



| IBM Software Group

Automated Performance Management Using IBM Tivoli: Techniques And Best Practices

Session 12880

Tuesday, February 6th: 11:00 AM - 12:00 PM

Tivoli software



Agenda

- Why Integrated Automation?
- Where Automation?
- What is Automated Performance Management?
- Integrated Performance Automation
 - ▶ Tools, Tips, Techniques
 - ▶ Situations and Policies
- Examples And Best Practices
- Recommendations



Why Integrated Automation? Islands Of Automation Complicate Management

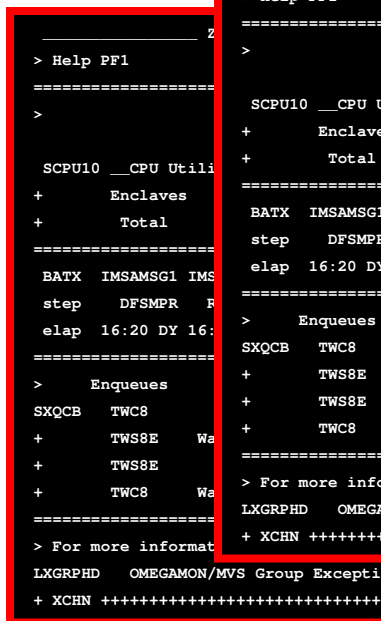
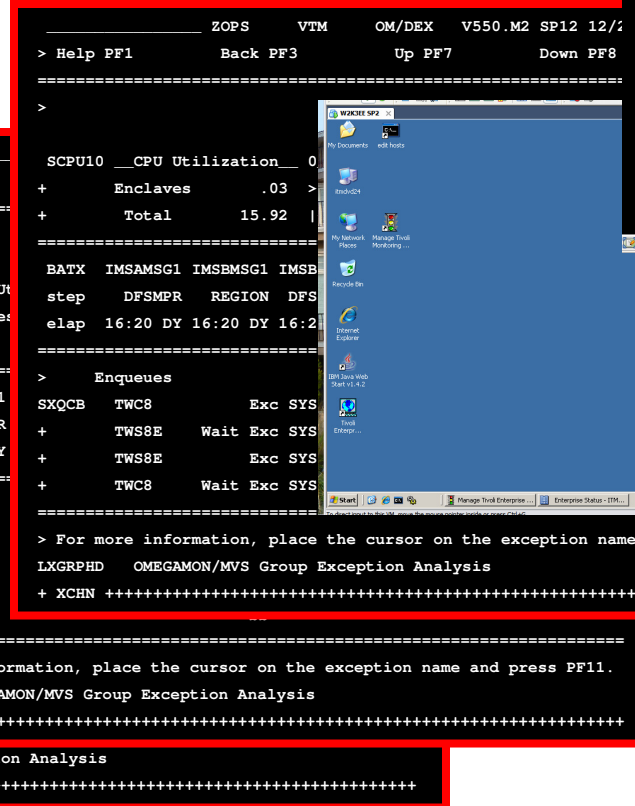
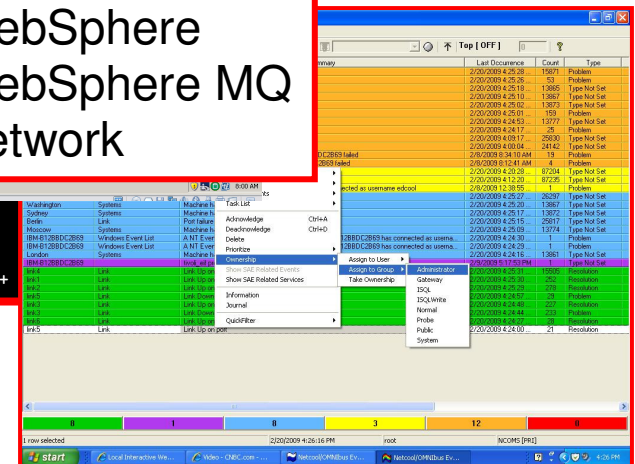
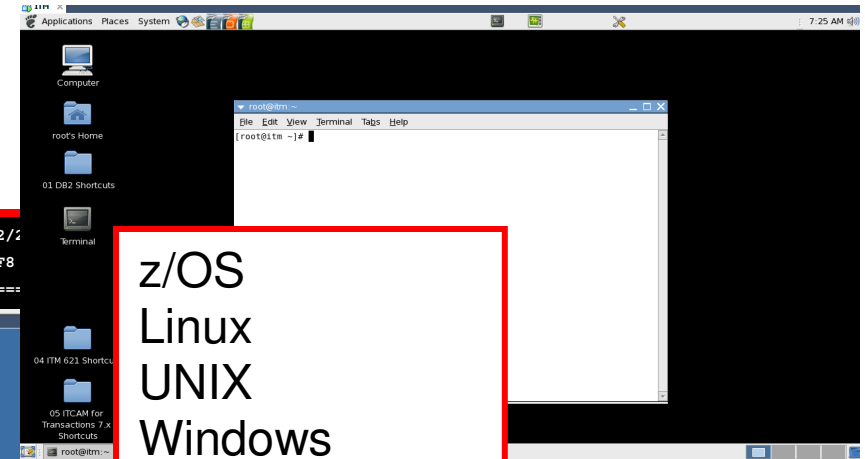
Potentially many consoles,
screens, and technologies to
monitor and manage

CICSC

DB2B

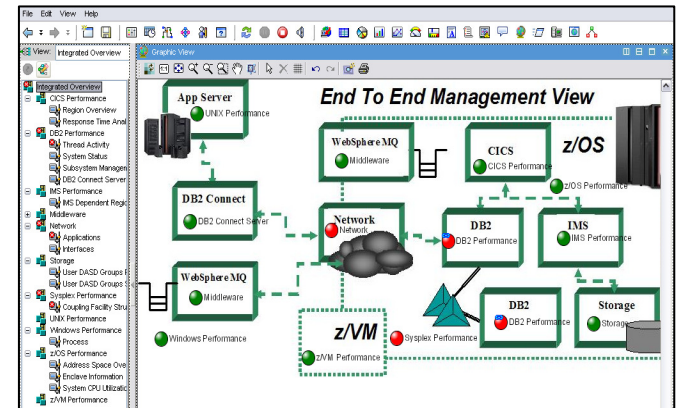
MVSA

z/OS
Linux
UNIX
Windows
Database
WebSphere
WebSphere MQ
Network



Automated Performance Management Addressing Islands Of Automation

- Many technical platforms, components and core technologies to manage
 - ▶ Often each with it's own group of Subject Matter Experts (SMEs)
 - ▶ Potentially with it's own set of management tools
- The problems
 - ▶ Complex SME tools with different User Interfaces
 - ▶ SME tools that do not integrate or share information
 - More difficult to navigate
 - More difficult to do problem identification, isolation, and resolution
 - ▶ More challenging to automate corrective actions without clearly defined integration



Recommendation – Where feasible pursue a more integrated approach

Where Automation?

Automation Many Occur At Many Levels

- Traditional z/OS console automation
 - ▶ Automated resource management
 - System start up and shut down
 - ▶ Console message management – message suppression
 - ▶ Resource and application management
 - Abend/failure management
 - Subsystem support management
 - WTORs - log management – archive management
- Automation within monitoring and analysis technologies
 - ▶ Command and corrective action capabilities within tools
 - ▶ Alerts and notifications
- Event/Network management
 - ▶ Alerts, notifications and corrective actions managed by the “Manager of Managers” – example Netcool OMNIbus

z/OS console

Address spaces
Messages
Resource status

Monitoring

Resource monitor
Analytics
Real time
History
Alerts – messages

Event Management

Event correlation
Notification
Correction



A Goal For Many Shops Make Systems Management More 'Proactive'

- In many shops systems management tends to be done 'ad hoc'
 - ▶ Some alert generation – varies by shop
 - Some shops very alert driven – many are not
 - ▶ Often notification consists of 'call the help desk'
- Many customers want to be more 'proactive'
 - ▶ Definition of proactive may vary
 - Proactive for some installations may mean more rapid alert and notification of technical and/or business application issues
 - Proactive for some installations may mean notification **prior** to the problem
 - Alert when utilization indicates a potential issue in the future
 - Alert when I'm within 90% of the wall
 - Proactive may mean an automated workaround or resolution

