Enterprise Systems Management

Daniel J. Oberst
Director of Enterprise Services

Charles A. Augustine
Manager, Enterprise Systems Management
Enterprise System Management

- Integrated approach to managing complex computing environment
- Consolidated view of enterprise systems
- Business process view of data and networking systems
- Scalable, Modular, Expandable
Why Enterprise System Management?

- Applications no longer monolithic
- Point solutions are inefficient and ineffective
- Need to adapt to rapidly changing environment
- Distribute knowledge and support
- Provide proactive, preemptive support
Mainframe World

Is the Purchasing System Running?
Distributed World

Is the Purchasing System Running?
Tivoli Overview - DJO 9/9/98

Help Desk

Global Enterprise Manager (GEM)

Network

HW/SW inventory

Tivoli Enterprise Console (TEC)

Tivoli Framework

Availability:
  - Unix
  - NT/Novell
  - Oracle
  - Mail
  - Peoplesoft

Managed Nodes

Job Scheduling
  - Maestro

Output Management
  - Destiny

NetView
Issues for Higher Education

- Economic justification is difficult
- Different time constraints than business
- Few colleges or universities use ESM systems
- Many locally developed tools
Perceived Benefits

- Distributed, Object-Oriented Design
  - Robust - Data and programs distributed
    - CORBA-based
  - Modular - Simplifies adding components
  - Efficient - Minimizes network traffic
  - Cross-platform - Runs on multiple Unix, NT
Perceived Benefits (cont.)

♦ Availability/Distributed Monitoring
  ♦ Alerts from network, servers, applications centrally monitored
  ♦ Modules for Oracle, mail, web
  ♦ Thresholds & correlations for preventative maintenance
  ♦ Statistical analysis and capacity planning
Bringing Tivoli to Princeton

- Princeton Partnership 2000 project
  - Convert applications to distributed platforms
  - Provide infrastructure comparable to mainframe

- Tivoli/Princeton partnership
  - Broad range of products
  - Evaluate applicability in higher education
Implementation Timeline

♦ June 1998 – Purchased Tivoli suite
♦ December 1998 – TWS (Maestro) in production
♦ Sept-Dec. 1998 – Framework/DM/TEC architecture planning and initial implementation
♦ April 1999 – Permanent ESM group formed
♦ Sept. 2000 – All production systems monitored
Reality Check

♦ Hard to do “in the background”
  ♦ Job Scheduling & Output Management First
♦ Prix Fix Menu *AND* À La Carte
  ♦ Inhouse Help Desk
  ♦ Dazel Output Management
  ♦ Inhouse monitoring plus Netview/Tivoli
♦ Hard sell to systems staff (NIH)
Tivoli at Princeton

- Monitoring
  - Distributed Monitoring
  - Enterprise Console
  - Manager for Oracle
  - Application Performance Monitoring
  - Web Services Manager
  - Netview

- Job Scheduling
  - Workload Scheduler

- Backup
  - Storage Manager
Tivoli at Princeton [cont.]

♦ Monitoring
  ♦ 130 Unix and NT hosts
  ♦ 11 applications

♦ Job Scheduling
  ♦ Schedule on 28 Unix and NT hosts
  ♦ Run 3500 jobs per week

♦ Backup
  ♦ 7000 clients (servers and desktop systems)
  ♦ Hundreds of terabytes of storage
Tivoli Monitoring Architecture

- Central configuration management
- Distributed monitoring agents
- Central event processing and correlation
- Adapters for events from other applications
- Easy to add locally developed monitors
Tivoli Monitoring Architecture

Distributed Monitoring

Local Processing
- Schedules
- Thresholds
- Responses

Application performance

Logfile monitoring

Configuration mgmt. and event processing

Custom monitoring

syslog
Application log
Implementation Process

- Identify users and requirements
- Plan what to monitor
- Select from supplied monitors
- Create custom monitors as needed
- Develop rules to process events
- Develop central notification process
Lessons Learned

♦ Do not underestimate time and staffing needs
  ♦ Dedicate staff, at least initially
♦ Do not try to replace existing tools
  ♦ Pick your battles: no Alaskan iceboxes
  ♦ Find your niche: (e.g. NT vs. Unix)
♦ Focus on application availability
  ♦ Top-down instead of bottom-up strategy
Why Did We Underestimate Effort?

♦ Tivoli is a toolkit, not a turnkey application
♦ Old habits are hard to change
♦ Tivoli has a steep learning curve
♦ After initial rollout, operation competes with development
Tivoli Monitoring Toolkit

- Distributed Monitoring infrastructure
  - Basic set of OS level monitors
  - Handful of application monitoring package
  - Tools for creating custom monitors
- Event processing engine
  - Minimal event processing rules
- Consider Tivoli Rapid Deployment Program
Development Strategies
Top Down vs. Bottom Up

- **Top down**
  - Applications first
  - Analyze problems
  - Application owners
  - Integrate existing monitors

- **Bottom up**
  - Infrastructure first
  - Synthesize state
  - Sys admins are users
  - Replace existing tools
Development Strategies
Pros and Cons

♦ Top down
  ♦ Models end user experience
  ♦ Adds value immediately
  ♦ More complex monitors
  ♦ Less sophisticated users

♦ Bottom up
  ♦ Models application dependencies
  ♦ Requires high reliability
  ♦ Needs admin buy-in
  ♦ Replaces existing tools
Next Steps at Princeton

♦ Monitor all Oracle databases
♦ Application response monitoring
♦ Feed events to the help desk
♦ Better filtering of transient errors
♦ Automated pager response
For More Information

♦ Princeton ESM web site
  ♦ [http://www.princeton.edu/~esm](http://www.princeton.edu/~esm)

♦ E-mail to
  ♦ oberst@princeton.edu
  ♦ augustin@princeton.edu

♦ Tivoli web site
Questions?