ENTERPRISE MONITORING AT A MULTI - FACILITY HEALTHCARE ORGANIZATION

by

Chris L Morey

A capstone project submitted in partial fulfillment of
the requirements for the degree of

Masters of Biomedical Informatics

Oregon Health Sciences University

2006

TABLE OF CONTENTS

School of Medicine

Oregon Health & Science University

Certificate of Approval

This is to certify that the Capstone Project by

Chris S. Morey

"ENTERPRISE MONITORING AT A
MULTIFACILITY HEALTHCARE
ORGANIZATION"

Has been approved

[redacted]

Professor in Charge of Capstone/Date

ACKNOWLEDGMENTS

I would like to thank my capstone advisor, Paul Gorman, for his guidance during the project and particularly the IRB process. I appreciate the time he was able to give during a busy schedule. This degree also would not have been finished without the love and support of my wife while time was split between work and school. I am also grateful for all the help Andrea Ilg has provided me while working through this program. Finally, I appreciate the support my employer for providing a flexible work schedule to complete school and who has provided resources for projects/classes.

ABSTRACT

The implementation of information technology in the healthcare environment has grown rapidly in the last several years. Increased reliance on information systems can result in negative end-user acceptance and impact patient care when there are system issues or application downtime. Enterprise monitoring is a means of preventing this negative impact on end-users and patients. This study reviewed enterprise monitoring of a laboratory information system at a multi-facility healthcare organization. This review was accomplished by gathering current monitoring parameters from the monitoring application, IT staff workflow from the organization troubleshooting database and current staff training procedures. Though enterprise monitoring is effective at this organization, areas to improve upon include improving alert instructions or unclear messages, checking for obsolete system functionality, looking for underutilized or misdirected alerts and implementing staff training documentation.

Introduction

A laboratory technician registers a clinic patient in the registration system but then can't find the corresponding registration in the laboratory information system to place orders. A nurse in the intensive care unit enters orders for STAT potassium blood levels but the phlebotomist doesn't come to draw the specimen. Doctors in the emergency room try to look up cardiac test results which should be complete but cannot find the results in the emergency room management system. The calls quickly start to come in to the information system help desk which quickly escalates the issue to the Lab IT team. Analysts on the team determine the inbound and outbound interfaces from the laboratory information system are no longer processing. The vendor is called for help but needs time to login and look at the system. In the meantime, hospitals and clinics throughout the state are sent to downtime procedures by the control center for lab orders and result reporting. This downtime results in frustration by staff as orders are handwritten, sent to the laboratory and then results are manually faxed back. Staff will then need to reorder tests and manually enter results after the interfaces are back up. All this costs time, money and has the potential to negatively impact patient care. It all possibly could have been prevented or the downtime shortened if enterprise monitoring had been in place.

Enterprise monitoring provides the ability to monitor and manage a variety of applications across multiple hardware platforms from a central location [1]. Utilized properly, enterprise monitoring can help prevent system issues and provide rapid problem resolution.

The healthcare setting is an environment where information technology has been implemented rapidly due to reports such as To Err is Human: Building a Safer Health System which indicates 44,000 to 98,000 deaths occur each year due to human error [2], efforts by groups such as The Leap Frog Group [3] and government initiatives from The United States Department of Health & Human Services [4]. Applications have become increasingly complex resulting in more strain on IT support departments. As information technology is increasingly utilized in healthcare, application or hardware issues can adversely affect patient care and end user acceptance.

This study involves a large regional multi-facility healthcare system using multiple software applications across a variety of hardware platforms to provide patient care. The laboratory information system (LIS) is one of these clinical applications used for management of laboratory data. The LIS is found in seven hospital facilities and multiple outpatient stations/clinics. The application generates approximately 20,000 patient charts per day, 30,000 outbound system interface transactions daily, 4,000,000 laboratory orders per year and interfaces 70 plus laboratory instruments [5]. The LIS interacts with other healthcare system applications such as the inpatient document system used by clinical staff, the outpatient electronic medical record system, the web front ends to several clinical systems and the healthcare system's lifetime electronic medical record. Due to the widespread utilization of the LIS and the application's generated data, the LIS is a critical twenty four hour a day, seven day per week application. Analysts on the Lab IT/Surgery team are responsible for keeping this system running and monitoring performance.

PATROL Enterprise Manager from BMC software was implemented in 2002 by the healthcare system, providing a central interface for monitoring multiple applications, with email notification and text page alerting of IT staff regarding system issues. PATROL is used to monitor the "health" of various systems and their corresponding hardware components. Currently, the LIS is the most extensively monitored application using PATROL at the healthcare system. PATROL software monitors the LIS application, Unix OS and Oracle database components.

BACKGROUND

PATROL Overview

Historically, the LIS has been monitored by Lab IT/Surgery analysts using manual methods with only one or two knowledgeable users relying on PATROL. Training with PATROL has been self motivated with few written procedures and minimal formal training. Frequently when alerts were first received, analysts had little knowledge of what the alert meant or the corrective action required.

PATROL is a systems, applications, and event management tool for database and system administrators. It provides an object-oriented graphical workspace where you can view the status of every vital resource in the distributed environment you are managing [6]. This particular healthcare system monitors approximately 440 hardware devices (Unix computer systems, etc), and 130 production software applications [7] using PATROL in the region. PATROL monitors these computers and applications and can attempt to recover issues according to defined parameters or escalate warnings for human intervention. Escalations are displayed as either warnings or alerts in the graphical interface. PATROL consists of multiple components, but there are four integral pieces that impact this project and are utilized to monitor the LIS:

- PATROL Agent
- PATROL Knowledge Module (KM)
- PATROL Event Manager (PEM)
- PATROL Operator Console

PATROL Agent

A PATROL agent is installed on each monitored computer. The Agent is the piece of software responsible for monitoring and interacting with both hardware and applications. On startup, the Agent loads specified parameters and commands supplied by the knowledge module. Information is also logged by this component which is viewed by the PATROL Operator Console. The Agent also receives commands from users via the console to interact with the monitored system.

PATROL Knowledge Module

The KM can be envisioned as a program that was written to tell the Agent what to do. This program tells the Agent parameters to monitor (e.g.: computer central processing time), when to take action after a parameter reaches a certain value (e.g.: central processing time is 95% for 10 minutes) and what action to take (e.g.: send an alert to the console). One reason PATROL has high potential for use with the LIS is the vendor uses the same software to monitor their in-house applications. The vendor has developed a collection of BMC PATROL Knowledge Modules (KM) to proactively monitor and prevent issues affecting the availability of the LIS system. These KMs effectively extend PATROL's "predict and prevent" approach to the heart of a LIS environment [8]. The following "key" features are included in the LIS KM:

- Monitors LIS server definitions and instances (processes within the application that perform work).
- Monitors MQ series queues and shared services queues (where messages are stored as they are passed between servers).
- Monitors operations jobs (predefined tasks that run daily such as printing chart distributions or generating charges for billing)
- Monitors interfaces (how orders/results/ADT messages pass between the LIS, laboratory instruments and other PHS applications)

PATROL Event Manger

PEM acts as an interface allowing users to interact with events generated by the PATROL Agent. PEM integrates with the PATROL console allowing actions on events. Here, automatic recovery actions can be evoked or text pages (PEM alerts) may be generated bringing the attention of analysts to system issues (e.g.: send the alert to an analyst pager or close out the alert).

PATROL Operator Console:

The PATROL Operator Console is the component utilized by analysts on the Lab/Surgery team to monitor the LIS. The console is a graphical interface that represents various monitored parameters and their current state. Common tasks performed from the console include:

- Choosing hardware platforms and applications to monitor
- Managing computers and applications by manipulating the Agent and KM.
- Retrieving historical data

- Initiating commands to attempt system recovery.

