A Service Management Architecture Component Model

Jianwen Chen, Rhonda Childress, Iain Mcintosh,
George Africa, Arthur Sitaramayya
IBM Global Service
601 Pacific Highway, St. Leonards, NSW 2065, Australia

jchen@au1.ibm.com, childres@us.ibm.com,
iainmci@au1.ibm.com, georgeaf@au1.ibm.com,
arthursi@au1.ibm.com

Abstract—**In this paper, we are proposing a four-layer enterprise architecture model for service management. The aim of the proposed architecture is to provide a framework for the automated, centralized, real-time enterprise service management solution. We developed a component model to address the functional requirements for different integrated service management functions.**

Keywords-service management; architecture; component model

I. INTRODUCTION

Service management is the discipline used in industries to provide IT services. Today, there are some service management products and systems [1][2] provided by service vendors which include: Remedy Service Management, USD Service Management, Novell ZENworks, Oracle Enterprise Manager, Altrius Management Suite, Accenture, CA Unicenter, and Microsoft Systems Management Server. There are different limitations and disadvantages from these service management products and systems, most important, what is lacking for service management is an architecture to provide an enterprise level solution for a real-time, automated, and centralized service management system, integrated with toolset management and aligned with IT Infrastructure Library (ITIL) V3 process framework[3].

We propose a four-layer enterprise architecture for service management [4][5][6]. Compared with other service management system, the advantages of our proposed service management architecture include: 1) It provides a service oriented architectural approach for service delivery; 2) It complies with the ITIL V3 process framework; 3) It integrates service management, system management and toolset management; 4) It provides an architecture for an automated, centralized, real-time service management system; 5) It proposes a portal solution that provides a common presentation layer with an integrated view from the business, IT management and IT operation perspective.

The component model is used to address functional requirements for an architectural design. It is where the IT architect or designer captures the functional characteristics of what the solution must do – the functional specification of the system. It breaks down the solution design into manageable, modular pieces, called technical components, to facilitate development, testing and deployment of the service. We propose a component model that addresses the functional requirements of the service management functions and integration between these service management functions. In the component model, we define functional components, identify relationships between components and describe how function components can address the functional requirements of service management.

II. AN ARCHITECTURE OVERVIEW

We propose a four-layer enterprise service management architecture. The overview of this architecture is presented in Figure 1 below:

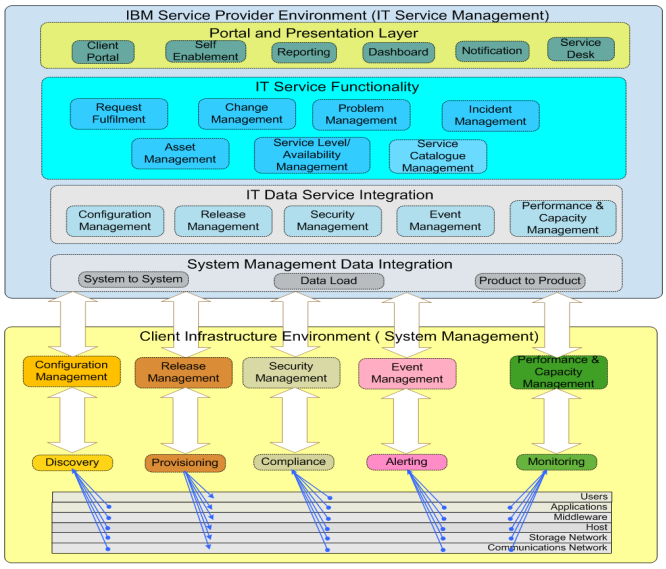

Figure 1: Service Management Architecture Overview

Layer 1 is the system management layer of a client environment consists of various ITIL service management functions and collection of data from operational domains utilizing system management tool agents or agent less methods. The essential service management functions in this layer are: configuration management, release management, event management, security compliance management, performance/capacity management. Layer 2 is system management and data integration layer: The collection systems within the client environment integrate to the service

management functions by utilising an Enterprise Service Bus (ESB) as transport and the underlying integration interfaces. Layer 3 is the IT service management layer provides the management for the data collected from the client environment. The essential ITIL service management functions in this layer are: Request fulfilment management, Incident management, Problem management, Change management, Asset management, Service level/availability management, Service catalogue management. Layer 4 is the service management portal and the presentation layer: service management staff can access the service management layer via a single portal and a presentation layer common to all service management tools.