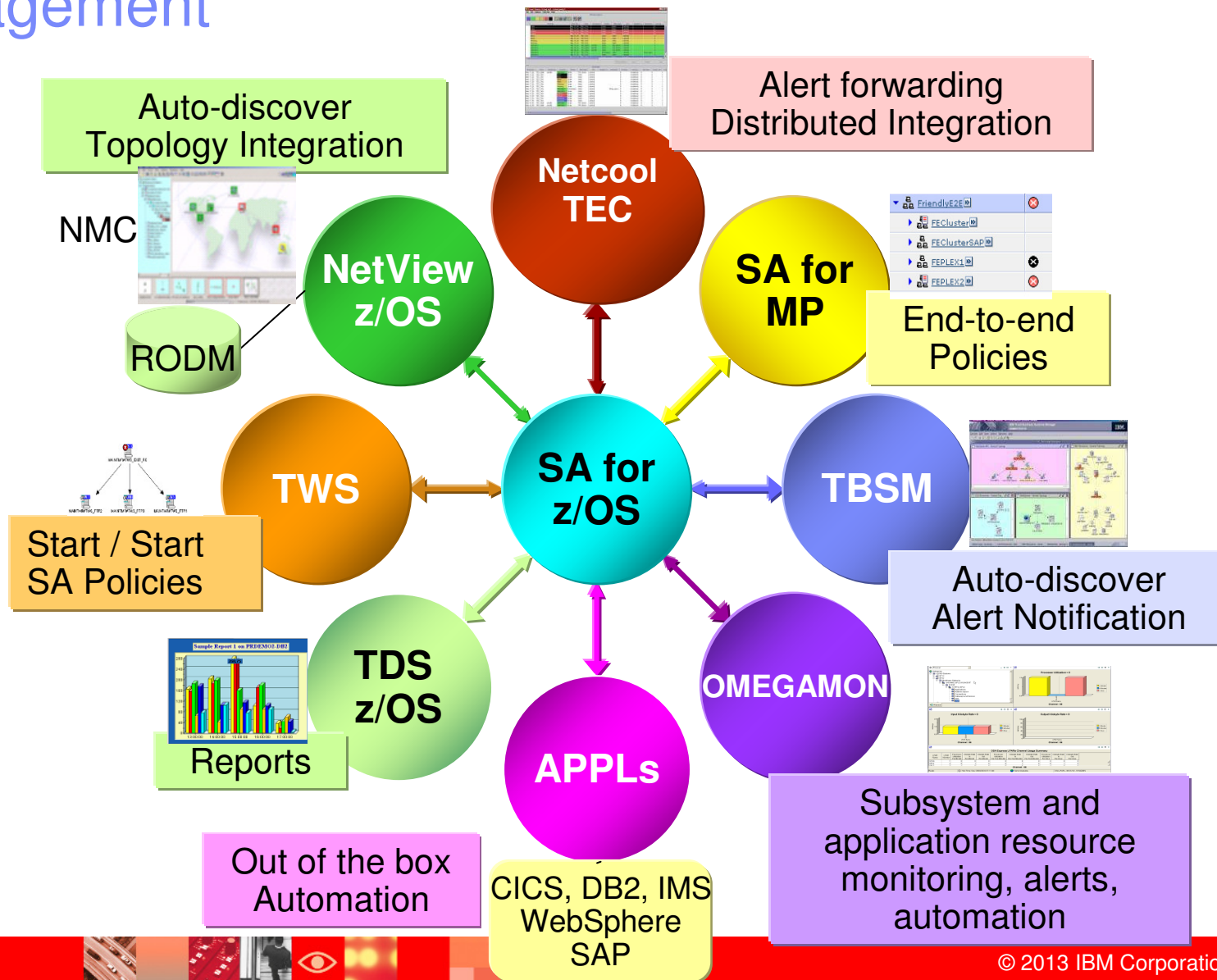


What Is Automated Performance Management?

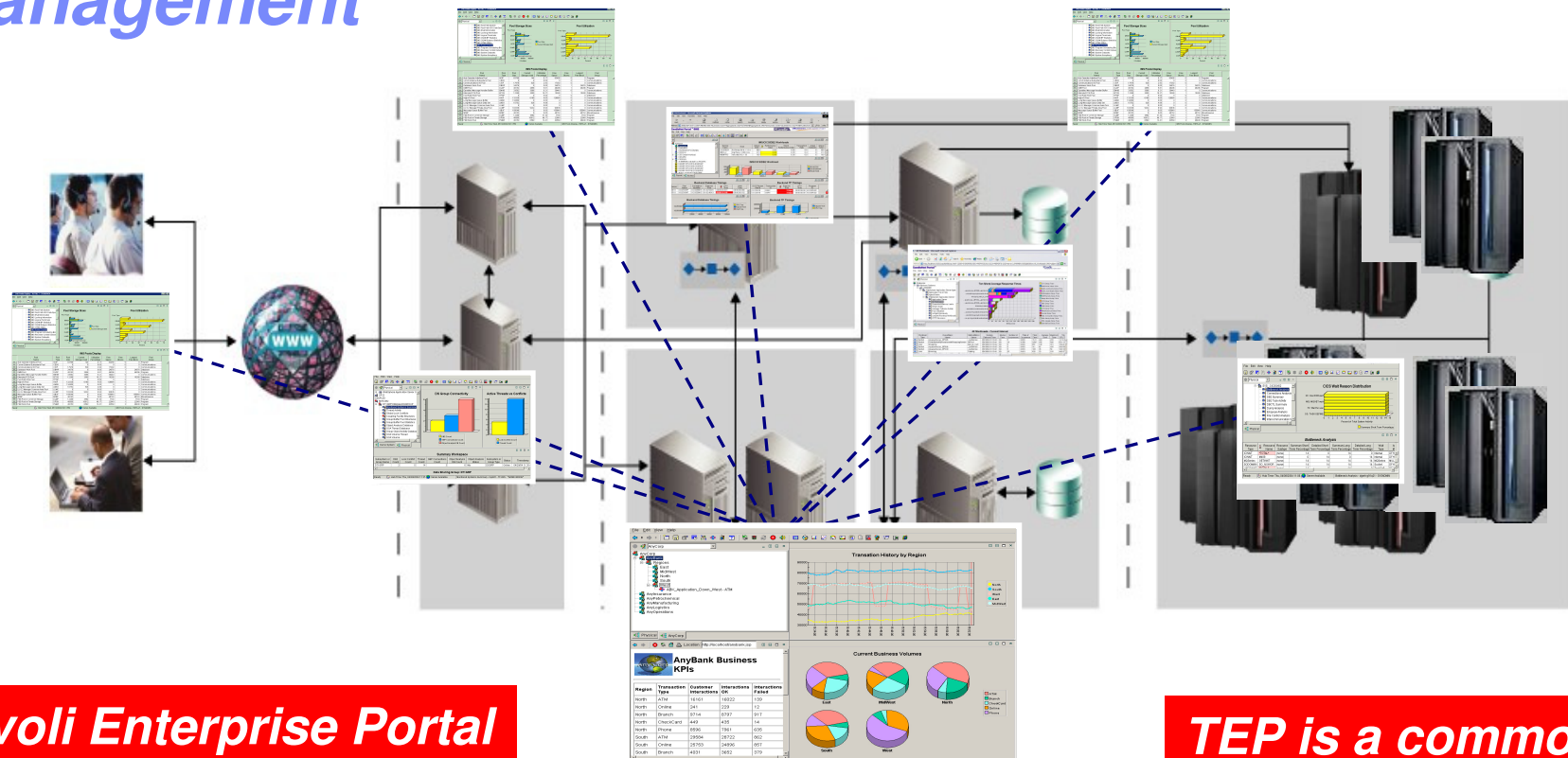
- Exploiting and leveraging the intrinsic monitoring and management capabilities of performance monitoring combined with event management and automation
 - ▶ Make automation more powerful and robust by incorporating performance metrics into automation routines
 - ▶ Incorporate information from the application and/or subsystem performance level
 - ▶ Incorporate systems and application knowledge of the staff into automation routines
- The benefits - Become more 'proactive'
 - ▶ Improved and more meaningful/timely alerts and notifications
 - ▶ Improved understanding of systems and systems management
 - ▶ Reduce the time for problem identification and isolation
 - ▶ Improve MTTR (mean time to resolution)
 - ▶ Where possible solve problems at machine speed



IBM System Automation Integrates With Monitoring And Management



Tivoli Enterprise Portal (The TEP) Integrated Performance, Availability, And Systems Management



Tivoli Enterprise Portal enables integrated alert and automation capabilities

Tivoli Enterprise Portal (TEP)

TEP is a common user interface for a variety of Tivoli solutions

Leverage The Integration Capabilities Of The Tivoli Enterprise Portal To Provide “Visualization”

The Portal enables customized views specific to installation requirements

Alerts by subsystem

Problem jobs as reported by Tivoli Workload Scheduler

Possible looping jobs and system CPU as monitored by OMEGAMON

Resource status as monitored by IBM SA

Important WTORs as monitored by IBM SA

Critical messages as monitored by IBM SA

Manual corrective actions

Hub Time: Thu, 01/31/2013 10:25 AM Server Available EW OPS View - dem17lnx.democentral.ibm.com - Ed Woods

Leverage Resource And Status Information From System Automation

The screenshot displays the Tivoli Enterprise Portal interface. The main window shows a 'Graphic View' with a 'CICS Status' and 'z/OS Status' visualization. A 'Properties - EW_OPS_View' dialog box is open, showing a 'Preview' of the 'Unavailable Task Status' table. Below the preview, there are 'Filters' and 'Data Snapshot' sections. A red box highlights the 'Unavailable Task Status' table and the 'Filters' section. A red text box is overlaid on the bottom left of the screenshot.