Figure 1 represents the interaction of the Agent, KM, PEM and Operators Console on a monitored application.

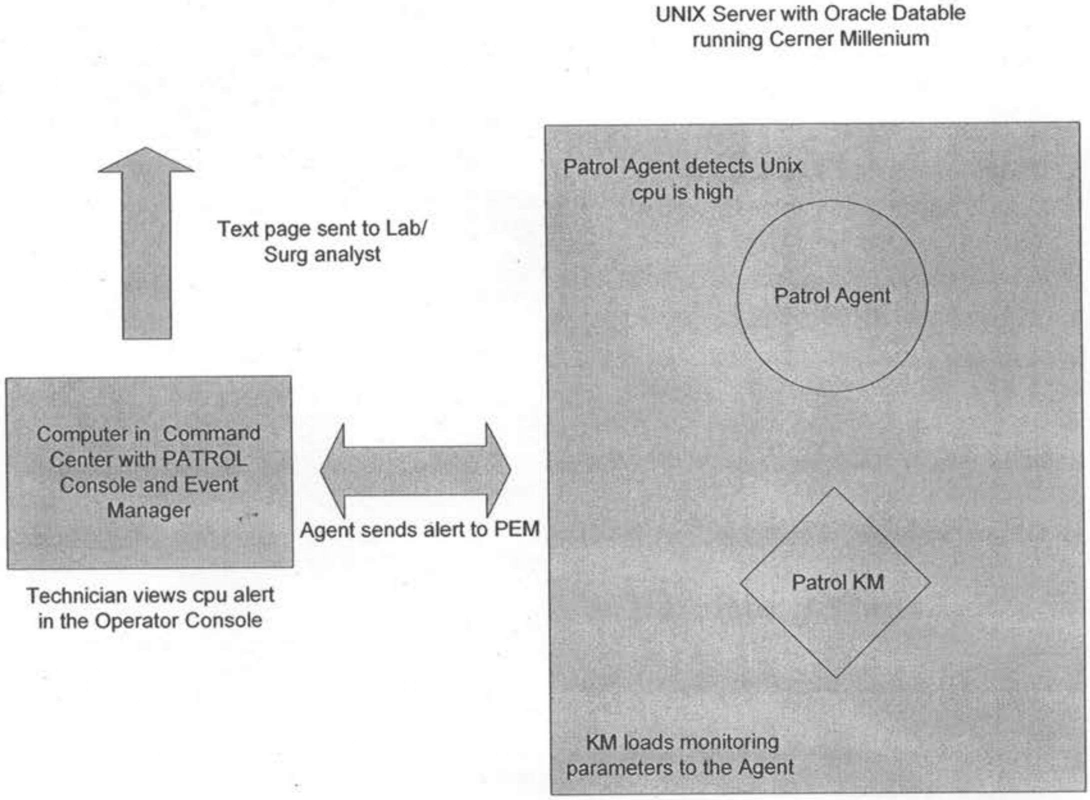


Figure 1 representation of PATROL components and interaction

STUDY OBJECTIVE

The objective of this study is to review enterprise monitoring for the LIS at the healthcare system by analyzing current enterprise monitoring alerts in place and the workflow of staff from the healthcare systems troubleshooting database. Current company procedures relating to enterprise monitoring will also be reviewed for staff training. These reviews will result in a discussion for improvements of the use of enterprise monitoring for the LIS and corresponding hardware.

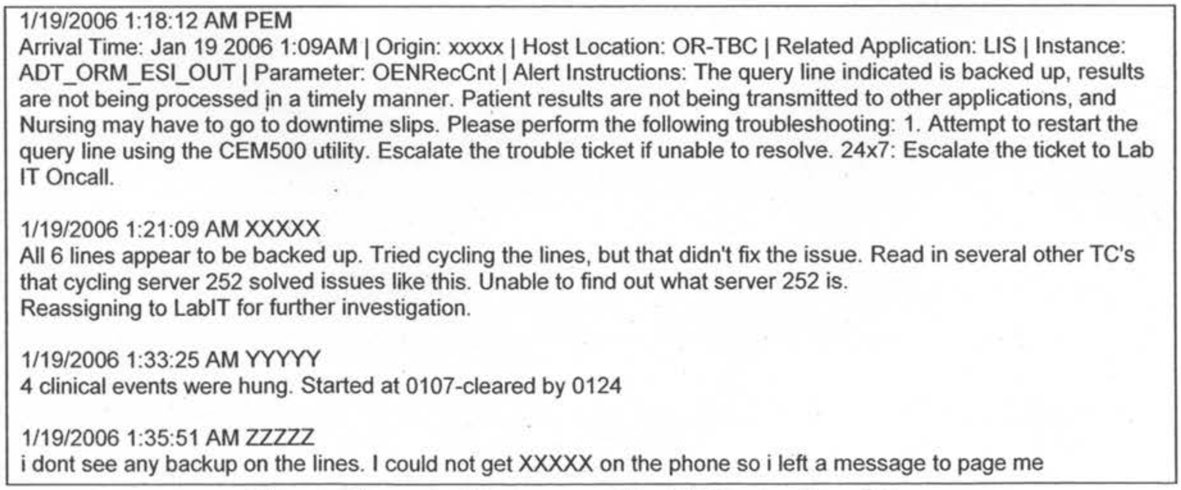
MATERIALS AND METHODS

Three data sources were utilized to address the study objective:

- A troubleshooting database
- The PATROL Console and database
- Current organization procedures

Troubleshooting Database

Historical data points towards staff workflow when alerts are received, displaying response time and time taken to resolve the issue. The troubleshooting database is an application used to record information system issues and the trouble steps followed to bring about problem resolution. Alerts from PATROL requiring action by IT staff automatically generate an entry in the troubleshooting database. Reviewing historical entries in this database will provide insight into actions generated by PATROL alerts, general length of time for resolution and steps taken by staff (See figure 2).



1/19/2006 1:18:12 AM PEM
Arrival Time: Jan 19 2006 1:09AM | Origin: xxxxx | Host Location: OR-TBC | Related Application: LIS | Instance: ADT_ORM_ESI_OUT | Parameter: OENRecCnt | Alert Instructions: The query line indicated is backed up, results are not being processed in a timely manner. Patient results are not being transmitted to other applications, and Nursing may have to go to downtime slips. Please perform the following troubleshooting: 1. Attempt to restart the query line using the CEM500 utility. Escalate the trouble ticket if unable to resolve. 24x7: Escalate the ticket to Lab IT Oncall.

1/19/2006 1:21:09 AM XXXXX
All 6 lines appear to be backed up. Tried cycling the lines, but that didn't fix the issue. Read in several other TC's that cycling server 252 solved issues like this. Unable to find out what server 252 is.
Reassigning to LabIT for further investigation.

