Published in final edited form as:
Proc IEEE Int Conf Pervasive Comput Commun. 2005 March 21; 2005: 209–212. doi:10.1109/PERCOMW.2005.22.

Cascading Policies Provide Fault Tolerance for Pervasive Clinical Communications

Rose Williams, **Srikant Jalan**, and **Edie Stern**
IBM Thomas J. Watson Research Center, Hawthorne, NY

Yves A. Lussier, MD
Department of Biomedical Informatics, Columbia University, New York, NY

Abstract

We implemented an end-to-end notification system that pushed urgent clinical laboratory results to Blackberry 7510 devices over the Nextel cellular network. We designed our system to use user roles and notification policies to abstract and execute clinical notification procedures. We anticipated some problems with dropped and non-delivered messages when the device was out-of-network, however, we did not expect the same problems in other situations like device reconnection to the network. We addressed these problems by creating cascading "fault tolerance" policies to drive notification escalation when messages timed-out or delivery failed. This paper describes our experience in providing an adaptable, fault tolerant pervasive notification system for delivering secure, critical, time-sensitive patient laboratory results.

1. Introduction

Inadequate access to information and ineffective communication among patient care team members are potential proximal causes of medical errors [1]. It is well documented that automated alerting systems can improve provider response time by efficiently detecting and communicating a wide variety of clinical conditions [2,3]. It is also documented that the frequent occurrence of asynchronous messages in clinical data processing necessitates the inclusion of a general notification system as one of the core components of healthcare information management architectures [4]. Wireless handheld technology, including devices such as two-way alphanumeric pagers and personal digital assistants (PDAs), offers portability and mobile access to necessary information. While two-way pagers can receive messages and transmit responses wirelessly to any email address, PDAs offer more functionality, larger screen sizes and better input mechanisms. Additionally, PDA phones, such as the Blackberry handheld device, offer all-in-one solutions by combining the features of a cellular phone and PDA.

We successfully implemented a novel and complex system that delivers content to medical providers based on their roles and hospital notification policies at the time of a clinical event. We used IBM's WebSphere EveryPlace Access (WEA) and the Intelligent Notification System (INS) for delivering the notifications [10]. INS is capable of delivering notifications across a heterogeneous set of mobile gateways that support mobile devices, pagers, instant messaging, SMS and email. INS also allows the user to define different set of devices to receive notifications of different priority, for e.g. a user can specify that the Normal messages are notified using BlackBerry device and Urgent messages are notified

{rosemw@us.ibm.com, sjalan@us.ibm.com, estern@us.ibm.com}
yves.lussier@dbmi.columbia.edu

using Email, Pager and the BlackBerry device. However for this project, our design was predominantly centered on the delivery of the clinical notification using the new Blackberry “mobile data push” technology. As such, our system delivered secure, time-sensitive patient data to the “right person” at the right time – as long as the device remained powered on and in-network. As we began our preliminary clinical trials, we encountered numerous challenges associated with disconnection and lost mobile-delivery laboratory results. In designing a solution to this problem, we created and implemented cascading fault-tolerant message delivery policies.

2. Problem

Mobile devices are increasingly a key element in critical business applications, as well as in our daily lives. Critical applications require a high degree of confidence that actions will be done with certain reliability guarantees [6]. However, transient failures in network and device are common for mobile handsets [5]. Our objective for this project was to provide secure and reliable messages related to critical medical events to the right physician in a timely manner. In our solution we anticipated the need to accommodate minor network outages, loss of signal strength and cellular dead zones. In a hospital environment, loss of connectivity to a device is very common. There are cellular dead zones within the building, physicians are very mobile within the hospital, and in some areas (e.g. the intensive care unit or ICU), hospital regulations require that devices be turned off in order to prevent possible interference with medical equipment. We planned to overcome this by storing undelivered messages, and resending them when the target device was back in communication. However, in addition to these communications challenges, during our deployment of the solution, we came across some major unanticipated problems. We experienced:

- Cellular data network outages lasting for hours; and in one case for days. Note that the outage did not involve the heavily used cellular voice network.
- Human error on the part of the system administrator during configuration changes in mid-deployment. This affected messages delivered to all devices on the wireless network. User error affecting individual devices also occurred.
- Problems in the message delivery middleware system for the devices. Messages not delivered due to devices being out of coverage were not reliably retransmitted once communication was reestablished.
- Random failure of the devices to register with the cellular data network.