Unavailable Task Status

Resource Name	System	Observed Status	Desired Status	Automation Status	Automation Flag	Hold Flag	D
DEMO_CICS01	DEMOMVS3	SoftDown	Available	Idle	Yes	No	CICS R
DSNCSPAS	DEMOMVS	HardDown	Available	Idle	Yes	No	DB2 st
OM_IMSB	DEMOMVS	HardDown	Available	Idle	Yes	No	OM
DEMO_CBJ	DEMOMVS	Degraded	Available	Idle	Yes	No	Job
DEMO_CBS	DEMOMVS	Degraded	Available	Idle	Yes	No	Sys
DEMO_CPU	DEMOMVS	Degraded	Available	Idle	Yes	No	Lo

Filters

	Resource Name	System	Observed Status	Desired Status	Automation Status	Auto F
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2			!= Available	== Available		
3						
4						

Data Snapshot

Resource Name	System	Observed Status	Desired Status	Automation Status	Auto F
DEMOMVS	DEMOMVS	Available	Available	Idle	Yes
DEMOMVS2	DEMOMVS2	Available	Available	Idle	Yes
DEMOMVS3	DEMOMVS3	Available	Available	Idle	Yes
APPC	DEMOMVS	Available	Available	Idle	Yes
APPC	DEMOMVS2	Available	Available	Idle	Yes
APPC	DEMOMVS3	Available	Available	Idle	Yes

Important WTORs

Resource System	Severity	Reply ID	Message ID	Message Text
DEMOMVS	NORMAL	89	DSI802A	CNM16 REPLY WITH VALID NCCF SYSTEM OP
DEMOMVS	UNUSUAL	86	HWSC0000I	*IMS CONNECT READY* IMSDCON
DEMOMVS	UNUSUAL			
DEMOMVS	NORMAL			
DEMOMVS	NORMAL			
DEMOMVS	NORMAL			
DEMOMVS	UNUSUAL			
DEMOMVS	NORMAL	78	DFS996I	*IMS READY* IMSD
DEMOMVS	NORMAL	76	DFS996I	*IMS READY* IMSC
DEMOMVS	NORMAL	72	DFS996I	*IMS READY* IMSA
DEMOMVS2	NORMAL	36	DSI802A	CNM17 REPLY WITH VALID NCCF SYSTEM OP

IBM SA provides z/OS resource status, critical message, and outstanding WTOR information to the Tivoli Portal

Leverage Resource And Analysis Information From OMEGAMON Monitoring

The screenshot displays the OMEGAMON monitoring interface with several key components:

- Navigator:** A tree view on the left showing the hierarchy of monitored resources, including CICS Status, DB Status, Thread Activity, Network Status, Applications, Connections, z/OS Status, Address Space Overview, System CPU Usage, and Physical/Logical views.
- Graphic View:** A central area showing graphical representations of system status, including CICS Status and Operational Alerts.
- Potential Looping Task:** A table showing tasks that may be looping, with columns for Managed System, Job Name, Using CPU, Using IFA, Using z/IPS, CPU Loop Index, Active I/O, Queued I/O, Enqueue Wait, and Tape Mount. A red box highlights this table, and a red arrow points to it from the text "Example – track possible looping jobs".
- System CPU Usage:** A table showing CPU usage statistics for various managed systems, including Average CPU Percent, RMF MVS CPU Percent, RMF LPAR CPU Percent, Total TCB%, Total SRB%, Average IFA Percent, Average IFA on CP Percent, and Average z/IPS Percent.
- Unavailable Task Status:** A table showing the status of tasks that are unavailable, including Resource Name, System, and Severity.
- Important WTORs:** A table showing critical messages (WTORs) with columns for Resource System, Severity, Reply ID, and Description.

Example – track possible looping jobs

OMEGAMON provides detailed resource analysis at the z/OS operating system, component subsystem (CICS, IMS, DB2, Websphere, Websphere MQ), network, and application level

Tivoli Enterprise Portal

Performance Automation Integrated Within The Portal

- The Portal provides manual commands and corrections
 - ▶ 'Take Action' provides for manual command capability
 - ▶ Commands may be predefined
- The Portal enables automated commands and corrections
 - ▶ Implement machine speed corrective actions, issue alerts, and allow for later human intervention
 - ▶ Use for automated commands for dynamic subsystem management and 'tweaks' as the workload and system changes
 - ▶ Two core types of automated actions
 - **Situations** - Use for simple "fire and forget" type of scenarios
 - **Policies** – Use for more sophisticated performance automation scenarios



About Situations And Policies

- Situations are the building blocks of systems management logic in the Tivoli Enterprise Portal (TEP)
 - ▶ Situations may be used to highlight performance and availability problems within key operating systems, subsystems, and mission critical resources
 - ▶ Situation logic may be distributed to the agent (IRA architecture)
 - Situations typically run at the level of the agent (TEMA)
- Policies extend concepts established with situations and add additional functionality to the TEP
 - ▶ Situations remain the essential starting point
 - ▶ Policies add additional function and flexibility
 - ▶ Note - Policies run within the TEMS infrastructure



Situations - A Basic Example

Alert On DB2 Threads With More Than 'n' Getpages

The screenshot shows the 'Situations for - Detailed Thread Exception' dialog box. The left pane shows a tree view with 'Detailed Thread Exception' expanded, showing 'EVW_Threat', 'MVS DB2', 'KDP_V', and 'KDP_V'. The right pane has tabs for 'Formula', 'Distribution', 'Expert Advice', 'Action', and 'Until'. The 'Formula' tab is active, showing a 'Getpage Count' table with a formula '> 1000'. The 'Sampling interval' is set to '0 : 0 : 1 : 30'. The 'Sound' section has 'Enable critical.wav' checked. The 'State' section has 'Critical' selected and 'Run at startup' checked. Red arrows point from text boxes to these elements.

Start/stop situation

- Create New...
- Create Another...
- Start Situation
- Stop Situation
- Delete Situation
- Associate
- Dissociate

Distribution tab to specify where situation runs. Expert advice is customizable. Action tab to execute command.

Specify alert criteria. This may include one or multiple attribute criteria.

Specify sampling interval

Specify severity and whether to run at Omegamon startup

Situations

'Action' To Perform Commands And Corrections

Where command is executed

Action Selection
☒ System Command ☐ Universal Message

System Command
 LPG 'DB2 Thread Alert Message - &{DB2_Thread_Exceptions.Plan_Name} &{DB2_Thread_Exceptions.Authorization}

Attribute Substitution...