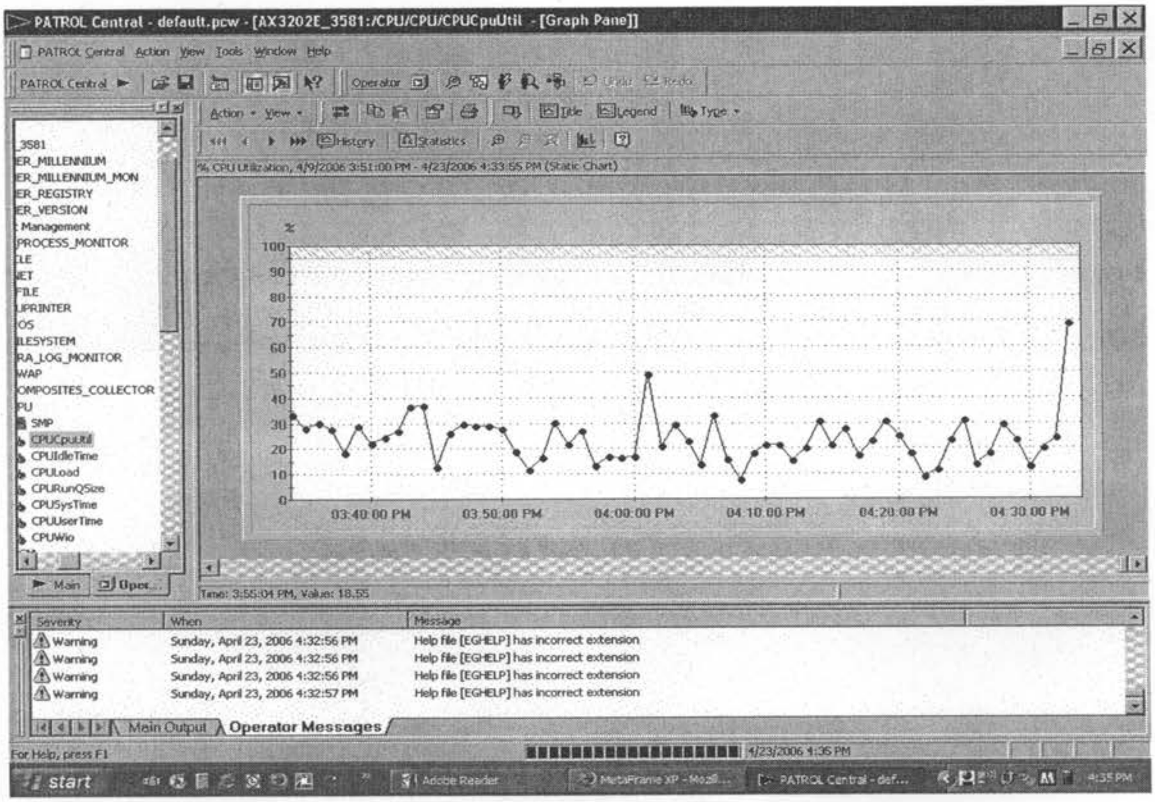
1/19/2006 1:33:25 AM YYYYY
4 clinical events were hung. Started at 0107-cleared by 0124

1/19/2006 1:35:51 AM ZZZZZ
i dont see any backup on the lines. I could not get XXXXX on the phone so i left a message to page me

Figure 2 sample ticket generated by PEM and staff workflow

PATROL Console

The PATROL console provides a central point for enterprise monitoring and viewing current alert thresholds. The console can be used to review LIS features that are monitored and the thresholds to generate alarms (See figure 3). Many of these thresholds start at default values and need to be evaluated for the LIS placed within different healthcare organizations.


PATROL Central - default.pcw - [AX3202E_3581:/CPU/CPU/CPUCpuUtil - [Graph Pane]]

% CPU Utilization, 4/9/2006 3:51:00 PM - 4/23/2006 4:33:55 PM (Static Chart)

Time: 3:55:04 PM, Value: 18.55

Severity	When	Message
Warning	Sunday, April 23, 2006 4:32:56 PM	Help file [EGHELP] has incorrect extension
Warning	Sunday, April 23, 2006 4:32:56 PM	Help file [EGHELP] has incorrect extension
Warning	Sunday, April 23, 2006 4:32:56 PM	Help file [EGHELP] has incorrect extension
Warning	Sunday, April 23, 2006 4:32:57 PM	Help file [EGHELP] has incorrect extension

Figure 3
PATROL console and alert threshold for
system cpu usage

Procedures

Finally, staff training is an important step needed to promote acceptance of enterprise monitoring to IT personnel and facilitate application use. Reviewing current Lab IT team procedures will allow training and staff resources to be evaluated. Currently written procedures for analysts using BMC PATROL have not been widely implemented. The Lab IT team has relied on self motivation and online help files contained within the PATROL application for staff training. The quantity and quality of current procedures will provide direction for the amount of detail needed in the operational documentation.

RESULTS

Patrol Review

PEM alerts were pulled from the PATROL database for review. PATROL monitoring had thirteen categories of PEM alerts for the LIS defined when this study started. PEM alerts are sent to defined recipients notifying individuals or groups that action is needed. Table 1 presents the thirteen categories of defined alerts along with characteristics pertaining to each category:

Parameter Name	**Event Description**	**Impact**	**Alert Recipient**	**Priority**	**Alert Message**
SRV_InstStatus	Monitors LIS "servers" stuck in a stopping or stopped state	Delay in application processing of results and orders	Lab IT/Surgery	Urgent	LIS server instance [object] is hung
OENOutCycle	Monitors the LIS query line to identify if messages are being sent outbound.	No charges will go out of the LIS.	Lab IT	High	LIS [object] line is not receiving messages
OENInCycle	Monitors the LIS query line to identify if messages are being No charges will go into the LIS. received.	No charges will go into the LIS.	Lab IT	High	LIS [object] line is not receiving messages
PROCActivePr ocessCount	Monitors the LIS ChartServer processes.	Clinical staff will not receive any lab reports.	Command Center Lab IT ortbclabitPATROL @xxxxxorg	Urgent	The LIS ChartServer process is not running
FSCapacity	Monitors filesystems used by the LIS application	LIS application will not function correctly if filesystems are full.	Lab IT	High	LIS filesys [object] is [parameterValue]% full
Cont. next page					

PROCStatus	Monitors the LIS ChartServer processes.	Clinical staff will not receive any lab reports.	Command Center Lab IT ortbclabitPATROL@xxxxx.org	High	The LIS ChartServer process is not running
ServiceStatus	Monitors the LIS FaxServer processes.	Clinical staff will not receive any lab reports.	Command Center Lab IT ortbclabitPATROL@xxxxx.org	High	The LIS FaxServer process is not running
TEXTFound	Monitors for Dr. Watson errors on the LIS FaxServer	Clinical staff will not receive any lab reports.	Command Center Lab IT ortbclabitPATROL@xxxxx.org	Urgent	A Dr. Watson error occurred on the LIS FaxServer
SRV_Deficit	Monitors the status of selected LIS "servers". Refer to HNAM_ServerList.txt for the list of monitored servers.	No inbound processing of lab requests, all facilities.	Command Center Lab IT 5036049480;ortbclabitPATROL@xxxxx.org	High	xxxxx server [object] is down
JOUNorPendAge	Monitors critical application transaction pending time for several key processes.	Lack of access to time sensitive clinical information.	Lab IT	High	Transaction pending time exceeded for [object]
CMCqmFsiEsoTr1Cnt	Monitors if there are greater than 100 records unprocessed on the cqm_fsieso_tr_1 table.	Results and Orders are not crossing to foreign systems from the LIS.	Lab IT	High	There are [parameterValue] records unprocessed on the cqm_fsieso_tr_1 table.
FSCapacity	Monitors filesystems used by the LIS Oracle instances.	Unable to use laboratory system to process orders, results, and charges.	Prod Sys-DBA	Moderate	LIS db fs [object] is [parameterValue]% full
CPUCpuUtil	This parameter displays the percentage of CPU utilization. CPU utilization is calculated by adding user time and system time.	The server is being heavily utilized, most likely application performance is degraded.	Lab IT	High	Sustained CPU utilization at [parameterValue]%

Table 1 Defined PEM alerts for the LIS

Reviewing these thirteen categories pointed out issues in four areas:

- Obsolete alert instructions
- Unclear alert messages
- Obsolete monitoring functionality
- Underutilized or misdirected alerts

Obsolete alert instructions:

OENInCycle (time since last record was received) and OENOutCycle (time since last record was processed) are used to monitor the time between HL7 messages entering and leaving the LIS interfaces to and from foreign systems. The event description for these parameters refers to messages not passing through the query lines. The query lines were used to view results stored in the LIS from the patient care application. The patient care application was a widely utilized application when the LIS was first implemented but use has dropped as it is replaced by different applications. Both the impact and instructions point to billing messages not entering or leaving the LIS. The more immediate concern is this same parameter monitors inbound/outbound ADT, order and result messages for the LIS interfaces. Monitoring billing is a high priority, but runs only once per day and does not need immediate resolution. Issues with the LIS interface directly impact patient care and frequently result in critical calls requiring staff to use downtime procedures for laboratory orders and results. The instructions within this alert should be changed to reflect the importance of the LIS interfaces.