Even more than in most application domains, loss or severe delay of messages in the medical environment is of serious concern. Notification algorithms required in these scenarios must be fail-safe; unless “positive” acknowledgement is received, the message must be delivered to another responsible party [7]. Further, we discovered that physicians will not even proceed with testing if the system is not reliable. They will adopt a system only if it is fast, reliable, easy to use and provides benefit without increasing their existing process time.

3. Solution

We implemented a novel and complex system of cascading hospital policies in our solution to provide fault tolerance in a heterogeneous environment with both wired and wireless endpoints. Different approaches have been used in other solutions to provide reliable delivery of messages over a specific network. One technique employs a Store-and-Forward service: the message to be forwarded is stored in a temporary local cache, and is transmitted when the network connection is available again [8]. Another approach lies in transmitting the alert to a call center rather than to an individual (e.g. a Managed Care Center) [9]. Our approach towards fault tolerance is different from existing work. We have approached the problem as one of guaranteeing message delivery to the most appropriate of several valid recipients, using any available device and network in a timely manner, rather than guaranteeing message delivery to a particular device.

In our solution, hospital notification policies are used to define a cascading series of roles and users who are to be notified by the system. When messages are sent, a positive acknowledgement is expected from the recipient in a specified timeframe. If this does not occur, the next user in the series is notified. Notifications can be sent to different devices with different priorities using different wireless, cellular and wired networks, thus avoiding a problem specific to any given network or middleware platform. We also defined the concept of a final catch-all user, able to receive messages on the most reliable network. The messages are sent to this catch-all user if there is no positive acknowledgement after exercising all the policy definitions, that is, if we have exercised the full chain of recipients without receiving a response. Note that user errors in entering hospital notification policy definitions, role assignments and the like will result in the messages being sent to the catch-all user, thus allowing graceful correction. Further, we note that a hospital administrator need not be versed in the failure modes of different devices or networks in order to create these policies; policies are based on domain specific requirements of role and hierarchy, independent of the underlying physical infrastructure.

Users may provide a message acknowledgement from the BlackBerry device or from any other device, including desktops, capable of accessing a web interface. If the user is notified on multiple devices, they may choose to provide acknowledgement using any of the available devices. Users can provide a negative acknowledgement if they do not accept responsibility for handling a message. This may occur if the user believes the message has been erroneously sent to them, or if they are unable to handle the message due to exigencies of the moment.

An example of a set of hospital notification policies is given in Table 1. We define 4 users: Intern1, Intern2, Resident and the Hospitalist in increasing order of hierarchy. Messages are initially delivered to the responsible intern with a Normal priority, as determined from the patient information and hospital schedules. In Table 1, we show the first delivery attempt is retried with the same recipient but with an Urgent priority if there is no positive acknowledgement within the specified period of 10 minutes; afterwards they escalate to those in higher authority: first the Resident, then the Hospitalist.

If no positive acknowledgment is received from the last one on the chain, then the catch-all user is notified via E-mail. We assume that the Email system is the most reliable network and the catch-all user has been configured properly by the system administrator (note that the catch-all user does not appear in the policy definition). When the catch-all user has received a message, due to error or circumstance, it is that user's responsibility to physically track down a responsible person using any means of communication available.

Physicians are uncomfortable with systems that are not reliable from the beginning. While clinicians develop coping mechanisms to deal with miscommunications (e.g. questioning results that do not “fit” the pathophysiology, seeking test results that should have been available by now), such coping increases the burden on them. Lack of confidence in the system quickly results in lack of adoption. Because of this, even a system used on a trial basis must be reliable, or negative early reports will prevent further deployment. Figure 1 shows the flow of events that take place in our system, from the generation of the alert information, to the acceptable disposition of the message.