If the condition is true for more than one monitored item:
☒ Only take action on first item
☐ Take action on each item

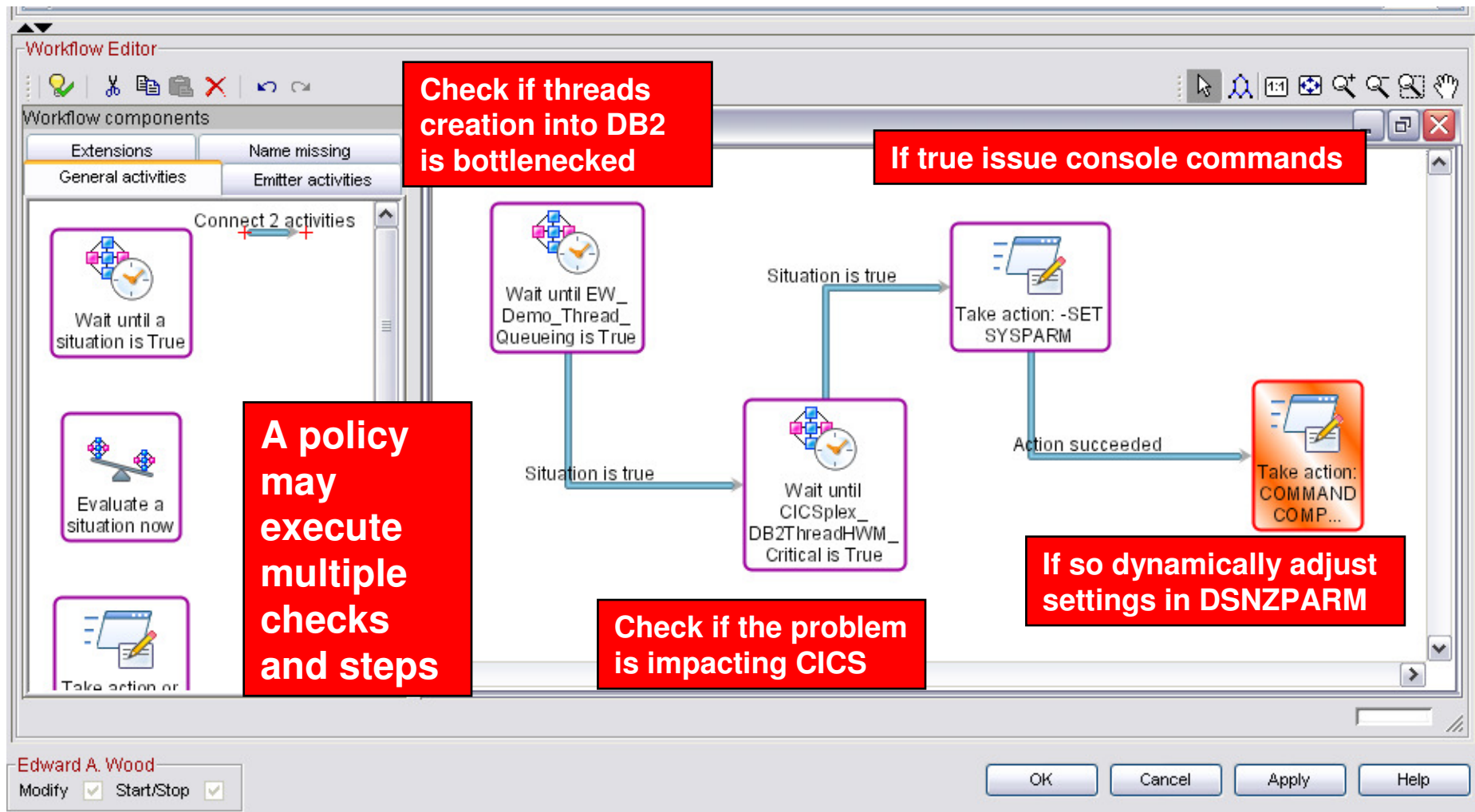
Where should the Action be executed (performed):
☐ Execute the Action at the Managed System (Agent)
☒ Execute the Action at the Managing System (TEMS)

If the condition stays true over multiple intervals:
☒ Don't take action twice in a row (wait until situation goes false then true again)
☐ Take action in each interval

System commands may be executed when the situation is true

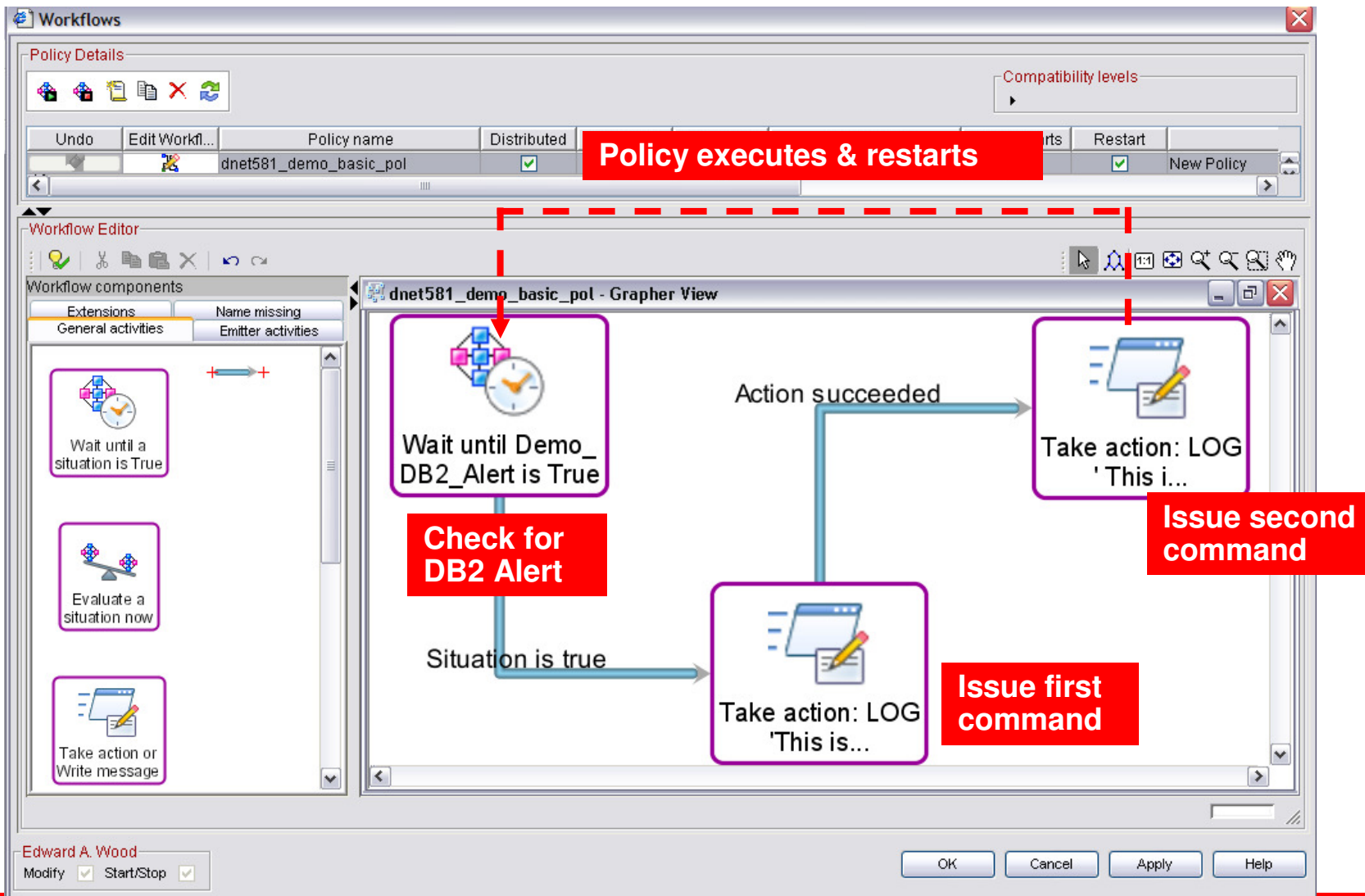
Examples of actions include:
 DB2 thread kill command
 Issuing messages to the console
 Any valid z/OS console command
 Issuing commands to drive notification

Policies Expand The Concept Of Automated Performance Management



Basic Policy - Example Scenario

Have A Situation Trigger Multiple Commands



Take Action Options Within A Policy

The screenshot displays the IBM Workflow Editor interface. On the left, the 'Workflow components' pane shows 'General activities' and 'Emitter activities'. The 'General activities' pane contains two icons: 'Wait until a situation is True' and 'Evaluate a situation now'. The 'Emitter activities' pane contains one icon: 'Wait until Demo_DB2_Alert is True'. The main workspace shows a diagram with a 'Wait until Demo_DB2_Alert is True' activity connected to a 'Situation is true' label. A red box highlights the 'Wait until Demo_DB2_Alert is True' activity. A red arrow points from this activity to the 'Action Settings' dialog box.

The 'Action Settings' dialog box is open, showing the 'System Command' tab. The 'System Command' field contains the text: `LOG 'This is a test message - DB2 message &WaitOnSituation1:DB2_Thread_Exceptions.Plan_Name'`. A red box highlights this field. A red arrow points from the 'Attribute substitution in the command line' text to the 'Attribute...' button. The 'If multiple data items are returned' section has the 'Only take action on the first item' radio button selected. The 'Where should the action be executed' section has the 'Execute the action at the TEMS' radio button selected.