Unclear alert messages

SRV_Deficit monitors the number of servers that are present. The alert text indicates “The LIS server XXX is down” and relays instructions to restart the server in the LIS application. This information has the potential to be misleading since many LIS servers may have more than one instance running to process messages. A better message would be “There is a deficit in the number of LIS Server XXX instances” and then refer staff to count the “actual” server instances to compare to the number defined to run.

Obsolete LIS functionality

The initial implementation of the LIS used a system called the message bus and journal to track messages as they are moved through out the system for processing. With the upgrade to a newer version of the LIS in 2004, the journal was rendered obsolete and replaced by MQSeries for message transactions from IBM. Unfortunately, this alert parameter was never replaced by its MQ series equivalent. With this parameter no longer in place, reaction to delays in processing have the potential to impact patient care and system uptime.

Underutilized or misdirected alerts:

Finally, the PATROL console pointed out an important parameter that should be assigned a PEM alert to Lab IT. SWPTotSwapUsedPercent measures the system’s total swap space that is in use and is displayed in the PATROL console. This is important because a system’s cpu can only run programs that are in memory. When the memory on a system runs low, processes can be written to disk freeing up memory. When a process is written to disk the time needed to load that process back into memory can significantly impact system performance. Over time, LIS processes cache memory and can cause the system to start swapping. There should be a PEM alert in place to notify a Lab IT/Surgery analyst when swapping reaches a certain level.

Troubleshooting Database Review

PEM alerts generate a ticket in the troubleshooting database which documents staff action to resolve the alert. These tickets can be assigned different priorities and response messages depending on the analyst submitting the alert request. Priorities fall within four categories:

- Critical – 15 minute response time
- Urgent – 30 minute response time
- High – 2 hour response time
- Moderate – 24 hour response time

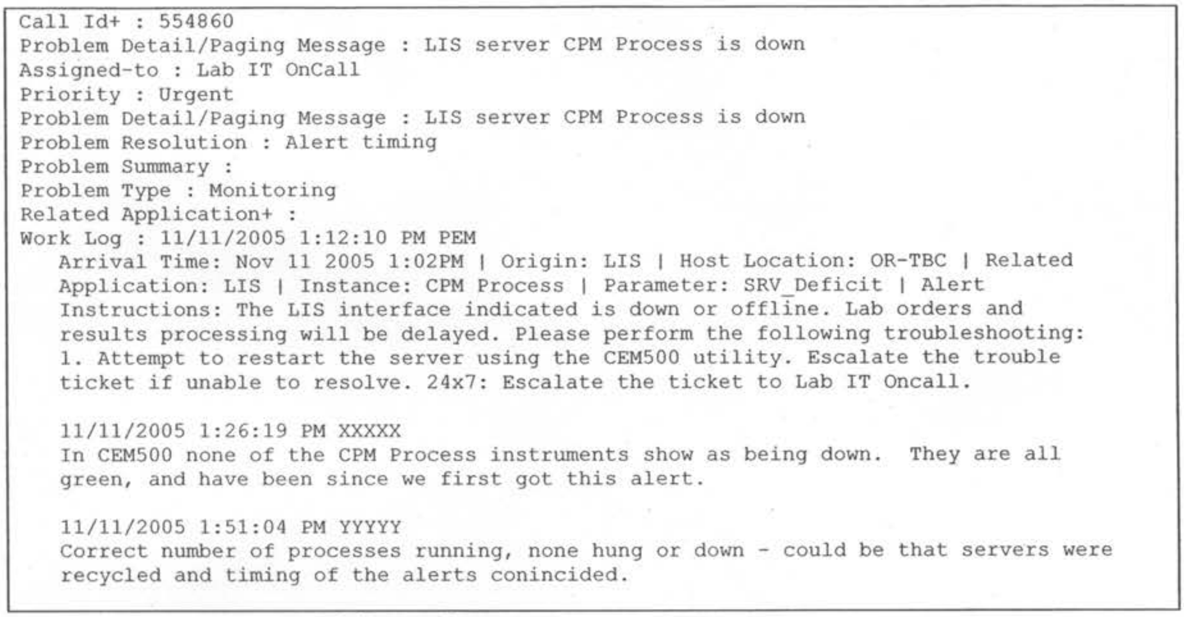
Troubleshooting database tickets generated by PEM alerts were pulled from a period of October 15th, 2005 through April 9th, 2006. Total tickets broken down by the priority assigned are displayed in Table 2:

Total Tickets	Critical	Urgent	High
351	1	137	151

Table 2 PEM troubleshooting database tickets by priority

The tickets displayed in Table 2 were reviewed for staff workflow and possible improvements in the alert message or actions for resolution.

Within the LIS the Operations Jobs are used to schedule activities for system management and clinical events. One of the system management activities that run out of operations is the cycling of LIS server instances to free up memory. Out of 350 urgent or high priority tickets, 19 were the result of alerts caused by operations jobs or scheduled system events. See Figure 4 for example workflow.



```
Call Id+ : 554860
Problem Detail/Paging Message : LIS server CPM Process is down
Assigned-to : Lab IT OnCall
Priority : Urgent
Problem Detail/Paging Message : LIS server CPM Process is down
Problem Resolution : Alert timing
Problem Summary :
Problem Type : Monitoring
Related Application+ :
Work Log : 11/11/2005 1:12:10 PM PEM
   Arrival Time: Nov 11 2005 1:02PM | Origin: LIS | Host Location: OR-TBC | Related
   Application: LIS | Instance: CPM Process | Parameter: SRV_Deficit | Alert
   Instructions: The LIS interface indicated is down or offline. Lab orders and
   results processing will be delayed. Please perform the following troubleshooting:
   1. Attempt to restart the server using the CEM500 utility. Escalate the trouble
   ticket if unable to resolve. 24x7: Escalate the ticket to Lab IT Oncall.

   11/11/2005 1:26:19 PM XXXXX
   In CEM500 none of the CPM Process instruments show as being down.  They are all
   green, and have been since we first got this alert.

   11/11/2005 1:51:04 PM YYYYY
   Correct number of processes running, none hung or down - could be that servers were
   recycled and timing of the alerts conincided.
```