III. A COMPONENT MODEL FORMALIZATION

A. *Component List*

We start to formalize the component model by identifying and specifying the functional components. Table 1 has listed all the identified functional components on the four layers of the proposed service management architecture:

Service	Component Name
Service Management	Incident Management
	Problem Management
	Change Management
	Request Fulfillment
	Service Catalogue Management
	Service Level Management
	Asset Management
System Management	Event Management
	Configuration Management
	Release Management
	Security Management
	Performance/Capacity Management
Operational Domains	Application Management
	Host Management
	Storage Management
	Network Management
Integration	Data and Service Integration
Portal and Presentation	Client portal
	Delivery portal
	Self-service enablement
	Dashboard
	reporting
	Notification
	Service Desk

Table 1: Component List Specification

B. *Component Relationship Diagram*

After defining the essential functional components, we formalize the component model further by identifying and describing the relationships between the functional components and sub-functions. In the proposed component model, the component relationship diagram is used to define all functional components/sub-functions, and describe the relationships between components. The component relationship diagram is shown in Figure 2:

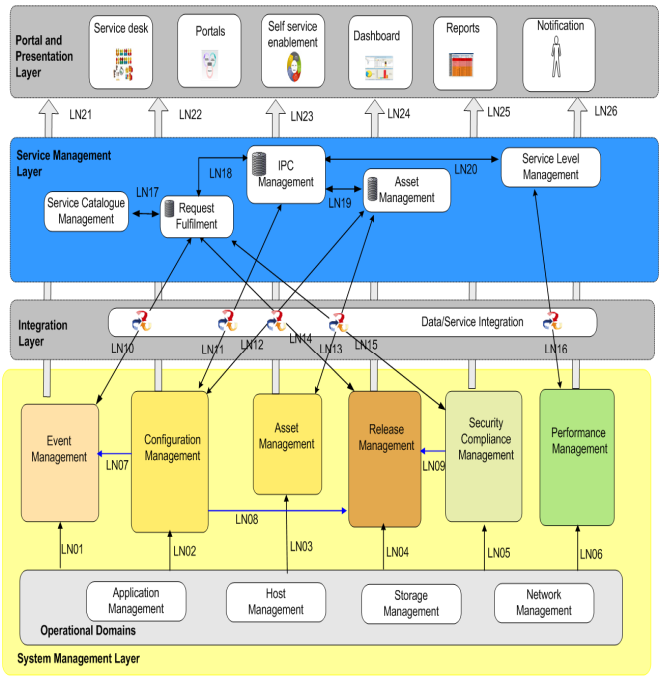

Figure 2: Component Model - Component Relationship Diagram

In the service management layer, linkages NL01-LN06 show the component relationships between components in operational domains and system management components. Linkages LN07-LN09 identify the component relationships within system management layer. Linkages LN10-LN16 describe the relationships between service management components and system management components. Linkages LN17-LN20 show the component relationships within core service management system. Linkages LN21-LN26 describe the relationships from system management components to presentation/portal components. Table 2 has described linkages in the proposed component model:

Linkage Number	Linkage Description
LN01	It is for event data collection/scanning from operational domains.
LN02	It is for configuration data collection/scanning from operational domains.
LN03	It is for asset data collection/scanning from operational domains.
LN04	It is for release management to provide provisioning service onto operational domains.
LN05	It is for security data collection/scanning from operational domains.
LN06	It is for performance/capacity data collection/scanning from operational domains.
LN07	It is to provide configuration information for event enrichment.
LN08	It is to provide configuration data for provisioning service.
LN09	It is for the provisioning of security patch.
LN10	It is for event management to raise event ticket to service management system.
LN11	It provides data collation /reconciliation between IPC management database and configuration management database.
LN12	It provides data collation /reconciliation between asset management database and configuration management database.

LN13	It is for collecting all asset related information into asset management database.
LN14	It is to request provisioning service through release management.
LN15	It is to request security deployment through security management.
LN16	It is to load performance data for SLA management.
LN17	It is for service request to use service catalogue.
LN18	It is for service request of incident, problem and change.
LN19	It provides data collation/reconciliation between IPC management database and asset management database.
LN20	It is for SLA management to extract information from IPC.
LN21	It is for portal/presentation services of event management.
LN22	It is for portal/presentation services of configuration management.
LN23	It is for portal/presentation services of asset management.
LN24	It is for portal/presentation services of release management.
LN25	It is for portal/presentation services of security management.
LN26	It is for portal/presentation services of performance/capacity management.

