab sends an alert for rising creatinine. Dr. Newby logs in and sees this message.

5. Mr. Sikly has access to the system through his home Web access to medical record. He sends a note to Dr. Guru (also his outpatient physician), asking about his medicines. Dr. Guru replies. Mr. Sikly logs in later and sees the answer to his question.

-Medium Priority: Return Receipts, Forwarding, Deletes

1. Dr. Newby is looking at Mr Sikly’s record and doesn’t know which medicine to prescribe. He sends a note to Dr. Checker, who also doesn’t know. Dr. Checker forwards the message to Dr. Guru. Dr Guru replies to Dr. Newby and Dr. Checker the answer.
2. The lab sends an alert note with a return receipt to Dr. Newby. As Dr. Newby clicks on the note to view it, he is prompted to send confirmation that he has received the note. Dr. Newby presses “OK” and reads the message. The lab sees the return receipt.

3. Dr. Newby “deletes” a note. The message disappears from his view, but is still present in the database.

-Lower Priority: Digital signatures, Logs, Attachments

1. Dr. Newby sends a note to Dr. Guru on Mr. Sikly. He is prompted for a password as he tries to send it. A checksum is generated incorporating the note, and the time. The checksum is signed with Dr. Newby’s private key and the system private key, and stored with the message. Dr. Newby’s public key and the systems public key are used to decrypt the checksum and verify that the note was unchanged.

2. Mr. Sikly requests to see all the users that have viewed his whiteboard items.

3. Dr. Guru wants to attach a pdf of a journal article to Dr. Newby regarding Mr. Sikly’s diagnosis.

Proposed Architecture

The system architecture will consist of several layers of interrelated modules (see attached diagram). Users will access the system through a standard web browser calling a cgi interface. The cgi interface will be responsible for session handling and authentication. The cgi will also detect the browser type then pass the page request to a browser specific formatting module. The formatting module will in turn call object modules that return meaningful chunks of unformatted data. The object modules will interface the back end SQL database to actually store the data.

Due to the limited time frame given for each iteration, there are several differences between system designed for this class and a production version. The system for this class will provide the base functionality for a real system with the ability to add functionality later. However, only the core functions will be implemented (top figure). For example, a real system (bottom figure) will need to support many browser types including palm devices, and older browsers with limited functionality and newer browsers with full standards support. These browsers share common data groupings (for example a providers patient list, or a list of note headers) but the html formatting on a palm device would need to be formatted quite differently and may need different page calls than a standard browser. Therefore, a separate formatting layer is used to minimize duplication of common code elements. In this class however, only one browser type will be implemented, though the formatting layer will is left in the design. Similarly, in the real system some information about providers (names, authorization) is managed in a LDAP, and patient information (names, demographics) and provider patient lists would come directly from the electronic medical record, which is on a mainframe. For this class, our system will provide simplistic tables to represent patient and provider data that would come from these other systems. Porting the system is facilitated by abstracting all data access through data access objects. Security in the production system would be handled thru the larger EMR system (WebCIS) which establishes session ID’s. For this class, a fake session ID will be given at login, and sessions will be simply verified to this artificial number (ie, no timeouts/relogin, no auditlogs, etc). Finally, A real system would need to include digital signatures, attachments, and return receipts, which we will plan room for, but will not implement. Our project will only include the skeleton functions to create, read and reply to notes. Second priority functions include the ability to forward notes and reply to all members of the team. Little priority will be given to interface aesthetics.
Figure 1: The Component Architecture
Controversies

The main controversy at this time is over the appropriate metaphor for the system. Some of the group feel that the primary metaphor should be a "whiteboard," stressing that posted notes be stored as a single note that is viewed, edited, or deleted by any of the team members. Under this metaphor the multiple roles that are allowed access would need to be defined at the time note is created. Although note versions would be archived, only a single latest note would be viewed, and replies would be implied through edits or deletion of the note. This metaphor has no clear concept of a hand-off of responsibility for note tasks except by defining who can view the note and the natural responsibility inherent in the providers' credentials to carry out the tasks requested in the note.

