Integrated change and configuration management

C. Ward
V. Aggarwal
M. Buco
E. Olsson
S. Weinberger

This paper provides a summary of the best-practice change-management and configuration-management processes that express a core which conforms to ITIL® and discusses how they are extended for the service provider domain. These customizable processes, coupled with an execution platform and a configuration-management database, form the essence of the IBM Tivoli® Change and Configuration Management Database (CCMDB)—the heart of the IBM strategy for information technology service management (ITSM). We provide an overview of ITSM best practices and present details of the best-practice processes developed by IBM for the CCMDB product. We also describe a number of insights gained from implementing these processes and discuss issues that are key to implementing them in a service provider environment.

INTRODUCTION

Information technology (IT) services are critical to almost every enterprise. Moreover, the competitive business climate dictates efficient and cost-effective delivery and support of IT services. Coupled with the complexity, diversity, and distributed nature of IT environments, it is clear that common, best-practice procedures to manage IT services are essential. The IT Infrastructure Library** (ITIL**)[1] is a framework for IT service management (ITSM) developed by the United Kingdom Office of Government Commerce, based on input from many industry leaders. ITIL is recognized as the de facto standard for managing enterprise IT. ITIL conformance is also becoming important to clients who outsource their IT services. The value of ITIL is that it provides both guidance and a common terminology for service management. However, ITIL is not prescriptive about implementation and does not specifically address ITSM from a service provider perspective; that is, from the perspective of an enterprise whose business is to manage IT for other enterprises.

The IBM Server Systems Operations team partnered with IBM Research to develop a set of best-practice processes for change-and-configuration-management service operations. Subsequently, IBM Server Systems Operations and IBM Research worked with IBM Tivoli* to develop a process solution for both configuration management and change management based on the IBM Tivoli Change and Configuration Management Database (CCMDB), leveraging this work.

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/07/$5.00 © 2007 IBM

This paper is organized in five major sections. The section "Industry best practices and context" provides an overview of ITSM best practices in the area of change and configuration management and places these practices in the broader context. The section "Integrated change and configuration-management processes" provides the details of a best-practice-based change and configuration-management processes developed by IBM and based on the CCMDB product. The sections "Configuration-management insights" and "Change-management insights" describe insights gained from implementing these processes. Lastly, the section "Service provider perspective" focuses on issues that are key to implementing the change-management and configuration-management processes in a service provider environment.

INDUSTRY BEST PRACTICES AND CONTEXT

Originally developed by the British government in the late 1980s, ITIL is comprised of a growing series of publications that outline a process-based set of best practices for IT service and systems management. "A Code of Practice for IT Service Management,"[2] from the British Standards Institution and based on the principles of the ITIL, depicts asset and configuration management and change management as core control processes that support all other ITIL service-support and service-delivery processes. According to ITIL, service delivery (which includes service-level, financial, availability, capacity, and IT-service-continuity management) provides the IT services required to support a business, while service support (which includes incident, problem, change, release management, and service desk) ensures that customers can access those IT services. ITIL concepts have contributed to the International Standards Organization (ISO) standard, ISO 20000,[3] based on the British Standard BS 15000-2, which has been superseded. Both the British Standards Institution and the Central Computer and Telecommunications Agency of the United Kingdom Office of Government Commerce recognize configuration and change management as a linchpin of ITSM.

ITIL best-practice change and configuration management

The goal of configuration management is to maintain a comprehensive and accurate logical representation of the IT environment. This online representation, known as the *configuration-management database* (CMDB), contains information on each component of the IT environment (e.g., hardware, software, documentation, service, and user) that needs to be managed separately as well as the relationships between components. These components are known as *configuration items* (CIs). The CMDB is more than a simple repository of configuration information. According to Gartner, Inc.,[4] a CMDB is distinguished by its capabilities for reconciliation, federation, mapping and visualization, and synchronization.

Although configuration management is related to asset management, the two are not equivalent. Assets and CIs are overlapping sets, but neither is a proper subset of the other. Moreover, asset management primarily supports accounting and is generally not concerned with relationships between items as is configuration management. Organizations often start by implementing an asset-management system before implementing configuration management.

Configuration management works hand in hand with change management to maintain the CMDB. In an IT environment, change is constant, and although most changes are intended to fix or improve the environment, they can often have unexpected, undesirable, and costly effects. The goal of change management is to minimize these adverse effects by requiring a request for change (RFC) and assessing the impact of a change before approving it.

