The Service Aware CMDB

What is it, why its important, and how to successfully build it

Ariel Gordon
VP-Products ServiceNow
Introducing Ariel Gordon ...

- VP Product Management in Service Now
- Co-Founder & VP Products at Neebula
- CTO of BMC Software
- VP R&D, CTO of New Dimension Software
Agenda

• What is a “Service-Aware” CMDB?
• What is the Value of Service Models?
• Why Most Implementations Fail ... and a Path to Success in Creating a “Service-Aware” CMDB
What are Service Models?

ITIL
A Service Model is a hierarchical model of the services, assets and the infrastructure created by recording the relationships in the CMDB between configuration items.
What is a “Service-Aware” CMDB?

A Configuration Management Data Base that maintains a near-real-time view, of hierarchical relationships among the components in IT service configurations. With the following capabilities:

• Integration and Federation - Obtaining information from many sources (MDR’s) & linking to information in other MDR’s
  – Reconciliation - Normalize data and combining it from many sources
  – Business Service Modeling and Visualization - creating and visualize service models
  – Synchronization - Synchronizing with internal process such as change
Problem of business services management gap

Computer Center Director of Computing Infrastructure

- **Servers**: Apollo server **fell down**
- **Applications**: "The response exceeded **SQL**"
- **Storage**: "We lack space on the **LUN**"
- **Virtualization**: upgrading the **hypervisor**
- **Cloud**: "**SSL certificate**" problem
- **Network**: "**VLAN decreased performance**"

Service Gap

- Which components support service?
- What service does not work?
- What is the importance of service?
- What should I fix first?
- Is the infrastructure will change my service?

Services to business

Good and available

- **Sales**: "Can not send** email"
- **Support**: "Our **support site** is not working"
- **Finance**: "You can make **quarterly** reports"
- **Human resources**: "You can not pay **salaries**"
- **Production**: “Unable to locate parts warehouse”
- **Management**: "Do I have a backup **system orders""
Solution by modeling business services

<table>
<thead>
<tr>
<th>Computer Center Director of Computing Infrastructure</th>
<th>Service Gap</th>
<th>Services to business Good and available</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Servers</strong></td>
<td>Apollo server <strong>fell down</strong></td>
<td><strong>Sales</strong></td>
</tr>
<tr>
<td><strong>Applications</strong></td>
<td>&quot;The response exceeded SQL&quot;</td>
<td><strong>Support</strong></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>&quot;We lack space on the LUN&quot;</td>
<td><strong>Finance</strong></td>
</tr>
<tr>
<td><strong>Virtualization</strong></td>
<td>upgrading the <strong>hypervisor</strong></td>
<td><strong>Human resources</strong></td>
</tr>
<tr>
<td><strong>Cloud</strong></td>
<td>&quot;SSL certificate&quot; problem</td>
<td><strong>Production</strong></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>&quot;VLAN decreased performance&quot;</td>
<td><strong>Management</strong></td>
</tr>
</tbody>
</table>
IT Configuration Management evolution

- CMDB
- CMDB
- Service Aware CMDB
CMDB Cataloging the IT infrastructure “jungle”

- Model physical component of the service:
  - Server configuration
  - Installed SW
  - Memory
  - Etc.
- Model logical component of the service (soft data):
  - Server Owner
  - Server criticality
  - Project
  - Etc.
Topological model showing the services provided CMDB

Infra Apps

DB

Servers

Network device

Storage

WH Service

Meta Data

Owner

Revenue

Users

IIS

Apache

Tomcat

Web Logic

IIS 2

IIS 3

Biztalk

XYZ

SQL1

SQL 2

Oracle1

Oracle2

SQL3

SQL4

SQL5

SQL6

UNIX 1

UNIX2

Lin 1

Lin 2

WinSRV1

WinSRV2

VM1

ESX

Switch 1

Switch 2

Switch 3

Switch 4

Switch 5

Switch 6

Switch 7

Switch 8

Storage 1

Storage 2

Storage 3

Storage 4
What is the Value of “Service Models”? 
Service Maps are Important ... Illuminate How IT Affects the Business