Figure 4 PEM alert generated by an Ops Job

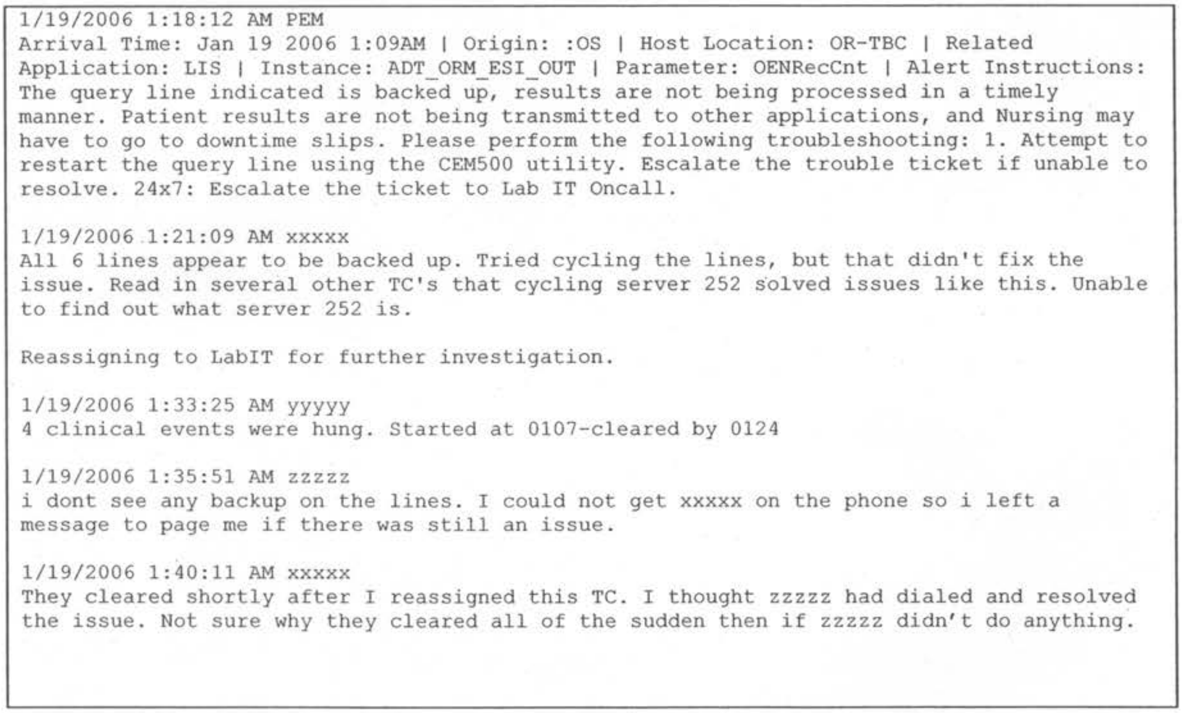
Figure 4 also shows an example of the SRV_Deficit PEM. PATROL detected one of the eighteen instances of the CPM Process server was down but the message indicates the entire server is down.

Frequently generated alerts that became routine also had an effect on staff monitoring. For example, high cpu utilization on the LIS system has been a frequent ongoing problem. Twenty eight alerts occurred during this six month period due to cpu utilization. At one time, the alert threshold was lowered to five minutes instead of ten minutes to monitor a system issue. Staff became non-responsive to these alerts because of their frequency, often not investigating why the alert was generated as seen below:

```
Call Id+ :
Problem Detail/Paging Message : Sustained CPU utilization at
99.90%
Assigned-to : Lab IT OnCall
Priority : High
Problem Detail/Paging Message : Sustained CPU utilization at
99.90%
```

```
Problem Resolution : CPU usage is fine now
```

PEM alerts requested by the Lab IT team have the functionality to be sent to multiple recipients. This is a useful feature which ensures that all the appropriate individuals are informed of system issues, but can result in some confusion. Figure 5 displays the confusion that may occur when more than one person responds to a text page attempting to correct the issue.



```
1/19/2006 1:18:12 AM PEM
Arrival Time: Jan 19 2006 1:09AM | Origin: :OS | Host Location: OR-TBC | Related
Application: LIS | Instance: ADT_ORM_ESI_OUT | Parameter: OENRecCnt | Alert Instructions:
The query line indicated is backed up, results are not being processed in a timely
manner. Patient results are not being transmitted to other applications, and Nursing may
have to go to downtime slips. Please perform the following troubleshooting: 1. Attempt to
restart the query line using the CEM500 utility. Escalate the trouble ticket if unable to
resolve. 24x7: Escalate the ticket to Lab IT Oncall.

1/19/2006 1:21:09 AM xxxxx
All 6 lines appear to be backed up. Tried cycling the lines, but that didn't fix the
issue. Read in several other TC's that cycling server 252 solved issues like this. Unable
to find out what server 252 is.

Reassigning to LabIT for further investigation.

1/19/2006 1:33:25 AM yyyyy
4 clinical events were hung. Started at 0107-cleared by 0124

1/19/2006 1:35:51 AM zzzzz
i dont see any backup on the lines. I could not get xxxxx on the phone so i left a
message to page me if there was still an issue.

1/19/2006 1:40:11 AM xxxxx
They cleared shortly after I reassigned this TC. I thought zzzzz had dialed and resolved
the issue. Not sure why they cleared all of the sudden then if zzzzz didn't do anything.
```

Figure 5 staff workflow on a trouble call involving multiple recipients

Most PEM alerts are first sent to the control center. The control center is staffed 24 hours a day and technicians there try to resolve the issue by following the instructions within the PEM alert. There was one instance in the six month period when a critical call was delayed within the control center. Staff was already working on an issue and did not have time to investigate or escalate to Lab IT for PEM alerts on the inbound interfaces. The problem involved ADT and order messages not crossing from the patient care system. A Lab IT analyst noticed the problem in the console, but inbound interface messages were already backed up fifteen minutes. The PEM alert is set to generate a page when no inbound messages have crossed the interface line in five minutes. As a result of the interface issue, facilities using the LIS had to go to downtime procedures for laboratory orders. This issue could be prevented through bypassing the command center for high impact parameters and sending these directly to the Lab IT on call.

The troubleshooting database contained a ticket where an interface issue that was prolonged due to the JOUNorPendAge parameter that was not replaced by the MQseries equivalent after being rendered obsolete. The call initiated when results and orders were not processing outbound to foreign patient care systems. The initial issue was resolved after troubleshooting with the vendor for approximately forty five minutes. During troubleshooting, one of the processes within the LIS application was thrown out of sync and intermittent outbound messages failed to process outbound. This was finally resolved and approximately three thousand messages were released after multiple users throughout the day reported issues with lab order/result statuses not updating outside of the LIS. If MQseries would have been setup post upgrade to replace the JOUNorPendAge parameter an alert would have been generated hours earlier bringing the situation to the attention of an analyst.

Organization Procedure Review

The Lab IT team decided to increase reliance on PATROL after upgrading both the LIS and PATROL in October 2005. Most analysts on the team were not familiar with this tool and responsibility for monitoring had fallen on one or two knowledgeable staff. Staff was introduced to the PATROL console during a twenty minute group training session. They were encouraged to use PATROL and then come to one of the "super-users" with questions. The PATROL LIS KM provides an extensive online help file. Some areas of the KM are not covered as extensively as others for example searching help on the OENOutCycle returns an extensive help section while searching on CMCqmFsiEsoTr1 returns nothing. Some staff members are not comfortable with online help versus written procedures resulting in the online health being under utilized. There currently are no written policies or procedures for the Lab IT team and the use of PATROL.

DISCUSSION

When PATROL was first implemented, its primary use was related to hardware monitoring. Over time, PATROL was increasingly used to monitor applications and their various components. The idea for this review was inspired by the expectation for Lab IT analysts to become familiar with and rely on PATROL as part of the troubleshooting and monitoring processes.