ITIL and the broader context

Although ITIL and its various representations (ISO/IEC 20000 and BS 15000) are experiencing widespread popularity, there are also alternate models that reflect process improvement frameworks. Of particular note is the Component Business Model for the Business of IT (CBMBoIT),[5] which provides a model for managing the IT business from a chief information officer's perspective. In CBMBoIT, the authors have defined the specific activities for each component in the framework, enabling process decomposition down to the activity level when required. Process activities follow de facto IT process standards, with necessary extensions for the expression of a complete reference model, the IBM Process Reference Model for IT (PRM-IT).[5] Another process improvement framework that has gained a strong following is Capability Maturity Model Integration (CMMI).[6,7] CMMI provides particular emphasis on the complete software maturity process

with continuous improvement, contrasting with CBMBoIT and the ITIL expression and development of all areas within an IT infrastructure. Yet another perspective on IT process modeling is that from Control Objectives for Information and Related Technology (COBIT**),[8,9] which provides a set of generally accepted measures, indicators, processes, and best practices for IT management, from the Information Systems Audit and Control Association (ISACA). These various process improvement frameworks have received comparative critiques[10,11] with experiences from four case studies of ITIL transformation projects expressed in Reference 12.

Narrowing the discussion to configuration and change management, there are a number of associated topics that are directly related. In the context of configuration management, there is software-configuration management,[13] which focuses on managing the elements and relationships that comprise a software configuration for the software-development life cycle, and the architecture[14] and determination of the content of the CMDB[15] and workflow-choreography infrastructure,[16] including the direction, expression, and representation of policy and rules within in a CMDB.[17] Change-management topics include the well-appreciated challenges in supporting dynamic change management[18]; integration with dynamic systems management, as with change management with planning and scheduling (CHAMPS)[19]; and the use of contracts to effect the change-management process.[20]

As with other process domains, it is essential that the configuration- and change-management process conform to and support various compliance and auditing requirements, such as the Sarbanes-Oxley Act.[21] The configuration- and change-management process domains should also provide a common way to interact with other process domains; for example, by using a service-oriented architecture (SOA),[22,23] and should enable autonomic elements as practicable.[24] The expression of these processes from a standards perspective is provided in Reference 25.

INTEGRATED CHANGE- AND CONFIGURATION-MANAGEMENT PROCESSES

Change management and configuration management have a symbiotic relationship. Change management requires that the information in the CMDB maintained by configuration management in order to properly assess the impact of a requested change; configuration management relies on change management to provide information about any changes made so that configuration management can update the CMDB appropriately. Change management and configuration management are described separately in ITIL Service Support (*Figure 1*) only because historically some organizations implemented change management without having a full configuration-management process to support it. However, ITIL states, "Ideally, Change Management should be regarded as an integral part of a Configuration Management system."[26]

IBM Tivoli Unified Process

IBM Tivoli Unified Process (ITUP)[27] is a prescriptive approach to ITSM and is aligned with ITIL best practices. ITUP outlines how to achieve ITSM and provides diagrams to expose the top-level change- and configuration-management activities within these processes. New roles are introduced beyond what is described in ITIL. In change management, the change assignee, change approver, and change implementor are lower-level delegations for the change manager. Similarly, in configuration management, the configuration auditor is a lower-level delegation for configuration manager. Additionally, ITUP expresses those IBM Tivoli service-management-platform and operational-management products (OMPs) applicable for each activity.

Modeling methodology

Processes for the main activities of configuration management and change management are presented in the sections that follow. These processes were modeled with IBM WebSphere* Process Modeler Advanced Version 6.0, and the legend for them is shown in *Figure 2*. The processes were originally realized in CCMDB in Web Services Business Process Execution Language (WS-BPEL) by using the IBM WebSphere Integration Developer 6.0.

Configuration management

Configuration management includes the following processes: identify CIs, control CIs, and verify and audit CIs. It also includes the report-configuration-status function.

Identify CIs

The identify-CIs process is used to discover CIs in the environment and update the CMDB as needed (*Figure 3A*). (The ITIL Version 2 focus for this