• Incident & Problem Management
  – Radically improve recovery speed (MTTR) – the main target of Incident management
    • Escalation of Incidents and Problems to the correct owner
    • Rapid problem isolation based on topology
    • Clear view of a CIs impact SLAs and OLAs

• Change Management
  – More agile change management process – understand impact of CI on a service
  – Reduction of incidents that originate from change – understand the risk of change
  – Faster isolation of incident due to changes – complete history

Valuable to following ITIL processes ...
  – Incident & Problem
  – Change & Release
  – Availability Management
  – Business Continuity
  – Compliance
  – Vulnerability Assessment
  – Data Center Migration
Service models and Availability Management
Business Context Required when Responding to Events

• NOC Operator Expected …
  – To handle 1000’s of cryptic events
  – Understand the impact on 100’s of services
  – Understand the correlation to customers’ service complaints

• And make these decisions within minutes to reduce MTTR
How Service Models Used for Event Management

Service

Application

Account to Account

Form Checking

International

Platform

WebSphere

Web Logic

MQ

Swift

Virtual Server

VM1

VM2

VM3

Host Server

ESX1

ESX2

ESX3

AIX

SUN

Storage

SAN1

Network

Router1

Swift
<table>
<thead>
<tr>
<th>Service</th>
<th>Funds Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Account to Account</td>
</tr>
<tr>
<td></td>
<td>Form Checking</td>
</tr>
<tr>
<td></td>
<td>International</td>
</tr>
<tr>
<td>Platform</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WebSphere</td>
</tr>
<tr>
<td></td>
<td>Web Logic</td>
</tr>
<tr>
<td></td>
<td>MQ</td>
</tr>
<tr>
<td></td>
<td>Swift</td>
</tr>
<tr>
<td>Virtual Server</td>
<td>VM1</td>
</tr>
<tr>
<td></td>
<td>VM2</td>
</tr>
<tr>
<td></td>
<td>VM3</td>
</tr>
<tr>
<td>Host Server</td>
<td>ESX1</td>
</tr>
<tr>
<td></td>
<td>ESX2</td>
</tr>
<tr>
<td></td>
<td>ESX3</td>
</tr>
<tr>
<td></td>
<td>AIX</td>
</tr>
<tr>
<td></td>
<td>SUN</td>
</tr>
<tr>
<td>Storage</td>
<td>SAN1</td>
</tr>
<tr>
<td>Network</td>
<td>Router1</td>
</tr>
<tr>
<td></td>
<td>Swift</td>
</tr>
</tbody>
</table>
### How Service Models Used for Event Management

<table>
<thead>
<tr>
<th>Service</th>
<th>Funds Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>Account to Account</td>
<td></td>
</tr>
<tr>
<td>Form Checking</td>
<td></td>
</tr>
<tr>
<td>International</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td></td>
</tr>
<tr>
<td>WebSphere</td>
<td></td>
</tr>
<tr>
<td>Web Logic</td>
<td></td>
</tr>
<tr>
<td>MQ</td>
<td></td>
</tr>
<tr>
<td>Swift</td>
<td></td>
</tr>
<tr>
<td>Virtual Server</td>
<td></td>
</tr>
<tr>
<td>VM1</td>
<td></td>
</tr>
<tr>
<td>VM2</td>
<td></td>
</tr>
<tr>
<td>VM3</td>
<td></td>
</tr>
<tr>
<td>Host Server</td>
<td></td>
</tr>
<tr>
<td>ESX1</td>
<td></td>
</tr>
<tr>
<td>ESX2</td>
<td></td>
</tr>
<tr>
<td>ESX3</td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td></td>
</tr>
<tr>
<td>SAN1</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>Router1</td>
<td></td>
</tr>
<tr>
<td>Swift</td>
<td></td>
</tr>
<tr>
<td>SUN</td>
<td></td>
</tr>
</tbody>
</table>