Others in our group feel that the metaphor is closer to that of an e-mail system. Under this metaphor, a single note is viewed by a single "role." Providers in a role may reply and forward notes to other roles, by creating a new copy of the original note with the ability to edit this copy. This metaphor assigns ownership for each note because each forwarded note has a unique author and note can not be edited once posted. Unfortunately, the e-mail metaphor breaks down when notes with tasks are forwarded to multiple roles because multiple providers may see different views as forked notes evolve differently over time. To solve this, larger roles (groups) can be defined such as "Whole Team," or "Any MD," and notes could be limited to a single recipient, but these larger groups need to be explicitly defined in advance of the message. For example, a note could not be written that was only shared by a Nurse and Social Worker unless a NurseSocialWorker role was defined.

Interestingly, the user interface for both metaphors can be strikingly similar. The user still needs to login to a role, see their patients, the note headers for that patient, and of course read, create, and reply/edit the note. The differences in functionality are primarily defined by differences in the data storage model and the whether the notes are addressed to a singular role or multiple roles. Since the e-mail metaphor is less complex to implement we will use it for this project and re-factor the back end model as it becomes necessary.
Appendix: Justification

Several studies have reported on the need to establish system processes that assure quality health care and systematically reduce medical errors. This is difficult since care is increasingly complex and provided by teams of professionals. Communication between providers is felt to be an area which system processes can be significantly improved. Specifically, providing an asynchronous messaging system with feedback to assure that messages were received and acted on is theorized to prevent medical errors by decreasing unnecessary interruptions that distract providers from a given task.

For example, suppose a laboratory needs to notify a provider that a lab result on patient is abnormal. The lab needs confirmation to ensure that the result was received by the provider. Currently, the provider would be paged to call the lab on the alert value. The provider receiving the page would likely be seeing another patient. Because he cannot tell how important the page might be he must interrupt his work flow on the patient, call the lab, wait while the lab operator looks up the reason for the page and the report the value, determine that the result can be addressed later, then resume where he left off with his original patient. He must also remember to act on the alert when he is finished. The lab has passed on its information and has verification that the result has been received, but at a great expense to the cognitive overhead of the provider.

Sending this result asynchronously so that the provider can review and respond when he is not busy seeing some other patient would be of benefit. Suppose the message were sent by e-mail with a return receipt. The lab has confirmation that the message was received and if no receipt is obtained in a given time, other methods of contacting the provider can be tried. But now the provider can check his e-mail at a more convenient time between patients.

Unfortunately, security and privacy issues make current e-mail systems impractical. Suppose, for example, that the patient requests that only providers that are directly associated with his care see his lab results. Digital encryption can assure that help prevent hackers from sniffing traffic, but tracking down where emails get stored and forwarded insecurely would be difficult if not impossible.

It can also be difficult to even identify the specific provider (and thus his e-mail address) associated with a patient at any given point in time. For example, a patient may be transferred from the Intensive Care Unit to the hospital ward where there may be a different team of providers. There may also be transfers of care for cross-coverage on nights and weekends. Much of this information is not currently available to the CIS, or when known may be old as patient care changes quickly. Alternatively, the roles associated with patients are relatively well defined even when different people take those roles. For example, in the academic setting, an Intern generally writes all orders for the patient even though the specific Intern may change with cross-coverage, or service transfers.

Therefore, a new system is needed that allows providers to view and send asynchronous messages about their patients. The system should have several characteristics:

**Patient Context:** Each note should have a patient identifier, and be stored so that all notes on a given patient can be easily logged and reviewed. Patients have roles associated with them which providers can accept.

**Security:** Data should be sent over encrypted channels, have strong authentication and authorization, and have the capability for digital signatures to ensure messages are unaltered. Messages should never be deleted from the database, but can be marked as obsolete so they are never displayed. Logs should be kept of access.

**Feedback:** Authors of notes should have the ability to request notification when recipients have read or acted on notes.

**Role Based:** Although authors can be defined, recipients may be identified by the role they play in patient care rather than the specific provider that provides that role.
Integration The system should be quickly accessible through the standard EMR and use the EMR for patient and provider identification. If providers have patient lists defined for the EMR, they should be available in the new system so only one list needs to be maintained.