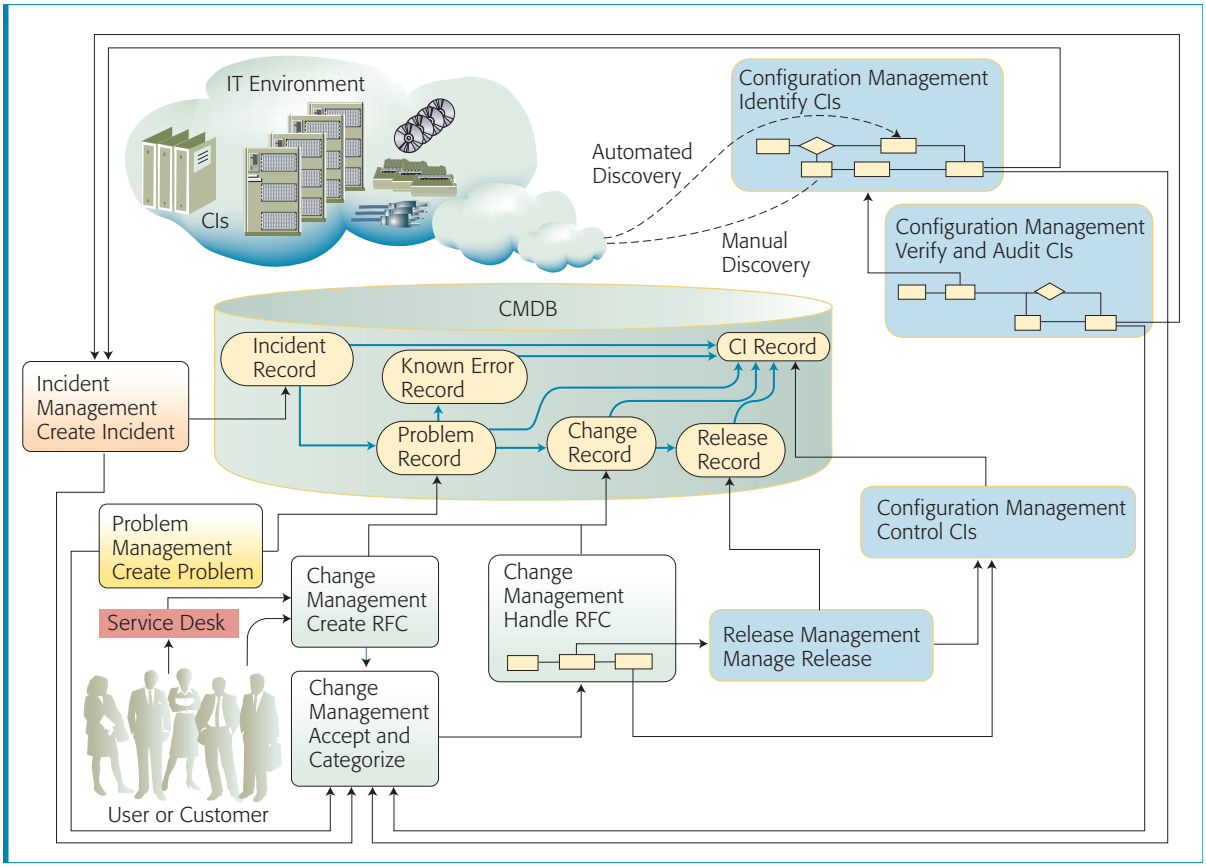


Figure 1
Service-support processes and artifacts

process is to establish the classes of CI types within the infrastructure. This might be done during the initial population of the CMDB or when a new type of CI is identified for inclusion in the CMDB.) The discovery can be accomplished by manual inspection or automated scanning tools. The gathered data should be filtered, mapped, normalized, and reconciled. Only then is the data ready to be compared with the contents of the CMDB and remediated; that is, the identified variances can then be corrected (*Figure 3B*).

There can be several reasons for a discrepancy between the CMDB and the gathered data: An unauthorized change (not associated with an approved RFC) was made to the environment; a timing problem occurred (e.g., an authorized change was made to the environment but the CMDB has not yet been updated); or an error was made performing an authorized change. The variances must be reviewed to determine the cause. An incident can be opened for further investigation, or an RFC can be opened to update the environment or the CMDB as appropriate.

Control CIs

The control-CIs process manages all updates and additions to the CMDB (*Figure 4A* and *Figure 4B*). It can only be invoked by the change-management, release-management, or other configuration-management processes. This level of control is necessary to ensure the best practice. The control-CIs process can be used to enforce whatever restrictions or policies the organization chooses to impose on the CI data. For example, for certain types of CIs, the process could check for the validity of attribute values or ensure that certain attributes are specified for particular CI life-cycle states before any update is made.

Verify and audit CIs

A key responsibility of the configuration-management process is to ensure that the CMDB accurately