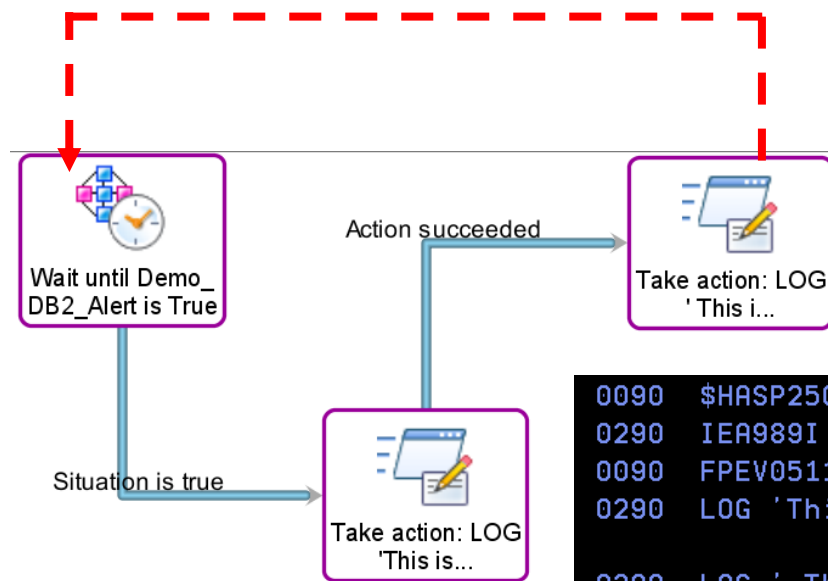
**Take the action at the agent
Which agent?
Where is the agent?**

**Take the action at the TEMS
Which TEMS?
Where is the TEMS?**

**Note - Where the action gets executed
dictates the appropriate type of action**

**Attribute substitution in
the command line**

Policy Command Execution



In the example the policy will:

Check the situation status
Execute the first command
Execute the second command
Restart

Note – The interval of the situation will have an impact on the duration of the policy

```

0090 $HASP250 DNET145 PURGED -- (JOB KEY WAS C1C5C854)
0290 IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=UNQ001 0010 0151.
0090 FPEV0511I DSNB HISTORY DATA SET WRAPPED, 4272 INTERVALS STORED
0290 LOG 'This is a test message - DB2 message ADHPLAN3'

0290 LOG ' This is a second test message'

0290
0290
0290 IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=UNQ001 0010 0151.
0290 LOGON
0290 LOG 'This is a test message - DB2 message ADHPLAN3'

0290 LOG ' This is a second test message'

0281 $HASP100 DNET581  ON TS0INRDR
0090 $HASP373 DNET581  STARTED
0090 IEF125I DNET581 - LOGGED ON - TIME=10.06.27
    
```