PEM alerts were chosen for review because this is how analysts receive initial notification of an issue if they are not actively using the PATROL console. While message instructions were incomplete for OENInCycle and OENOutcyle, the alerts did generate correctly. SRV_Deficit also has an issue with alert instructions which results in hampering staff in determining the correct number of instances that should be running for a LIS server. One roadblock to clarifying PEM messages is the limited number of characters that can be included and sent to a text pager. An idea to work around this limitation is the creation of a support document on a shared folder that refers to a section of the document that addresses the specific PEM alert. This would result in additional work and maintenance to keep the support document up to date with current alerts and responses. Staff on the Lab IT team and control center would need training on accessing and then correlating instructions for the correct PEM. See the example message in Figure 6

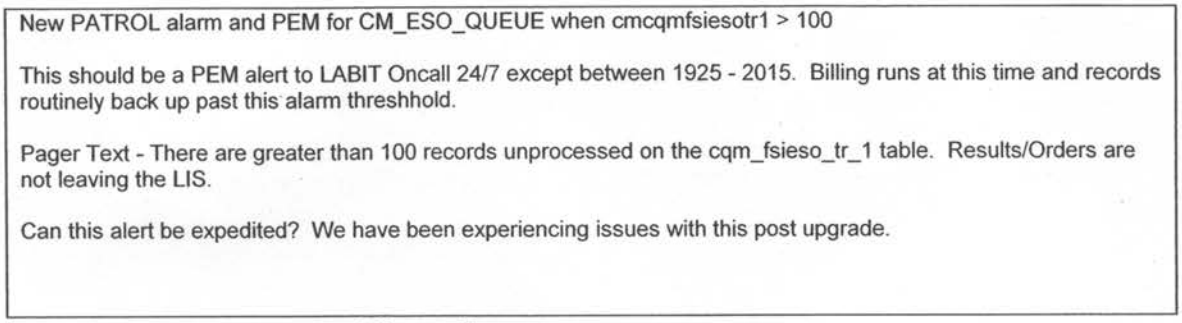
```
LIS server instance [object] is hung – Refer to entry 5 the LIS PATROL doc located \\phsornt\171\lis\shared\PATROL
```

Figure 6 alert referencing new support documents

This could help the Lab IT team which only takes call every 10 days, but the benefit to the control center would be questionable. Staff currently monitors or performs tasks for hardware/applications in multiple states. Would the control center have time to use a document in the troubleshooting process, or should the PEM just be escalated on to Lab IT? The results section has one issue when response was delayed because control center staff was too busy to escalate the PEM alerts for inbound interface (TC567481) indicating this could be the case. This PEM alert document would help during periods of staff turnover and training within the Lab IT team.

Lab IT increasingly relies on PATROL for monitoring, but this can result in a false sense of security if a monitor becomes obsolete or is missed altogether. The LIS upgrade rendered the message journal obsolete and the PEM for swap usage are instances when this occurred. The enterprise monitoring team keeps a list of PEM alerts, but the list is not easily accessible by other analysts and it can take several weeks to request a copy of the alerts. There has also been confusion about when an alert should generate. For example the alert for cpu utilization needed to be modified several times until it was directed to the correct recipient and generated at the proper parameters. Requests for PEM alerts are sent via work requests in Remedy with no defined format as shown in

Figure 7.



New PATROL alarm and PEM for CM_ESO_QUEUE when cmcqmfsiesotr1 > 100

This should be a PEM alert to LABIT Oncall 24/7 except between 1925 - 2015. Billing runs at this time and records routinely back up past this alarm threshhold.

Pager Text - There are greater than 100 records unprocessed on the cqm_fsieso_tr_1 table. Results/Orders are not leaving the LIS.

Can this alert be expedited? We have been experiencing issues with this post upgrade.

Figure 7 submission for a new PEM alert

Both requesting analysts and the enterprise monitoring team would benefit from a documentation defining the specific parameters requested for a PEM. A trial form has been created and stored on the Lab IT team shared drive containing the following fields:

Parameter to be monitored: for CM_ESO_QUEUE when cmcqmfsiesotr1 > 100

Priority: High

Who should receive the alert: LABIT Oncall 24/7 except between 1925 - 2015

Warning threshold:

Alert threshold: cmcqmfsiesotr1 > 100

Pager Text: There are greater than 100 records unprocessed on the cqm_fsieso_tr_1 table. Results/Orders are not leaving Cerner.

Documentation would also resolve discrepancies between what was submitted by Lab IT and the created PEM alert. This stored documentation would also offer a consolidated space to review functionality that may change after application upgrades.

Reviewing troubleshooting tickets brought into focus two areas which should improve monitoring. Staff had become so used to cpu PEM alerts, that there was little documented resolution or troubleshooting. Parameters for the PEM generation may need to be increased so that when the alert is generated staff will know there is an issue that needs to be addressed. Possibly the cpu drops back down to acceptable levels when Lab IT investigates or impact is not felt by the end users rendering the current parameter useless. If the current parameters are significant, then trouble tickets should be audited and reviewed with staff as to why the issue was not investigated.

There are also a significant number of alerts generated by scheduled jobs in the LIS application. An example is an Ops job cycling servers causing an alert for SRV_Instate or SRV_Deficit. Lab IT is working with an enterprise monitoring analyst to fine-tune these alerts so they are no longer generated during scheduled time slots. Within PATROL, time periods can be defined when a PEM alert will not be generated to the designated recipient. This was shown in the previous figure when the alert for cmcqmfsiesotr1 > 100 should go to LABIT Oncall 24/7 except between 1925 - 2015. Billing runs at this time and records routinely back up past this alarm threshold.

Procedural documentation was an area that was almost completely missing when it comes to monitoring the LIS with PATROL. Lab IT staff has increased its use of PATROL, but greater benefits could be realized with supporting documentation. The online help provides some guidance, but is written for the LIS KM in a general setting and not directed towards how the LIS is used at the healthcare organization. Work has already started on training documentation addressing a future PATROL console upgrade. This document will also be used to introduce new staff to PATROL and the applications capabilities. The initial expectation of a short training session and then follow up by staff with specific questions may be a good idea. However, follow up by staff has been poor and there can be some confusion when unfamiliar alerts are received. Also, without documentation the Lab IT team ends up relying on one or two members for assistance.

With the amount of turnover in today's workforce, there is always the potential that knowledgeable staff members will move on resulting in the resources no longer being available.

During the course of this project, additional areas of functionality were pointed out that may increase PATROL's benefit. It is possible to send a PEM alert to a mailbox each time one is generated. A PATROL LabIT mailbox has been created and receives alerts for designated PEMs. These e-mails can be used to follow trends or track the frequency of PEM alerts (see appendix A). These e-mail alerts have been used to track errors on the LIS chart servers as they were moved from old hardware to new hardware/technology. Enterprise monitoring has also moved to facilitate the use of PATROL. Development has started on an internal website to access statistics, alerts and procedures (appendix B). They have also started gathering hardware statistics for key applications so performance can be reviewed over months or years at a time.

CONCLUSION

This study sought to review enterprise monitoring for the LIS at the healthcare system. Data was analyzed from current enterprise monitoring alerts in place, the workflow of staff from the healthcare systems troubleshooting database and current organization procedures. Several issues were identified which could be categorized as:

- Obsolete alert instructions or unclear messages
- Obsolete system functionality
- Underutilized or misdirected alerts
- Lack of staff training documentation

This all has the potential to increase time to resolution and hinder the prevention of system issues. As PATROL is increasingly utilized, various improvements have been made as displayed in the appendices. Training for new staff has increased but no formal documentation has been prepared. Plans are also in progress to replace old monitors and simplify current alerts.

REFERENCES

[1] Ganesh, Variar, Monitoring Metrics, www.intelligententerprise.com, Aug 2001

[2] Linda T. Cohn, Janet M Corrigan and Molla S. Donaldson *To Err is Human: Building a Better Health System.* IOM, 2000