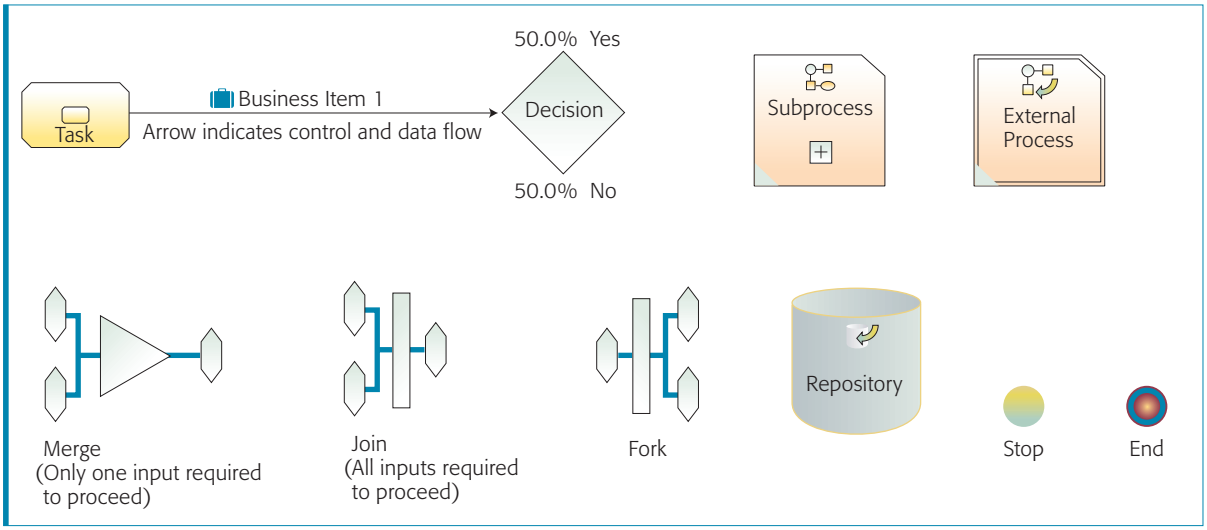


Figure 2
Legend for the process models in Figures 3-5

represents the environment and complies with the established IT standards and policies. To accomplish this, it is necessary to regularly check the contents of the CMDB against the IT environment and various standards. This responsibility is handled by the verify-and-audit-CIs process (***Figure 5***).

The verify-and-audit-CIs process invokes the identify-CIs process to scan the environment, if necessary, and compare the discovered CI data against the CI data in the CMDB. Other steps in the audit process include: determining which CIs in the CMDB have not recently been found in the environment ("recent" is determined by policy); ensuring that CI naming conventions have been followed; comparing CIs against associated gold standards to ensure compliance (a gold standard is a set of CI records or rules that serves as a model or template for how sets of CIs, for example, servers, should be configured); and checking the accuracy of the contents of the definitive hardware library and the definitive software library. Using the remediate-variances process (see Figure 3B), all variances are reviewed and resolved by opening incidents or RFCs, as appropriate.

Report configuration status

The report-configuration-status function makes CI information available to authorized users. The information can include attribute values, relationships to other CIs, change history, life-cycle state history, and relationships to other process artifacts (e.g., RFCs).

Change management

Managing change, whether to fix a problem or to improve the environment, is the domain of the change-management process. RFCs can be created by a customer or user by means of a change-management-supplied interface. Additionally, other service-support processes (e.g., configuration management) can compose an RFC and submit it directly. Once submitted, an RFC is routed to the appropriate personnel to be accepted and categorized. If accepted, the categorized RFC is handled by a process that is customized based on RFC key attributes, such as its category, group, type, and priority.

Handle RFC

The handle-RFC process manages the life cycle of a change. ***Figure 6*** depicts the sample best-practice process for handling a change and the five steps that comprise it: (1) Assess change, (2) approve and schedule change, (3) coordinate change implementation, (4) prepare, distribute, and implement change, and (5) review and close change.

The process depicted is appropriate for a major non-urgent change. A minor urgent change could be handled effectively with a small subset of the tasks shown. The attributes of an RFC may impact which activities are required to handle the RFC. For example,