First command

Second command

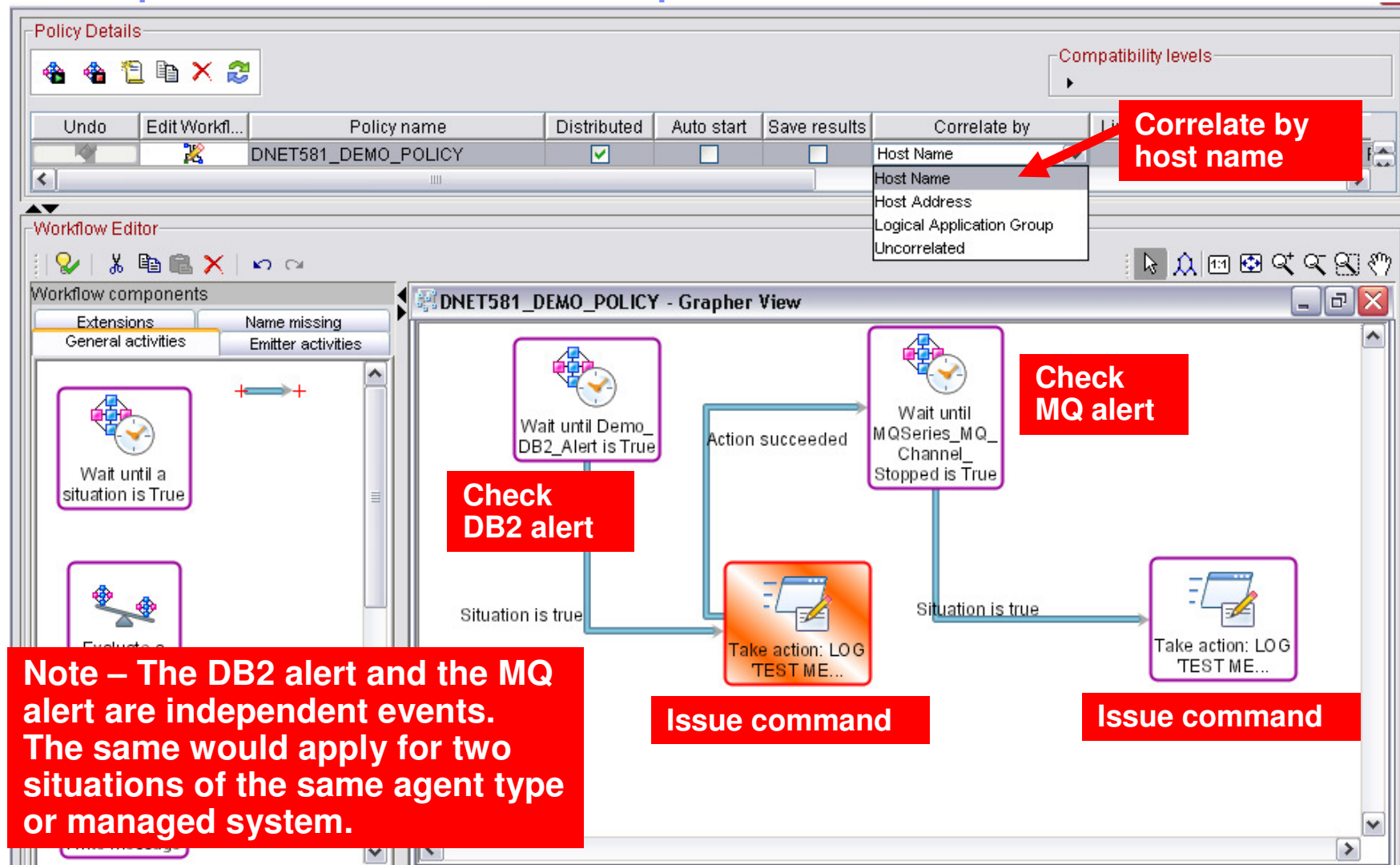
Restart

First command

Second command

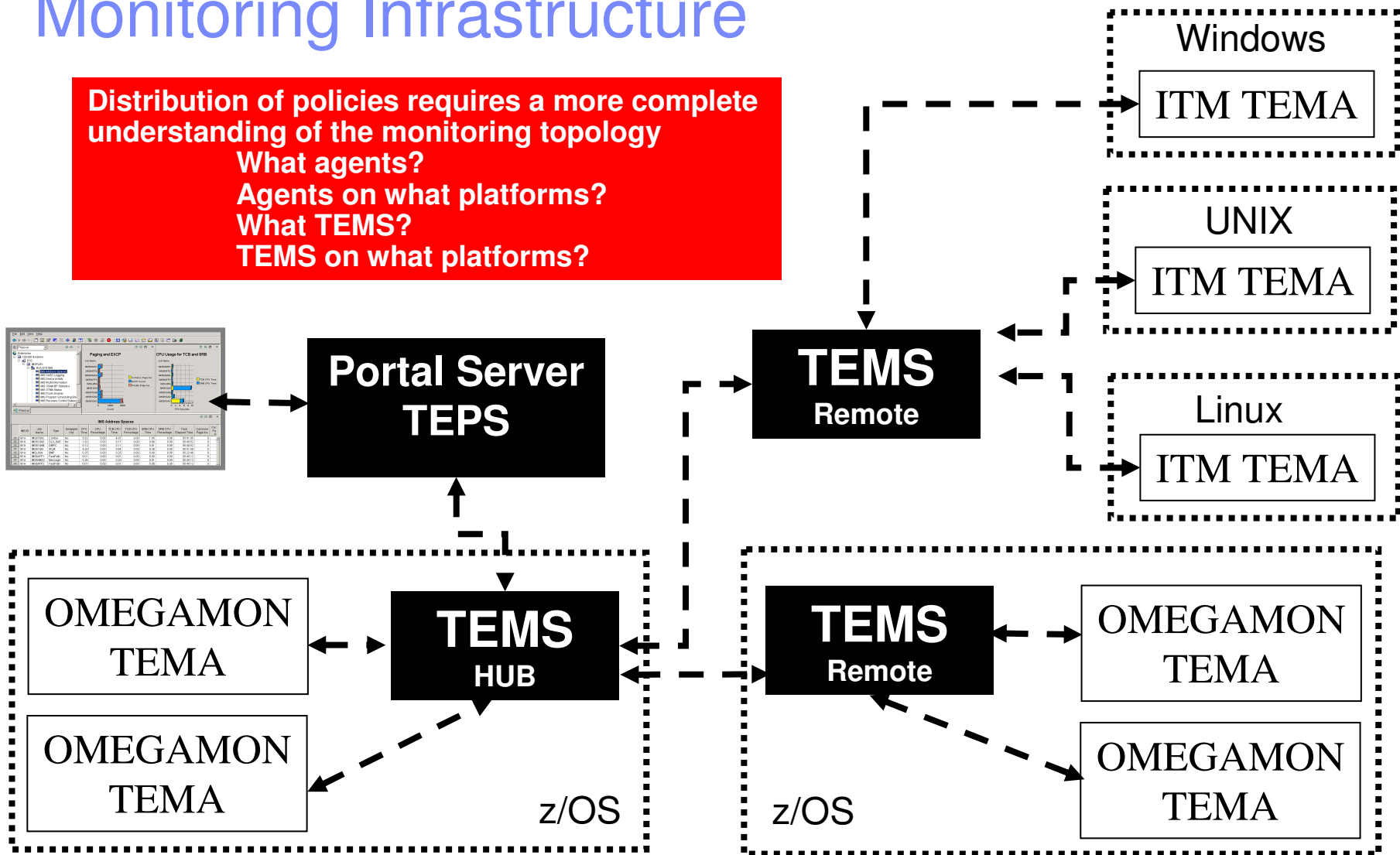
Policy Example

Multiple Situations, Multiple Commands



Policies Require An Understanding Of The Monitoring Infrastructure

Distribution of policies requires a more complete understanding of the monitoring topology
 What agents?
 Agents on what platforms?
 What TEMS?
 TEMS on what platforms?



Recommendations And Best Practices Situations And Policies What They Are And What They Are Not

▪ ***Situations And Policies – What they are***

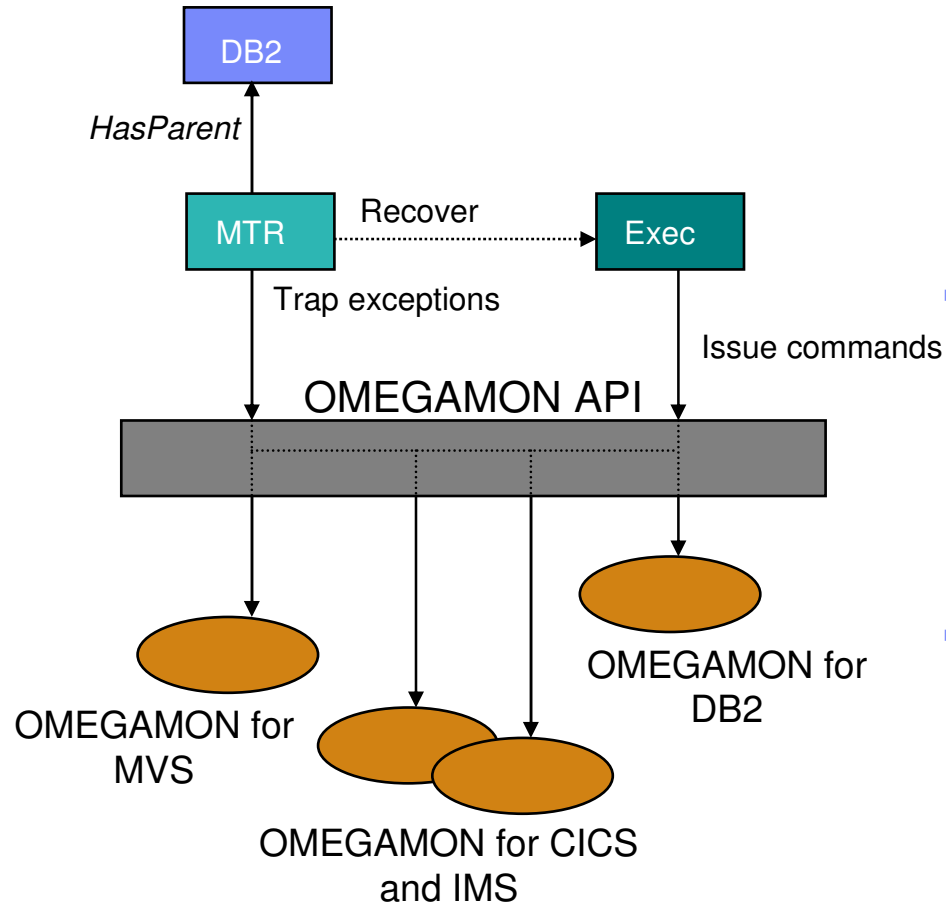
- ▶ Situations are the core alert building block of Tivoli monitoring
- ▶ Policies extend concepts established with situations and add additional functionality to the TEP
- ▶ Policies expand the integrated command and control capabilities of the TEP
 - Situations remain the essential starting point
- ▶ Policies add additional function and flexibility to situation capabilities

▪ ***Situations And Policies – What they are not***

- ▶ The command capabilities of situations and policies are not a substitute for a full function automation engine such as IBM System Automation
 - Use situations and/or policies for basic detection and command/correction scenarios
 - Use situations and/or policies to drive SA automation execs when needed
 - For more detailed scripts (such as REXX) and analysis use System Automation



System Automation Integrated With OMEGAMON Bi-directional Interfaces



- Use performance and availability information for Automated Performance Management
 - ▶ More metrics, more accurate decisions
 - ▶ Sources: MVS, DB2, CICS, IMS, Network, Webphere, Webphere MQ, Storage monitoring
- Provides APIs to communicate with OMEGAMON monitors to
 - ▶ Monitor OMEGAMON exceptions
 - ▶ Monitor/manage situation status
 - ▶ SOAP interface enables detailed performance data interface to SA
- Provides exception monitor based on the Monitor Resource concept
 - ▶ Monitors „interesting“ set of exceptions
 - ▶ Sets application health state based on existence of such exceptions
 - ▶ Provides means to react and resolve exceptional conditions

Any metric captured by OMEGAMON may be analyzed via automation



OMEGAMON, IBM System Automation And The Tivoli Enterprise Portal Provides SOAP Interface

The screenshot displays the Tivoli Enterprise Portal interface. On the left, the 'Navigator' pane shows a tree view of monitoring components, including 'Thread Activity' and 'MVS DB2'. The 'Physical' view is selected. Below the navigator, a bar chart titled 'Open Situation Counts - Last 24 Hours' shows data for 'NT_Log_Space_Low', 'Linux_alert', 'EW_Demo_Sit', and 'EW_DB2_Test'. The 'EW_Demo_Sit' and 'EW_DB2_Test' bars are highlighted in yellow. The main pane shows the 'Situations for - Thread Activity' configuration. The 'Action Selection' tab is active, showing 'System Command' selected. The 'System Command' field contains 'soap:CT_Execute,filename=test.xml'. Below this, the 'If the condition is true for more than one monitored item:' section shows 'Only take action on first item' selected. A red box highlights the 'System Command' field. Another red box highlights the 'test.xml' file in the file list. A third red box highlights the XML content in the 'test.xml - Notepad' window.