[3] www.leapfroggroup.org, last accessed 01/12/2006

[4] http://www.hhs.gov/healthit/mission.html, last accessed 01/13/2006

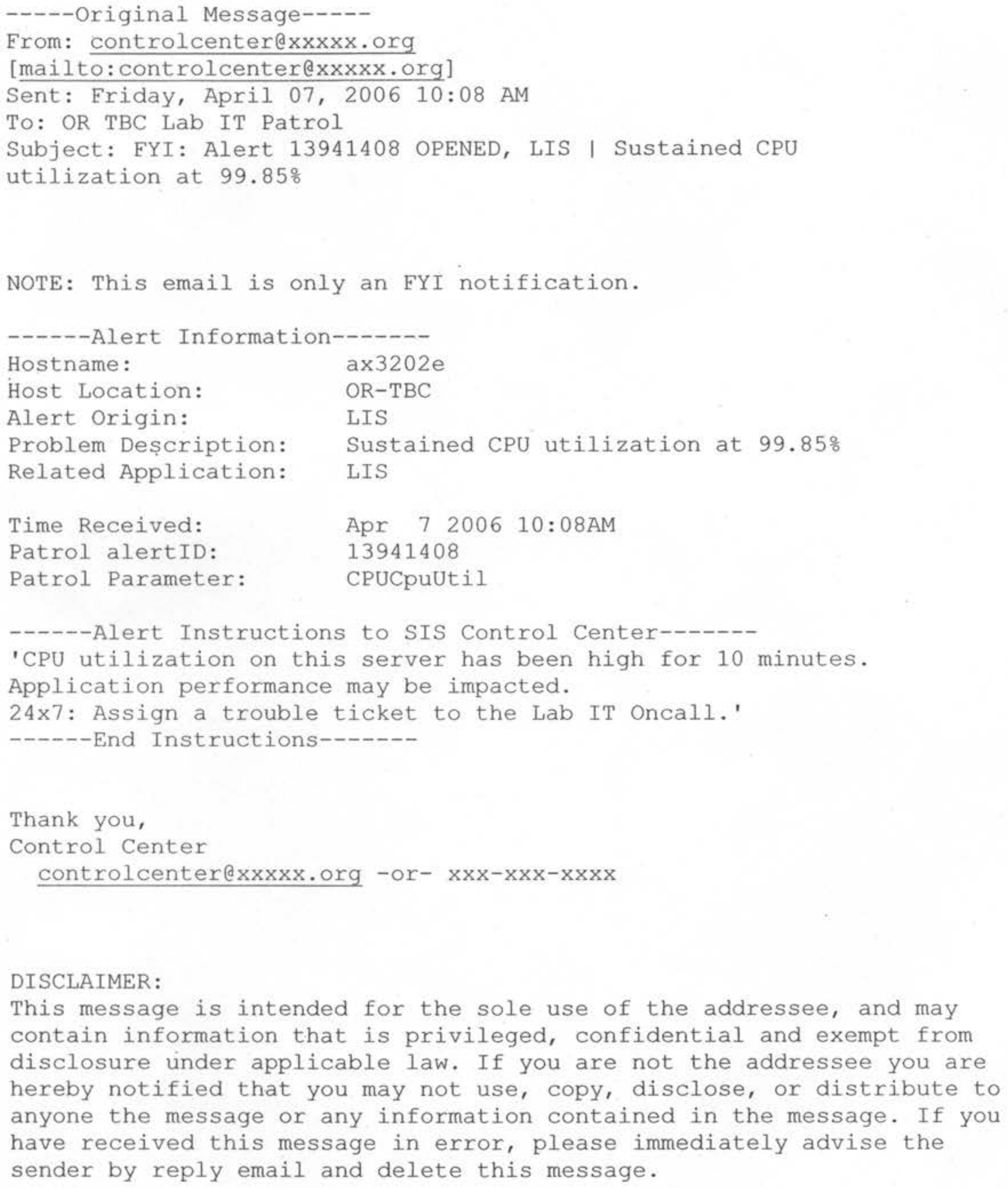
[5] Anonymous, *BMC PATROL and xxx*, e-poster

[6] BMC Software, *PATROL Console for Microsoft Windows, Volume 1—Understanding the Basics of PATROL User Guide*, BMC Software, March 14 2005

[7] http://appmon.html, last accessed 04/06/06

[8] Anonymous, *LIS Knowledge Module*, Online Help File

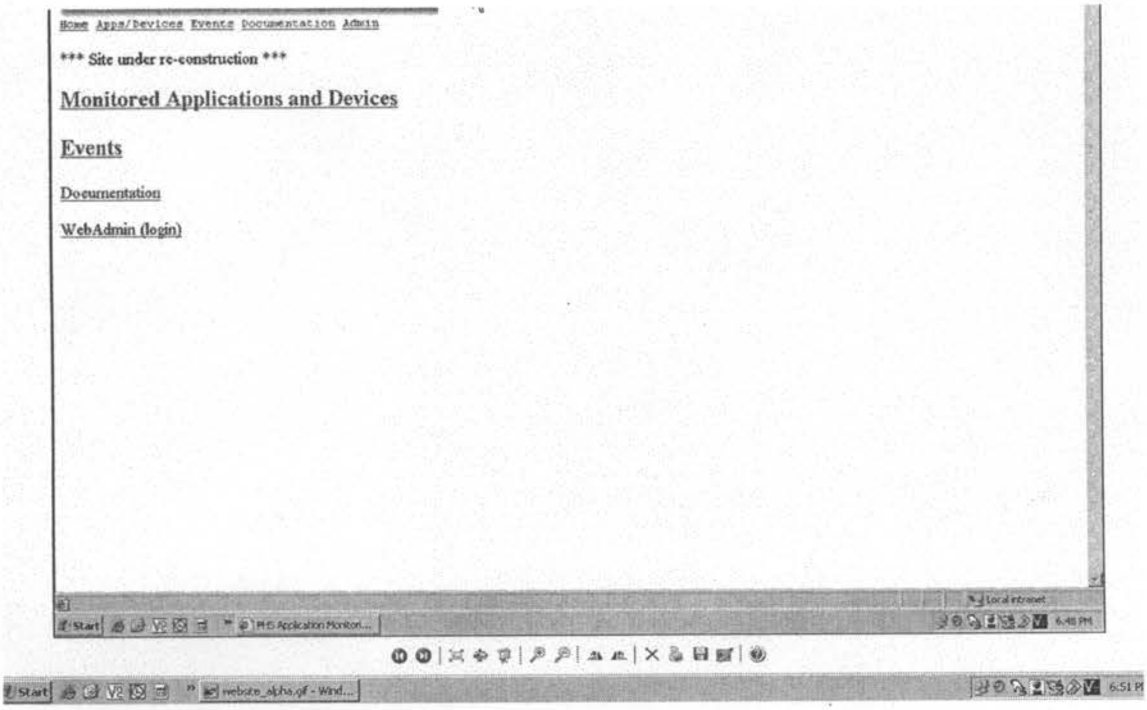
Appendix A



```
-----Original Message-----
From: controlcenter@xxxxx.org
[mailto:controlcenter@xxxxx.org]
Sent: Friday, April 07, 2006 10:08 AM
To: OR TBC Lab IT Patrol
Subject: FYI: Alert 13941408 OPENED, LIS | Sustained CPU
utilization at 99.85%


NOTE: This email is only an FYI notification.

------Alert Information-------
Hostname:                ax3202e
Host Location:           OR-TBC
Alert Origin:            LIS
Problem Description:     Sustained CPU utilization at 99.85%
Related Application:     LIS

Time Received:           Apr  7 2006 10:08AM
Patrol alertID:          13941408
Patrol Parameter:        CPUCpuUtil

------Alert Instructions to SIS Control Center-------
'CPU utilization on this server has been high for 10 minutes.
Application performance may be impacted.
24x7: Assign a trouble ticket to the Lab IT Oncall.'
------End Instructions-------


Thank you,
Control Center
  controlcenter@xxxxx.org -or- xxx-xxx-xxxx


DISCLAIMER:
This message is intended for the sole use of the addressee, and may
contain information that is privileged, confidential and exempt from
disclosure under applicable law. If you are not the addressee you are
hereby notified that you may not use, copy, disclose, or distribute to
anyone the message or any information contained in the message. If you
have received this message in error, please immediately advise the
sender by reply email and delete this message.
```