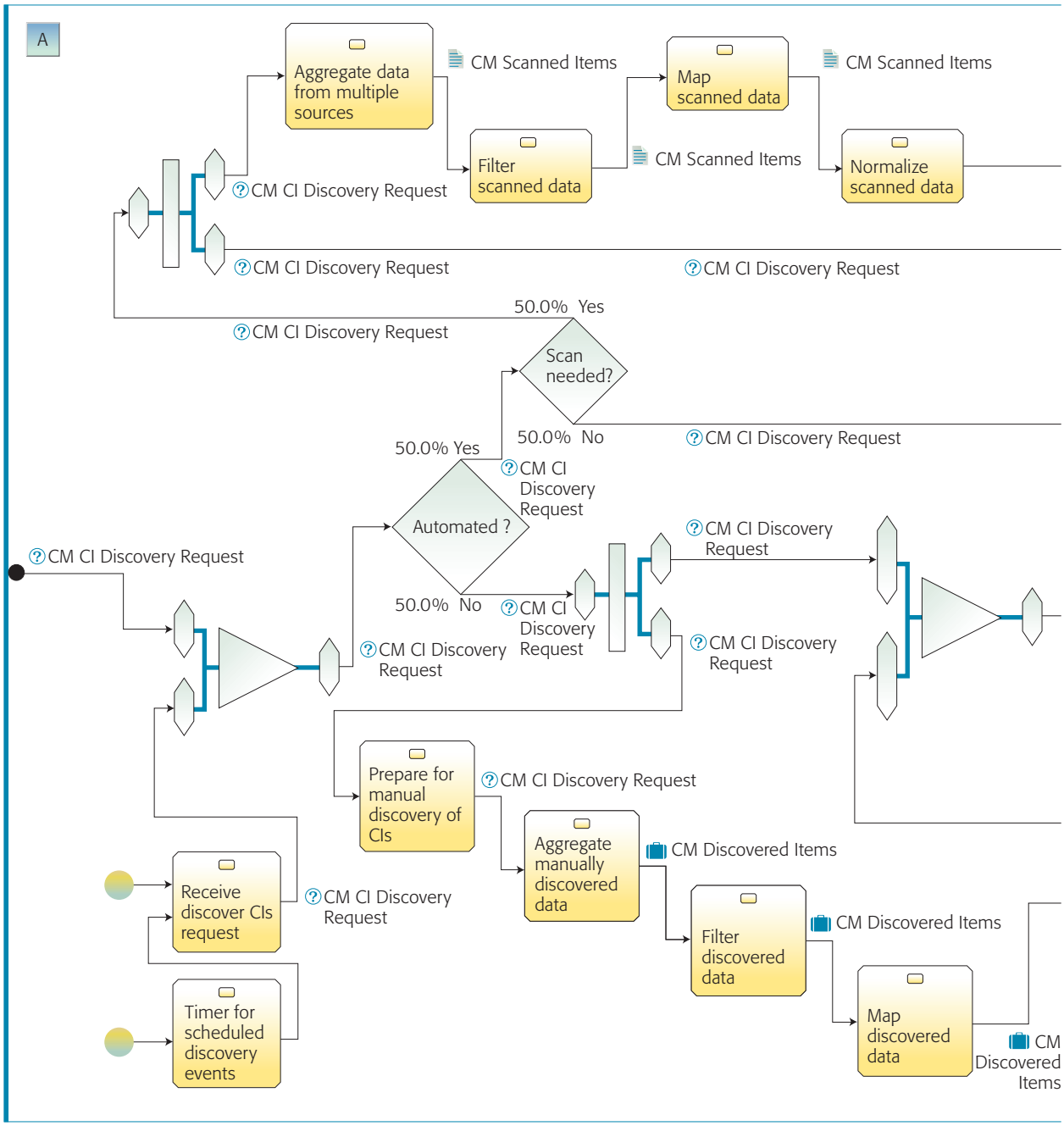
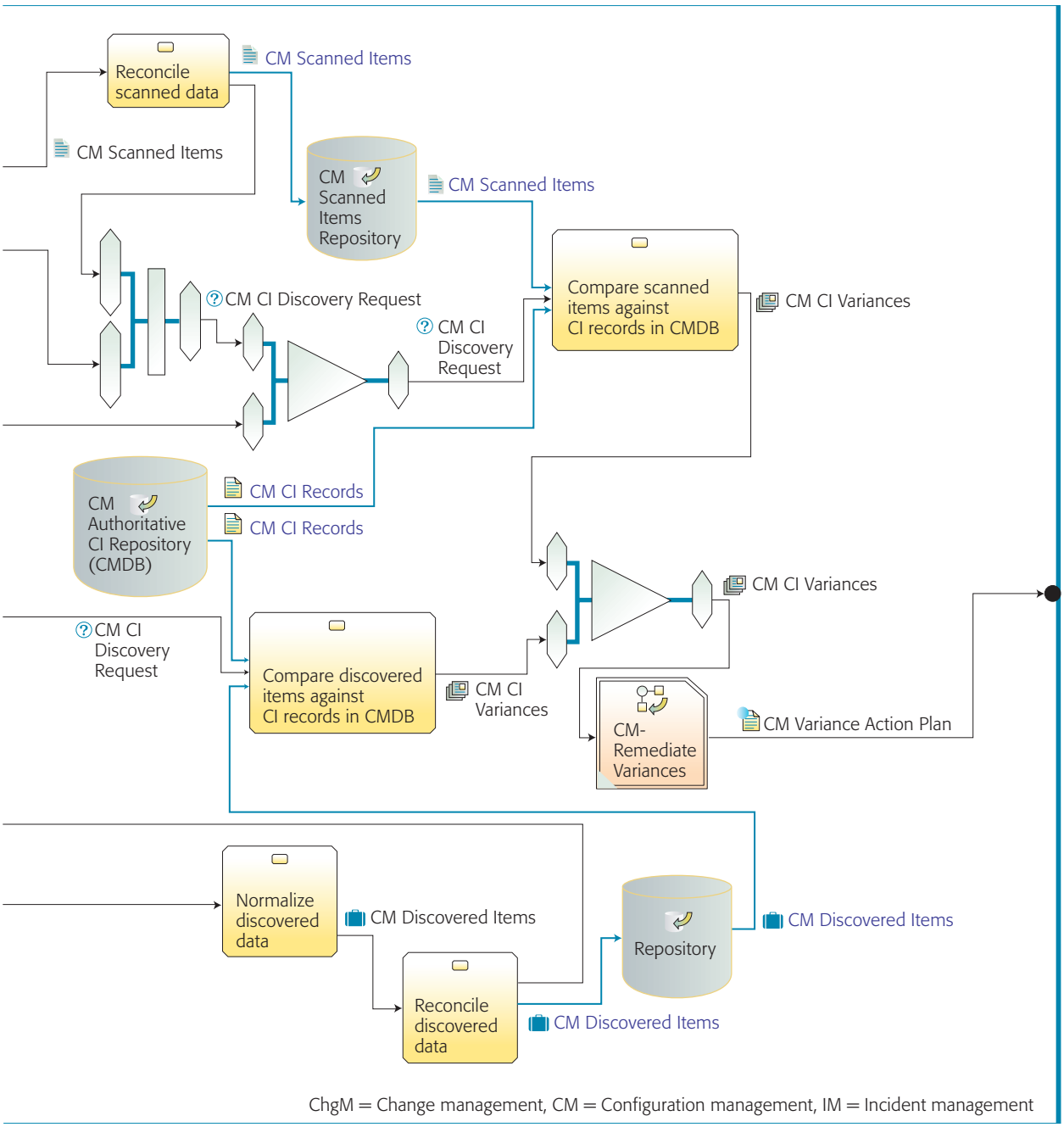