Table 2: Component Linkages Description

C. Sub-components and Relationships

In this paper, we use event management, asset management and configuration management to illustrate how we identify and describe sub-components of service management functions and relationships between them.

Event Management: An event is defined as the notification of a change of state in a managed system. Event management is the discipline of transmitting, correlating, and displaying those events in a focal point and determining the appropriate control action. In our proposed service management architecture, we have identified the following sub-components and relationships for event management as shown in Figure 3:

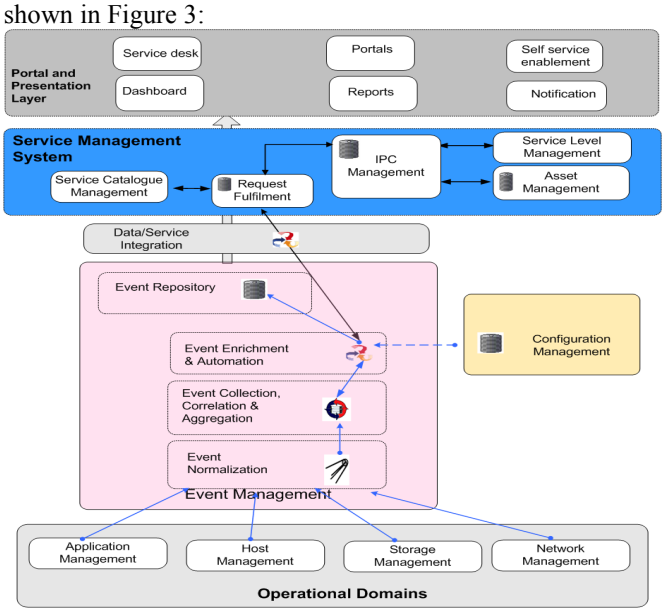

Figure 3: Component Model - Event Management

We specify four sub-components for event management:

- Event Normalization
- Event Collection, Correlation and Aggregation
- Event Enrichment and Automation
- Event Repository

Event monitoring agents capture all events across operational domains and send event data for normalization and standardization. Events are then directed into the central collection point for correlation and aggregation. Event management also takes input from configuration management for event enrichment and sends enriched event data to request fulfillment to create tickets automatically. There is a common repository to hold all event historical data used for reporting. Event management integrates with common portal/presentation layer for reporting, dashboard, and notification functions.

Asset and Configuration Management: Configuration management manages the logical and physical IT infrastructure, by identifying, maintaining and verifying Configuration Items (CIs) and their versions. Configuration management also tracks detailed hardware and software configuration and the logical relationships between the items. This CI data provides the foundation for automation between many other components. Asset Management provides processes that join financial, contractual and inventory functions to support life cycle management of hardware and software assets within the IT environment.

In our proposed service management architecture, we have identified the following sub-components and relationships for configuration and asset management as shown in Figure 4:

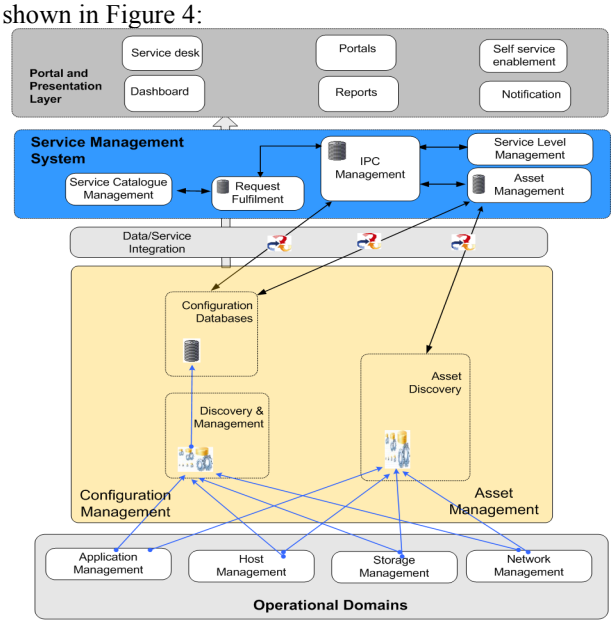

Figure 4: Component Model - Asset and Configuration Management

We specify two sub-components for configuration management:

- Discovery and Management
- Configuration Database

Configuration discovery and management sub-component collects, maintains and verifies configuration items, tracks the relationship between CIs, and populates CIs to configuration database. Asset management is a component

in the service management layer, it maintains the asset database, the relationship of assets with CIs and provides asset management services including asset tracking, product acquisition, asset financial management, logistics support, hardware tracking, inventory capture and software license management. Asset discovery sub-component located in the system management layer collects asset data from operational domains and loads asset data into asset database. The configuration management, IPC management and asset management integrate each other for data reconciliation and correlation between three databases.