E-mail notification that can be used for alert trending

Appendix B



Home Apps/Devices Events Documentation Admin

*** Site under re-construction ***

Monitored Applications and Devices

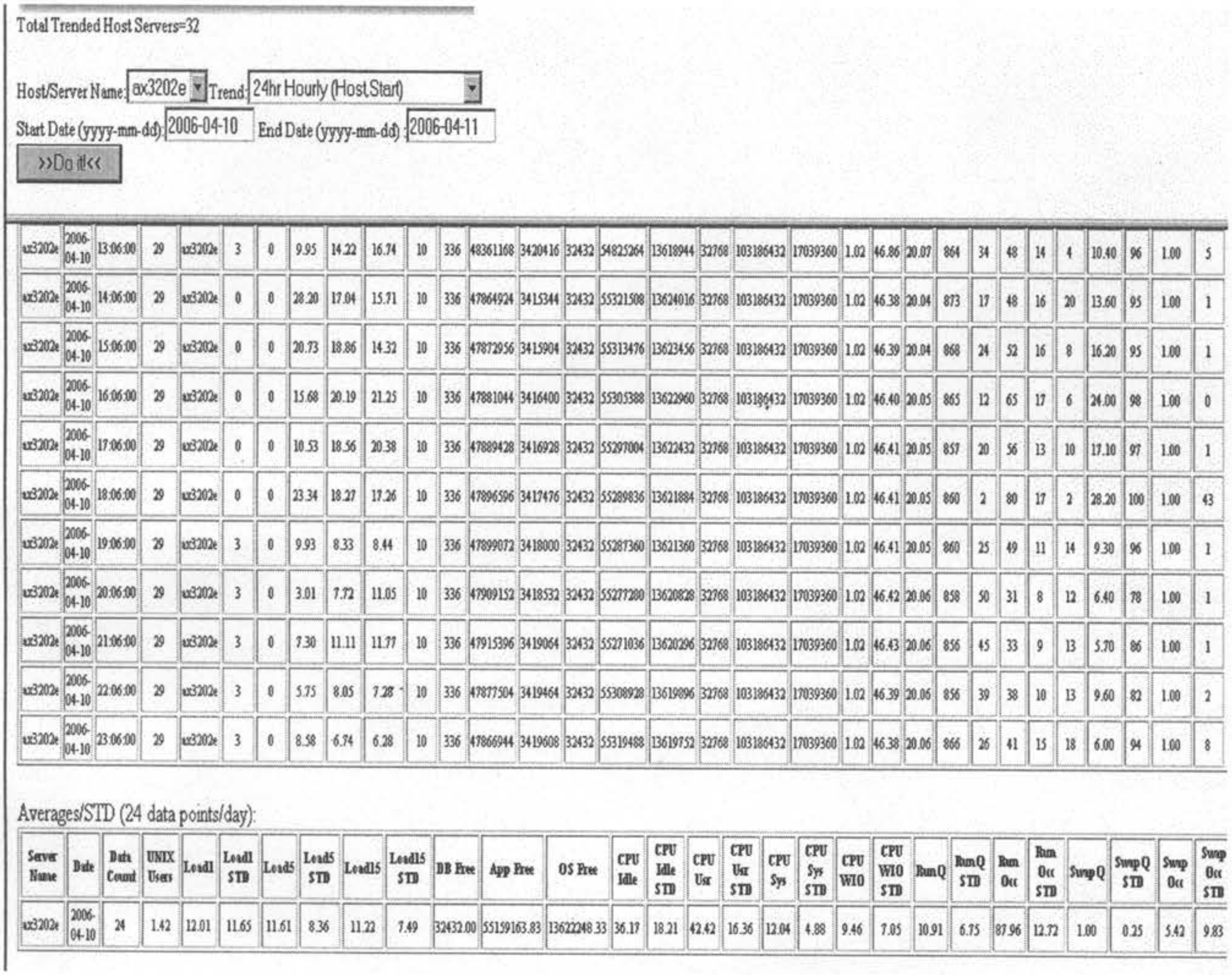
Events

Documentation

WebAdmin (login)

Website currently under construction by the enterprise monitoring team.

Appendix C



Total Trended Host Servers=32

Host/Server Name: ax3202e ▾ Trend: 24hr Hourly (HostStart) ▾

Start Date (yyyy-mm-dd): 2006-04-10 End Date (yyyy-mm-dd): 2006-04-11

ax3202e	2006-04-10	13:06:00	29	ax3202e	3	0	9.95	14.22	16.74	10	336	48361168	3420416	32432	54825264	13618944	32768	103186432	17039360	1.02	46.86	20.07	864	34	48	14	4	10.40	96	1.00	5
ax3202e	2006-04-10	14:06:00	29	ax3202e	0	0	28.20	17.04	15.71	10	336	47864924	3415344	32432	55321508	13624016	32768	103186432	17039360	1.02	46.38	20.04	873	17	48	16	20	13.60	95	1.00	1
ax3202e	2006-04-10	15:06:00	29	ax3202e	0	0	20.73	18.86	14.32	10	336	47872956	3415904	32432	55313476	13623456	32768	103186432	17039360	1.02	46.39	20.04	868	24	52	16	8	16.20	95	1.00	1
ax3202e	2006-04-10	16:06:00	29	ax3202e	0	0	15.68	20.19	21.25	10	336	47881044	3416400	32432	55305388	13622960	32768	103186432	17039360	1.02	46.40	20.05	865	12	65	17	6	24.00	98	1.00	0
ax3202e	2006-04-10	17:06:00	29	ax3202e	0	0	10.53	18.56	20.38	10	336	47889428	3416928	32432	55297004	13622432	32768	103186432	17039360	1.02	46.41	20.05	857	20	56	13	10	17.10	97	1.00	1
ax3202e	2006-04-10	18:06:00	29	ax3202e	0	0	23.34	18.27	17.26	10	336	47896596	3417476	32432	55289836	13621884	32768	103186432	17039360	1.02	46.41	20.05	860	2	80	17	2	28.20	100	1.00	43
ax3202e	2006-04-10	19:06:00	29	ax3202e	3	0	9.93	8.33	8.44	10	336	47899072	3418000	32432	55287360	13621360	32768	103186432	17039360	1.02	46.41	20.05	860	25	49	11	14	9.30	96	1.00	1
ax3202e	2006-04-10	20:06:00	29	ax3202e	3	0	3.01	7.72	11.05	10	336	47909152	3418532	32432	55277280	13620828	32768	103186432	17039360	1.02	46.42	20.06	858	50	31	8	12	6.40	78	1.00	1
ax3202e	2006-04-10	21:06:00	29	ax3202e	3	0	7.30	11.11	11.77	10	336	47915396	3419064	32432	55271036	13620296	32768	103186432	17039360	1.02	46.43	20.06	856	45	33	9	13	5.70	86	1.00	1
ax3202e	2006-04-10	22:06:00	29	ax3202e	3	0	5.75	8.05	7.28	10	336	47877504	3419464	32432	55308928	13619896	32768	103186432	17039360	1.02	46.39	20.06	856	39	38	10	13	9.60	82	1.00	2
ax3202e	2006-04-10	23:06:00	29	ax3202e	3	0	8.58	6.74	6.28	10	336	47866944	3419608	32432	55319488	13619752	32768	103186432	17039360	1.02	46.38	20.06	866	26	41	15	18	6.00	94	1.00	8

Averages/STD (24 data points/day):

Server Name	Date	Data Count	UNIX Users	Load1	Load1 STD	Load5	Load5 STD	Load15	Load15 STD	DB Free	App Free	OS Free	CPU Idle	CPU Idle STD	CPU Usr	CPU Usr STD	CPU Sys	CPU Sys STD	CPU WIO	CPU WIO STD	RunQ	RunQ STD	Run Occ	Run Occ STD	SwapQ	SwapQ STD	Swap Occ	Swap Occ STD
ax3202e	2006-04-10	24	1.42	12.01	11.65	11.61	8.36	11.22	7.49	32432.00	55159163.83	13622248.33	36.17	18.21	42.42	16.36	12.04	4.88	9.46	7.05	10.91	6.75	87.96	12.72	1.00	0.25	5.42	9.83

Apptrend website providing historical hardware performance statistics