Figure 3A
Sample best-practice identify-CIs process

urgent RFCs are handled differently from regular RFCs. Other attributes of the change (e.g., customer) may also factor into deciding which activities are included. For example, in a multicustomer environment, the customer may have unique business requirements, such as regulatory requirements, that need to be accommodated in the handle-RFC process.

For complex changes that would benefit from release management, the coordinate-change-implementation and prepare-distribute-and-implement-change activities can invoke the release management processes. Overall control of a change, even if release management is employed, remains with change management.



CONFIGURATION-MANAGEMENT INSIGHTS

As a result of developing best practices for aspects of the service operations of the IBM Strategic Outsourcing Services division and developing the configuration-management processes for the IBM Tivoli CCMDB product, we gained many insights in the area of configuration management.

Authorized and actual representations

The primary objective of configuration management is to underpin the delivery of IT services by providing accurate data to all ITSM processes when and where it is needed. The configuration-management process takes advantage of CMDB, a database used to manage CI records throughout their life cycle. Changes to CI

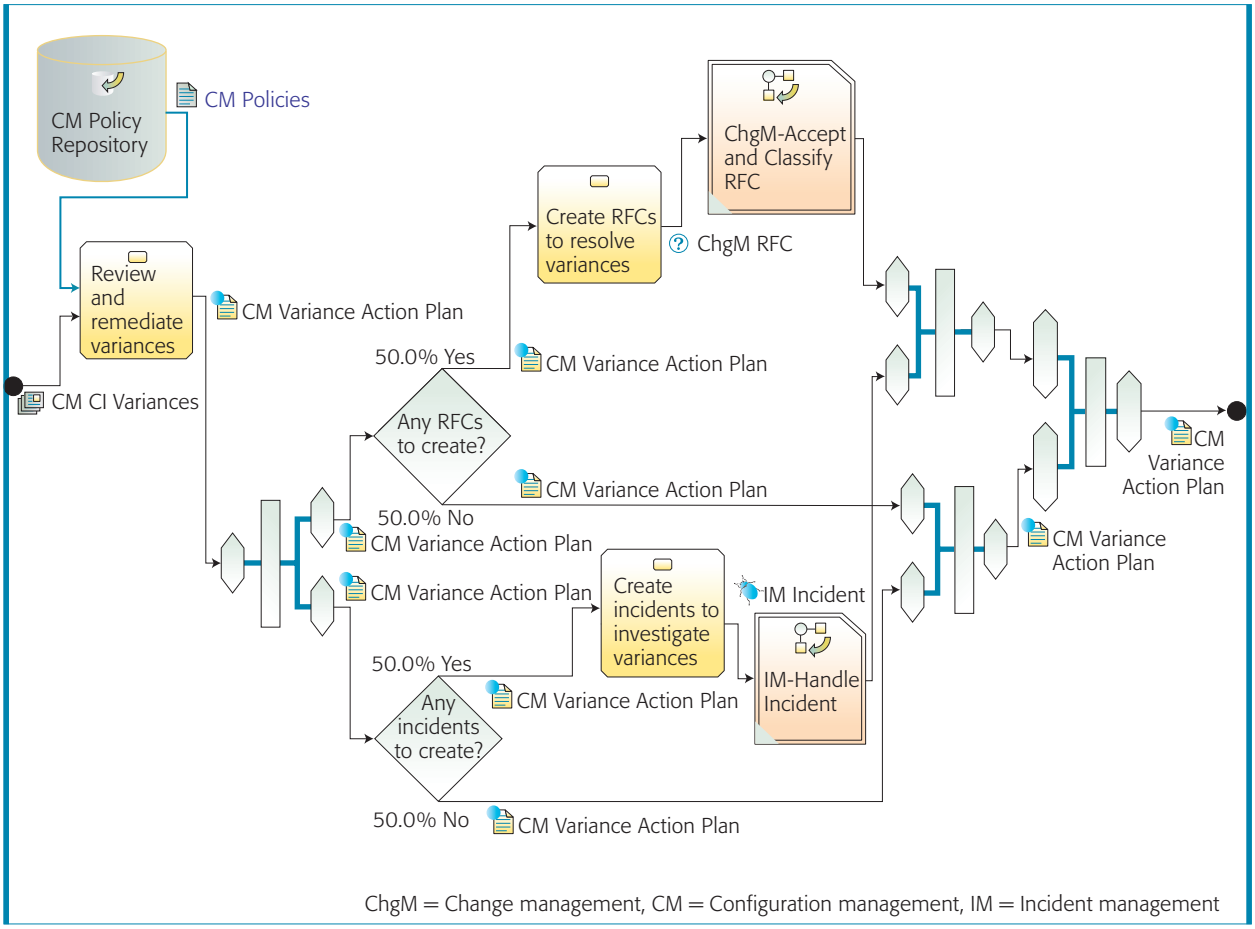


Figure 3B
Sample best-practice remediate-variances step of identify-CIs process

records may come from a variety of sources (e.g., discovery adapters, manual entry through a user interface, and bulk loads from applications) and as such must be controlled. Configuration control is concerned with ensuring that only authorized and identifiable CIs are recorded from receipt to disposal. It ensures that no CI is added, modified, replaced, or removed without appropriate controlling documentation, such as an approved change request.[28] The introduction of configuration control into the CMDB implies the need for a process which ensures that the necessary controlling documentation for an update is made before the update is input to the CMDB. Thus, the data reflected in the CMDB record for a CI may differ from the actual data for a CI as identified by tools such as discovery adapters.

We found that customers need both sets of data. They need control over their authorized environment and insight into the actual environment. In the event that the CMDB is used as a repository for both authorized data and actual data, we define the following:

- The *authorized representation* describes CI attributes (a subset of attributes for that type) updated by the control-CIs process called from the change-management process. These attributes have been approved in accordance with change control as reflected in the coordinate-change-implementation activity in the change-management process. This authorized representation is what ITIL refers to as the CMDB.
- The *actual representation* describes CI attributes (a subset of attributes for that type) according to the latest discovery-adapter uploads. These may record the same values or the values may be at variance with the authorized representation.

If the CMDB is maintaining both an authorized and an actual representation of the CIs, then there are security considerations to be addressed. A decision has to be made as to customer-defined access policies for these two representations and the relationship between them.

Configuration baselines