Once we implemented the cascading policies and the catch-all user concept, we were able to reliably deliver content to medical providers based on their roles and hospital policy. Deployment in real world environments of necessity entails equipment and network failures, and user errors. In using policies and cascades geared to the requirements of the specific domain, and the specific business entity, we were able to ensure satisfactory message delivery.

4. Concluding Remarks

During the course of our trial we learned the following:

- In medical environment, early failures in reliability will predispose the physicians from using the system, even if the system is made reliable at a later time.
- Cascading policies can provide reliability across failures in the different network, devices and middleware solutions. Policies should be associated with business requirements rather than infrastructure attributes.
- There should be a catch-all user defined to receive messages over the most reliable network. This user is notified for all messages that are not positively acknowledged by any user, and also if the policy definition does not resolve to any valid user due to an error in the definitions. The catch-all user can provide reliability in the face of common administrator errors and communication failures.

5. References

1. Chen ES, Mendonca EA, McKnight LK, Stetson PD, Lei J, Cimino JJ. PalmCIS: A Wireless Handheld Application for Satisfying Clinician Information Needs. J Am Med Inform Assoc Jan-Feb;2004 11(1):19–28. [PubMed: 14527976]
2. Kuperman GJ, Hiltz FL, Teich JM. Advanced alerting features: Displaying new relevant data and retracting alerts. Proc AMIA Annual Fall Symp 1997:243–7.
3. Warner HR Jr, Miller S, Jennings K, Lundsgaarde H, Pincetl P, Robinson EN Jr, Sommers J, Childress C. Clinical event management using push technology--implementation and evaluation at two health care centers. Proc AMIA Symp 1998:106–10. [PubMed: 9929191]
4. Geissbuhler A, Grande JF, Bates RA, Miller RA, Stead WW. Design of a General Clinical Notification System Based on Publish-Subscribe Paradigm. Proc AMIA Annual Fall Symp 1997:126–30.
5. Islam N, Zhou D, Shoaib S, Ismael A, Kihakkiniyil S. AOE: A Mobile Operating Environment for Web-based Applications. Proc Applications and the Internet 2004:15–24.
6. Joseph A, Kaashoek F. Building Reliable Mobile-aware Applications Using the Rover Toolkit. Proc MOBICOM'96. Nov;1996
7. Tsui FC, Pike J, Pike L. Design of a Clinical Notification System. Proc AMIA Symp 1989:989–93. MM.
8. Kaddour M, Pautet L. A Middleware for Supporting Disconnections and Multi-Network Access in Mobile Environments. Proc Pervasive Computing and Communications Workshops 2004:187–191.
9. Iordache SD, Orso D, Zelhinger J. A Comprehensive Computerized Critical Laboratory Results Alerting System for Ambulatory and Hospitalized Patients. Proc MEDINFO 2001:469–473.
10. IBM Websphere Everyplace Access. http://www.ibm.com/software/pervasive/ws_everyplace_access.

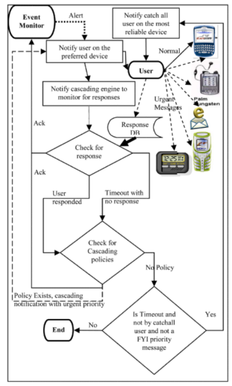

Figure 1.
System event flow

NIH-PA Author Manuscript
NIH-PA Author Manuscript

NIH-PA Author Manuscript
NIH-PA Author Manuscript

Table 1

Sample policy definition

Notify User	Notify Priority	Ack Time Min	Cascade To User	Priority	Device
Intern1	Normal	10	Intern1	Urgent	BB*, Email, Pager, SMS*
Intern2	Normal	10	Intern2	Urgent	BB, Email, Pager, SMS
Intern1	Urgent	5	Resident	Urgent	BB, Email, Pager, SMS
Intern2	Urgent	5	Resident	Urgent	BB, Email, Pager, SMS
Resident	Urgent	5	Hospitalist	Urgent	BB, Email, Pager, SMS

BB: BlackBerry device

SMS: Short Messaging Service for cellular phones