Use the SOAP interface to interrogate monitoring data and manage monitoring infrastructure

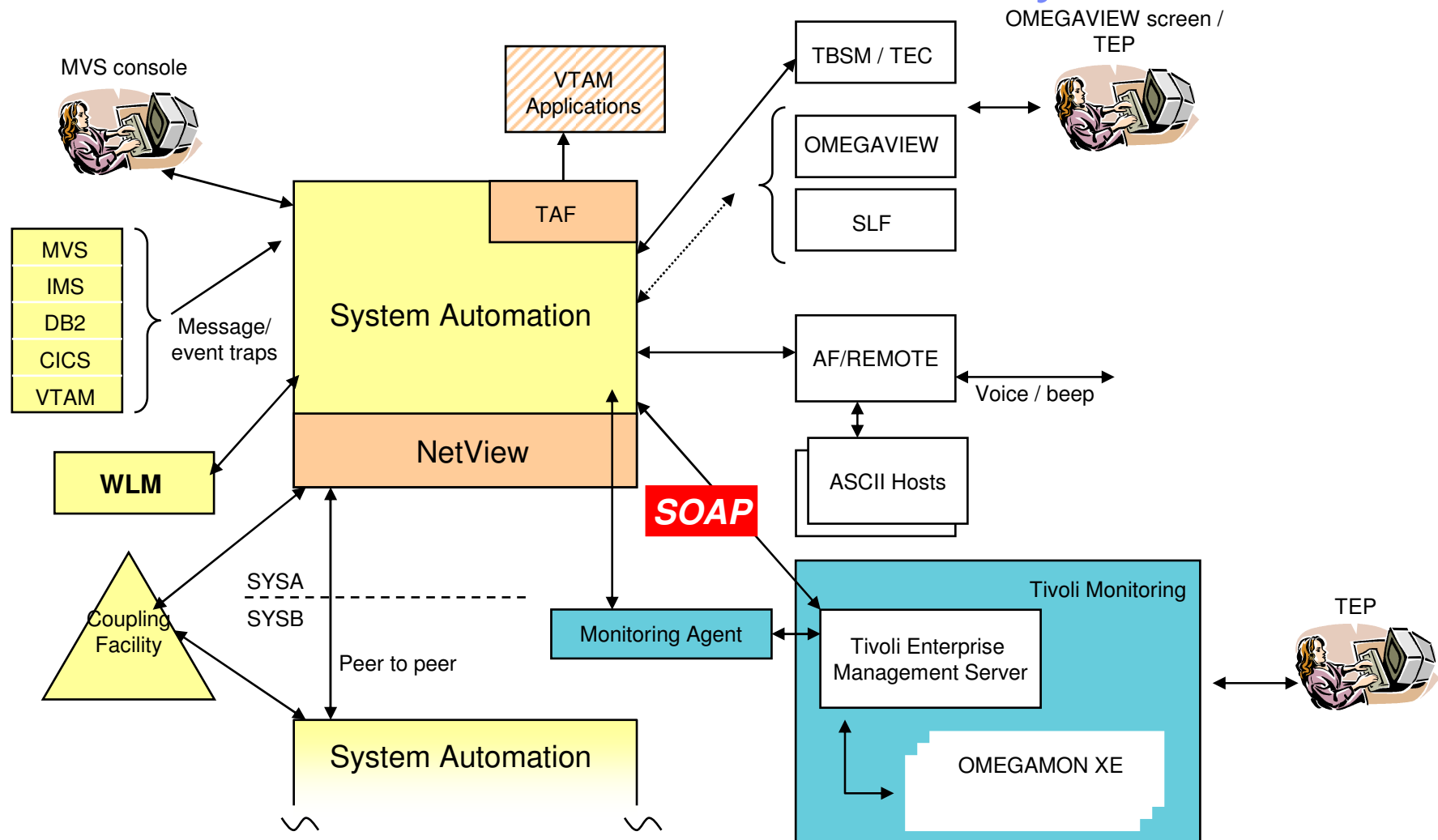
Example – use SOAP to activate/deactivate situation alerts

Example XML content from test.xml:

```
<CT_Activate>
<hub>SOAP</hub>
<name>EW_Demo_Sit</name>
<type> situation</type>
</CT_Activate>
```

SA / OMEGAMON Integration – Overview

SOAP Interface Enables Detailed Analysis



Examples Of Automated Performance Management Scenarios

- **z/OS example** - possible z/OS looping task
 - ▶ **Monitored symptoms** – high CPU loop index as measured by OMEGAMON >> WLM missing goals >> high overall system CPU usage
 - ▶ **Automation response** – adjust priority of problem task or if desired cancel the task
- **DB2 example** - DB2 object lock conflict
 - ▶ **Monitored symptoms** - long running SQL call >> high In-DB2 time >> longer thread elapsed time
 - ▶ **Automation response** - Increase priority of “owner” (as determined by automation) >> “Kill” problem thread
- **IMS example** - High IMS message region occupancy time
 - ▶ **Monitored symptoms** - IMS transactions queued >> longer IMS transaction scheduling time >> longer IMS response time >> lower IMS transaction processing rate
 - ▶ **Automation response** – automation starts additional message regions to handle workload >> issue IMS commands to adjust classes
- **MQ example** - Lower MQ message input rate >>
 - ▶ **Monitored symptoms** - Higher MQ message queue depth >> lower transaction processing rate >> longer CICS/IMS transaction response time
 - ▶ **Automation response** – issue calls to assess potential bottlenecks in CICS/IMS processing >> automation action based on results



[illegible]

z/OS Example

zIIP Processor CPU Resource Utilization Alert

Alert When zIIP Utilization Is High For zIIP Dependent Workload

Use wild card functions to track key tasks

In this example the situation will fire if zIIP utilization is high for the given workload (in this case DB2 DDF) or if a high percentage of work is spilling over to general CPs.

	Job Name	IFA Percent	IFA on CP Percent
1	== DDF	> 80.0	> 40.0
2	== DDF		> 60.0
3			

Consider using the persistence option to filter out outliers

Advanced Situation Options

Situation Persistence: ☒ Display Item

Situation Persistence

Consecutive true samples: 3

Formula Capacity: 32%

Sound: ☐ Enable critical.wav

State: **Critical**

☒ Run at startup

CICS Application Performance Example

Monitor CICS Transaction Response And Highlight High DB Wait

Using the PPS CICSplex_delay_in_Database as an example, create an alert that will highlight poor response time due to high wait time in database (either IMS or DB2).

Make the alert sensitive by tran code or WLM service class.

Note - These metrics may also be detected by System Automation via the SOAP interface

	Performance Index	% Wait on DLI	% Wait on DB2	Transaction ID	Average Response Time
1	> 1.00	> 90		== ABC	> 00:00:02
2	> 1.00		> 90	> XYZ	> 00:00:02
3					> 00:00:02

average time in milliseconds taken to... interval. This average is calculated by adding the response time of each task within a service class and dividing the sum by the total number of completed tasks during the collection interval. The value format is a positive integer, maximum 4 bytes.

Transferring data from tivtpe.demopkg.ibm.com...

Situation Formula Capacity **57%** Add conditions... Advanced...

IMS Application Performance Example

Situations To Monitor Response Time

Using boolean logic allows the alert to be application sensitive. This assumes that the KOIGBLxx macro is customized.

A single situation can handle multiple application groups, if needed.

Note – this is the RTA group name

Consider alerting on R0 versus R1 response time. R0 only considers Input Queue and processing time, and excludes outqueue time.

Note - These metrics may also be detected by System Automation via the SOAP interface

Consider using the persistence option to filter out outliers

Other Examples Of Common z/OS Critical Performance Metrics

WebSphere MQ

Queue depth
Message send/receive rate
DLQ depth
Channel status and performance

CICS

Transaction response time
Transaction rate
Region CPU rate
File I/O count
String waits
Abend messages

z/OS

System CPU rate
Paging rate
WLM Performance Index
DASD I/O MSR time and rate
Critical console messages

WebSphere

Method call count and elapsed time
Heap size
Garbage collection
Connection pool utilization

Network

Network Connection status and performance
Network interface utilization



Situations

General Recommendations And Rules Of Thumb

- Make situations Meaningful, Actionable, and Useful
- Meaningful situations
 - ▶ Situation naming is flexible – make the names understandable
 - ▶ Adopt a situation naming convention
 - Makes it easier to identify customer created versus product provided situations
- Actionable situations
 - ▶ Have appropriate notification
 - A workspace with an alert icon, command/message notification
 - ▶ As a standard have expert advice
 - ▶ Have pre-defined take actions where appropriate
- Useful situations
 - ▶ Eliminate phony alert indicators – tune out the noise
 - ▶ If an alert situation fires it should indicate an actual issue
 - An alert, an owner, and a consequence



Additional Situation Considerations And Recommendations

- Use the Product Provided Situations as examples or templates
 - ▶ Customization to user-created situations
- When creating and deploying a set of situations consider
 - ▶ The number of situations being deployed
 - ▶ The number of managed systems (i.e. z/OS LPARs and CICS tasks)
 - ▶ Refresh frequency of the situations
- Consider carefully the number of required situations
 - ▶ Use boolean logic to reduce the number of needed situations
 - ▶ Do not automatically make a warning alert to go with each critical alert
 - Create a warning if it will allow time to address an issue before going critical
 - ▶ Use managed system lists to send the right situations to the right managed systems
- Be aware of the situation refresh rates
 - ▶ Multiple situations on the same table with the same refresh rate may be optimized by the infrastructure
 - ▶ Potential to reduce monitoring overhead if done appropriately