Customers appreciate configuration baselines. A configuration baseline is a snapshot at a specific instant of a set of CIs and their interrelationships. These instance statements about some subset of the environment are useful for documentation, recovery, and comparison purposes. Baselines should be named and time-stamped and should not be editable. Service providers can create a configuration baseline at the inception of a relationship with a new customer and periodically thereafter in order to document the initial state of the customer's environment, to document important checkpoint states (e.g., before a major change), to facilitate recovery in the event of a disaster, and to show change in the environment over time. The schedule for creation of baselines can be defined by customer-specified policies.

Gold standards

Customers find high value in process support around gold standards. Relationships between a gold standard and any number of sets of CIs can be created to establish the applicability of a gold standard to those sets of CIs. These gold standards are used by the configuration-management verify-and-audit-CIs process to assess compliance with established policy. For example, an organization might create a set of CI records that represents a typical UNIX** server configuration and designate this set as a gold standard for how all other UNIX servers in the organization should be configured. The organization can then create relationships between the CI records of the UNIX servers in the environment and this gold standard. When the verify-and-audit-CIs process is executed on those UNIX servers, a report of any compliance discrepancies between the gold standard and the associated CIs is generated. Alternately, gold standards can be described as a set of rules or policies against which CIs are compared.

CMDB population through discovery

To maintain an accurate CMDB, it is essential that there be a methodology and tooling to populate the actual (i.e., the gathered) data within the CMDB through discovery. This approach should express a systematic way to aggregate, filter, map, normalize, reconcile, prioritize, and load discovered data into the CMDB. Details regarding these steps are described in the context of the CMDB architecture in support of ITSM.[14]

Importance of remediation

As described earlier in the section "Verify and audit CIs," the activity of remediation is responsible for ensuring that the CIs reflected in the CMDB are an accurate reflection of the configuration of the established standards and the managed resources in accordance with necessary controlling documentation. This includes understanding variances between the discovered environment and the authorized environment and acting to remediate or correct any noted variances. Remediation (an activity in the verify-and-audit-CIs and identify-CIs processes) involves deciding how identified variances should be corrected. For example, a resource that is incorrectly configured, as identified by a discovery adapter, may result in the generation of a change request to reconfigure the resource. Conversely, a resource that is correctly configured but has incorrect authorized values (e.g., as a result of an error during manual entry) may result in a change to the authorized CIs within the CMDB tied to the appropriate controlling documentation. In general, the remediation activity (Figure 3B) called from within the verify-and-audit-CIs process may result in one or more incidents or change requests to correct errors exposed during the process.