D. Addressed Functional Requirements

We illustrate how we address the functional requirements of service management using two examples below:

Functional Requirement	How it is addressed
Service management architecture needs to propose an automatic and centralised service management system with: • Automated bridges between incident, problem and change records. • Integrated IPC, request fulfilment, SLA, asset, service catalogue management functions within one service management system. • Integration between system management functions and service management functions. • No additional client service management system is required.	Our service management architecture proposes ISM as an automated and centralised service management system. • ISM has automatic bridges between incident, problem and change records. • ISM has integrated IPC, request fulfilment, SLA, asset, service catalogue management functions in one system. • ISM integrates service management functions with system management functions: event, asset and configuration, release, security, and performance/capacity management. • ISM is a centralised service management system and no additional client service management system is required.
Service management needs to propose a real time and automated event management system with: • A central point for event collecting, aggregation and correlation. • Event enrichment and auto ticket functions. • Repository for real-time and historical event data. • A single reporting/presentation layer for event management. • Integration with centralized service management system • Event monitoring on each layer of operational domains.	Our service management architecture proposes a real time and automatic event management system : • Event capture engines are all directed into Omnibus. Omnibus is a central point of event collecting, aggregation and correlation. • Impact is adopted to produce auto event ticket, provide event enrichment and integrate with ISM Service Management. • TDW is used as event data repository. • GSMRT are used to provide integrated single layer reporting for event management. • Event monitoring across all operational domains via ITM6-ITCAM-Omegamon family.

Table 3: Addressed Functional Requirements

We adopt IBM Service Management System (ISM) as the core service management system to provide an automated, centralized and real-time service management system. We propose a real time and automated event management that utilizes IBM Impact, OMNIbus and IBM Tivoli monitoring family; a centralized and automated configuration management through IBM TADDM integrated with ISM database; a centralized, process driven, automated asset management through IBM TAMIT; an automated release management through ISM release function; an automated, real-time, historical and proactive performance management through IBM Tivoli Data Warehouse (TDW) and IBM Tivoli Performance Analyser (TPA); an automated and centralized security management for security compliance through IBM TEM, Epolicy and TDW toolsets.

IV. CONCLUSIONS

In this paper, we firstly propose a four-layer enterprise architecture for service management. This architecture provides an automated, centralized, real-time service management system at the enterprise level and integrates service management, system management and toolset management automatically. We further formalize a component model that addresses the functional requirements for request fulfillment, incident management, problem management, change management, asset management, service level/availability management, service catalogue management, event management, configuration management, release management, security compliance management, performance/capacity management, and integration between these service management functions. The functional components, sub-components, and relationships between them have been identified and specified in the proposed component model. The proposed service management architecture and component model can be adopted as the framework for the solution design and implementation of service management system in the real business world.

REFERENCES

[1] Rob England, " Making ITSM Real", the14th Annual International IT Service Management Conference, 2010.

[2] Maheswaran Surendra, " IT Service Management and Delivery for an Enterprise," the 6th International Conference on Network and Service Management, 2010.

[3] Patrick Bolger, "ITIL State of the Nation: The Reality of ITIL", ITSM Fusion 10, 2010.

[4] J. Black, "An integration model for organizing IT service management," IBM Systems Journal, Vol 46, No 3, 2007.

[5] IBM Global Service Limited, IBM Global Delivery Architecture Framework.

[6] Deborah L Anthony, "Frameworks, Methodologies and Standards: Determining the Right Mix of Each for ITSM Success", ITSM Fusion 10, 2010.

GLOSSAY INDEX

GSMRT: IBM Global Systems Management Reporting Technology
Impact: An IBM tool for event automation and enrichment.
ITM6: IBM Tivoli Monitor Version 6
ITCAM: IBM Tivoli Composite Application Manager
OMNIbus: An IBM tool for event management
Omegamon: IBM tool for event monitoring on mainframe.
SLA: Service Level Agreement
TAMIT: IBM Tivoli Asset Management for IT
TADDM : IBM Tivoli Application Dependency Discovery Mapping