© 2014 ServiceNow All Rights Reserved
How the Model is Used for Event Impact

Service

Application

Platform

Virtual Server

Host Server

Storage

Network

Funds Transfer

Account to Account

Form Checking

International

WebSphere

Web Logic

MQ

Swift

VM1

VM2

VM3

ESX1

ESX2

ESX3

SAN1

AIX

SUN

Router1

Swift
Why Many Implementations Fail... and a Path to Success in Creating a “Service-Aware” CMDB
Reasons why Implementations Fail Today

• Service Modeling Use Cases are too Broad
• Lack of Management Commitment
• Other Project management issues
• Projects Take too Long to Show Value
• Lack of Service Model Automation Delays Results
• Accuracy of Service Models Lost Over Time
The Manual Approach ... Simple?

• Benefits
  – Requires no tools (Visio)
  – Enables implementation of CMDB and BSM
  – Forces implementation of processes
  – Forces a common language between the teams
  – Works well in a small organization

• Challenges
  – The model of each application is defined by the application owner
  – Populated into a CMDB and into the event impact tools
  – Maintenance - model must be updated every time a change occurs
The Enemy is Change and in particular

- Un-reviewed ad-hoc changes
ADM tools for service model building

“Bottom-Up” Discovery

Infrastructure Discovery By Domain

Application Dependency Mapping

ADDM Discovery

© 2014 ServiceNow All Rights Reserved
“Bottoms-Up” manual Service Modeling Approach Using ADM Tools

Build Model

Gather information from application owner

Discover all applications, the infrastructure that supports them, and their interconnections

Total 14 days
Use of APM Transaction Management Tools

• Benefits
  – Top-Down approach: Define critical transactions
  – On-Going - Automatically updated based on transactions

• Challenges
  – Coverage: only a small subset is modeled
  – Cannot be used to map everything
  – Integration to CMDB and other tools limited
  – Usually agent based

Not a viable solution ... can be used only for selected applications
Why “Service Models” are not Common?

Using CMDB Tools modeling tools is a Sisyphean Task

- ADM tools do not map real service dependencies
- Building manually takes a long time and, due to pace of change, model is most-likely outdated.
- Projects take to long to show value and are scraped
- Environment changes rapidly

- Real-time changes in dynamic environments
- Undocumented changes
- Lack of application knowledge
A new approach to service modeling

- Only needs service entry point
- Updated in real time
- Cross-technology
- Extensive vendor knowledgebase
- Easy creation of new patterns
- Built for virtualized environments
- Agentless
The difference between ADM tools and Top Down modelling

**Infrastructure Discovery By Domain**

**Application Dependency Mapping**

**Service Dependency Mapping**

ADM Tools

Top Down mapping tools
Creating a Real-Time Service CMDB

Enable benefits of true Service Management

Leverage existing CMDB investments

Automate and accelerate Service Modeling

Feed Service Models into CMDB

Service Models updated in real time as changes occur

Track changes in dynamic virtualized environments

ADM Tool

Infrastructure Discovery & Mapping

Configuration Items

CMDB API

Service Model

Top Down Mapping

Service Modeling & Mapping

Email

Payroll

E-Banking
## Proven Customer Value – Expected Value

<table>
<thead>
<tr>
<th>Product</th>
<th>Service Mapping Standard Apps</th>
<th>Service Mapping Proprietary Apps</th>
<th>Total # Days per 80 services</th>
<th>Project Length with 2 FTEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Down Discovery tool</td>
<td>1 work day</td>
<td>2 work days</td>
<td>120</td>
<td>3 months</td>
</tr>
<tr>
<td>Discovery + Manual Mapping</td>
<td>8 work days</td>
<td>14 work days</td>
<td>960</td>
<td>2 years</td>
</tr>
</tbody>
</table>

- Reduction in time for service modeling projects
- Time savings when resolving critical IT problems
- Increase labor productivity – assign to more critical tasks
- Reduction in change-related incidents
Thank You!