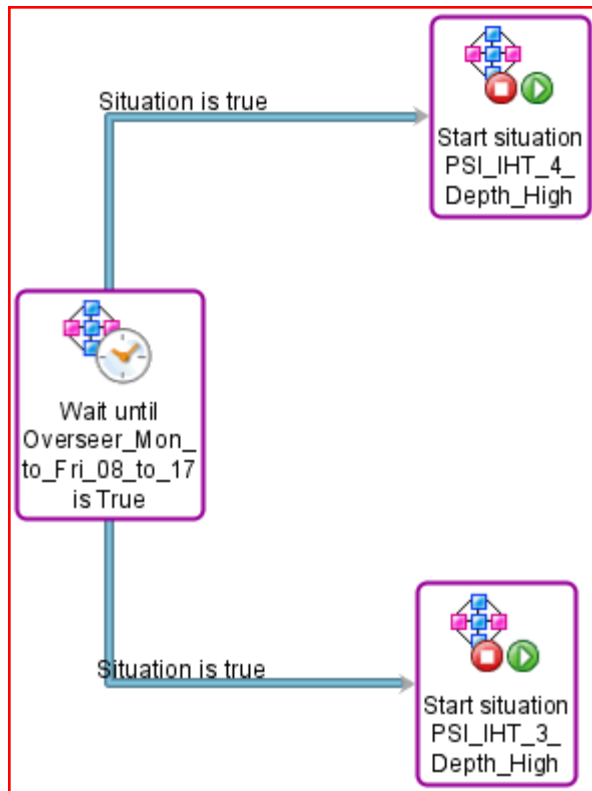
Policies And System Automation Recommendations And Rules Of Thumb

- Policies are not a substitute for System Automation and REXX command script capabilities
 - ▶ Policies work well as an extension of situation capabilities
 - ▶ Policies work well to manage start/stop of situation logic
 - ▶ Policies work well to issue multiple actions and “feed” other tools
- IBM System Automation
 - ▶ Use for full function automation logic and routines
 - REXX exec script capabilities
 - ▶ Use for more complex logic and actions
 - ▶ Exploit the ability of the SOAP interface to pull in key performance metrics from OMEGAMON

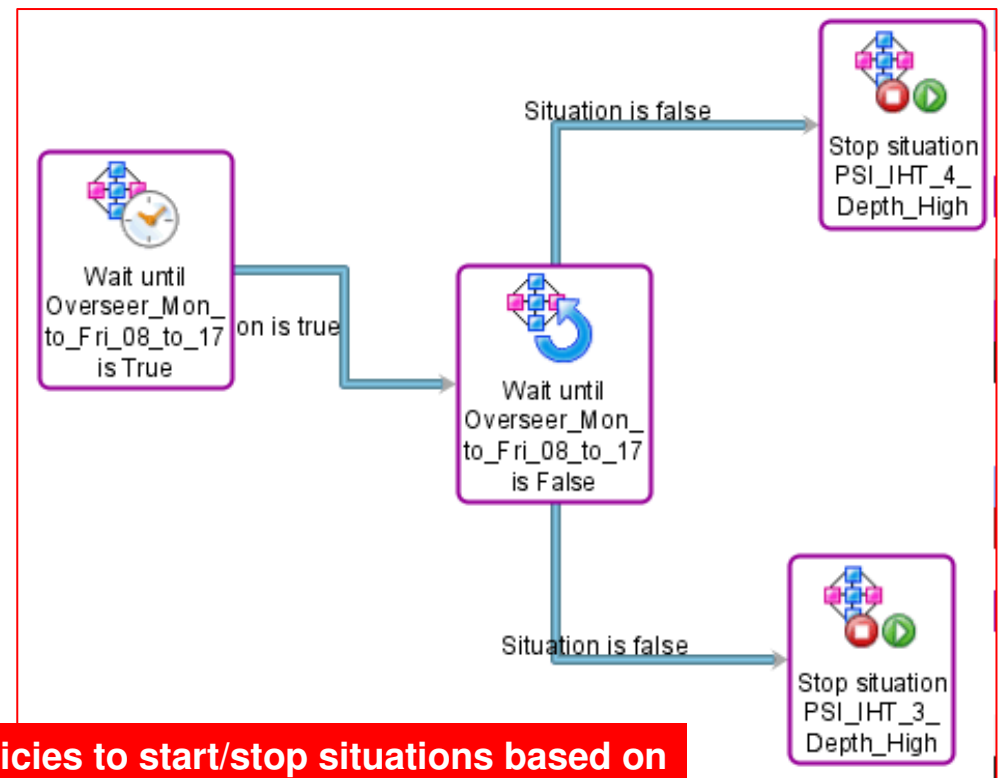


Example - Using A Policy To Manage Situations Based Upon Time Of Day Requirements

Overseer policy to start situations



Overseer policy to stop situations



Use policies to start/stop situations based on a variety of criteria

Time of day
Start specific analysis situations
and more.....

Roadmap

Automated Performance Management

- Use a building block approach
- **Situations** - Start with identification and definition of situation alerts
 - ▶ Meaningful alerts that represent true potential issues
 - ▶ Use the analysis to identify critical monitoring metrics
- **Policies** – Use policies where appropriate
 - ▶ Situation management and correlation
 - ▶ Issuing commands for basic performance/availability issues
- **Visualization** – Define useful Tivoli Portal views
 - ▶ Customize screens in the Portal for specific audiences
 - Operations, applications, management
- **System Automation** – exploit the power of integration
 - ▶ Define example performance automation management scenarios
 - ▶ Leverage the process as a template for additional scenarios



Summary

- The IBM Monitoring And Automation suite provides powerful automation capabilities in multiple core technologies
 - ▶ IBM System Automation console management
 - ▶ IBM Tivoli OMEGAMON monitoring
 - ▶ IBM Netcool OMNibus network monitoring and management
- Automated Performance Management leverages the intrinsic integration capabilities of the IBM technologies
 - ▶ Automation integration with monitoring
 - ▶ Integrated monitoring and management (including cross platform)
- Leverage Automated Performance Management to improve problem isolation and MTTR
 - ▶ Understand the unique capabilities of OMEGAMON and the Tivoli Portal
 - ▶ Use a building block approach to grow management logic over time



Thank You!!



Check Out My Blog

http://tivoliwithaz.blogspot.com



Tivoli With Az

This is a blog to discuss what is happening in the area of IBM zSeries, Tivoli, OMEGAMON monitoring, System Automation, and other relevant IBM Tivoli technology for z/OS performance and availability management.



Ed Woods
IBM Corporation

Friday, February 5, 2010

OMEGAMON DB2 Near Term History



OMEGAMON DB2 has a very useful Near Term History (NTH) function. NTH provides an easy way to be able to retrieve and review DB2 Accounting and Statistics records from the past few hours of DB2 processing. The data is stored in a set of VSAM files allocated to the OMEGAMON collection task. How far back the history goes depends upon the size of the files and the amount of data being written to these files. Now some of the data volume is driven by the DB2 workload activity. Accounting records are typically written when a DB2 thread terminates processing, and it is the Accounting data that is often looked at by the analyst when studying what DB2 applications have been doing. Statistics records are created on a time interval basis. Usually, you will have much more accounting data than statistics data. Also, OMEGAMON has the ability to pull in additional trace IFCIDs to get information on things such as dynamic SQL activity.

To understand the amount of data being gathered by NTH, there are displays that show the number of records written to the NTH files, by type. In the example I show, you see an example of common NTH settings/options, and then you see the record count in the NTH record information display. If you look carefully you see that 'Perf-Dyn SQL' has a lot of records written relative to the other record types. This is a good way to understand the impact of enabling certain collection options, such as dynamic SQL collection, and see how many trace records are being gathered, as a result.

Posted by Ed Woods at 3:13 PM 0 comments

ED WOODS

I'm an IT Specialist with IBM Corporation supporting Tivoli Performance solutions on z/OS. Please note that comments made on this blog are my own, and do not necessarily reflect the position of IBM Corporation.

[View my complete profile](#)

Links To My Articles

- [DB2 Thread Situations](#)
- [OM XE For Mainframe Networks](#)
- [Situation usage and best practices](#)
- [Situation best practices - part 2](#)
- [Article on policy automation](#)
- [Article on monitoring DB2 dynamic SQL](#)
- [IMS historical performance analysis](#)

Useful Links

- [Link to IBM Tivoli product information](#)
- [Link To Tivoli User Group](#)
- [Link to OPAL](#)
- [Tivoli System z Blog](#)