Importance of CI life-cycle state

As part of configuration control, every CI managed by the CMDB has associated with it a life-cycle state. The life-cycle state is used for tracking and should be kept current and made available for planning, decision making, and managing changes to the defined configurations. Example states for a CI are: ordered, received, in acceptance test, live, under change, withdrawn, and disposed. *Figure 7* illustrates reasonable life-cycle states in a default configuration of CCMDB.

Transition between life-cycle states must be managed to ensure that a CI is only moved from a particular state to another legal state (Figure 7). In addition, again as part of configuration control, there is the notion of best-practice enforcement of attribute-level semantic validation during life-cycle

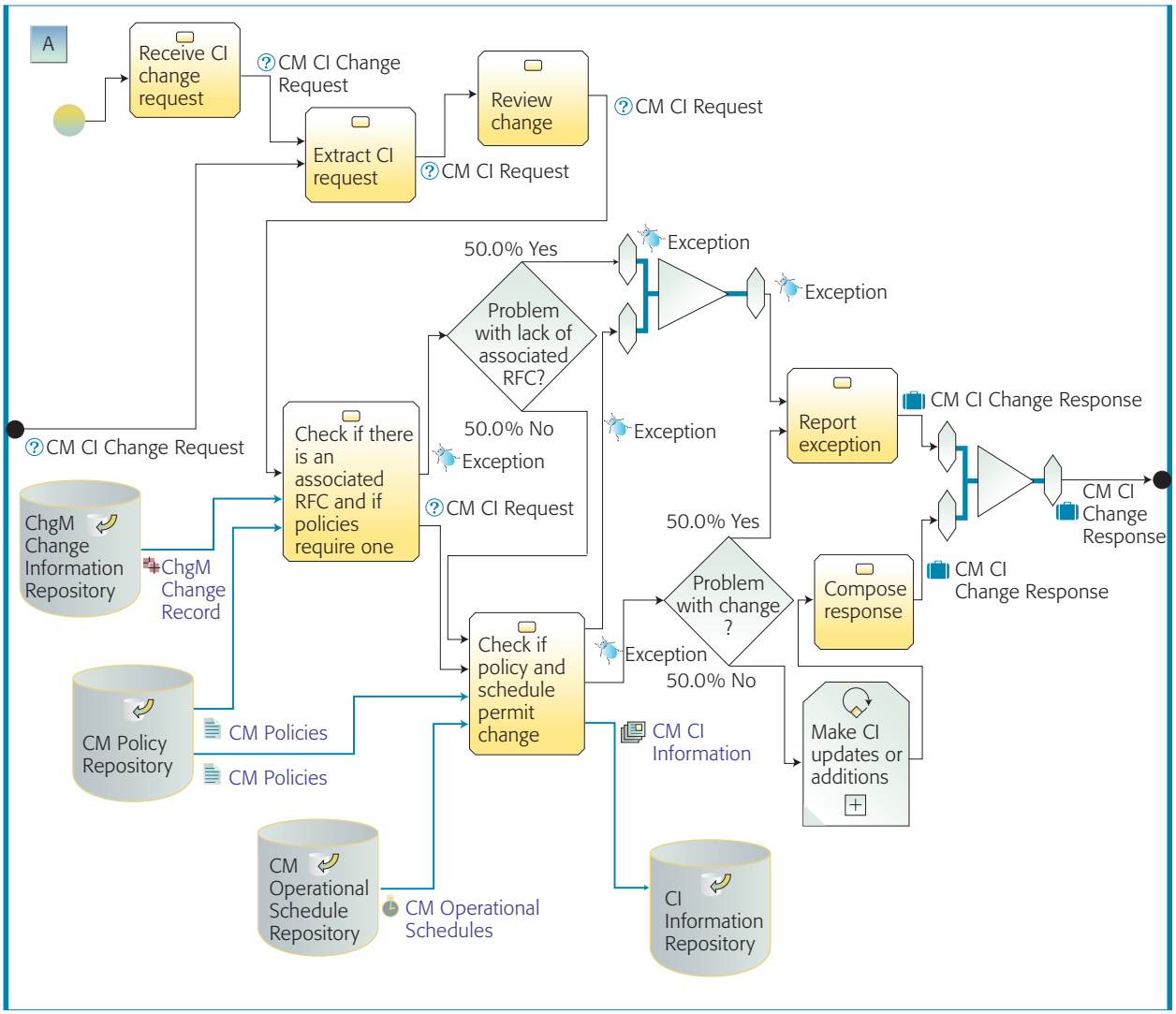


Figure 4A
Sample best-practice process to control CIs: step one, control CIs

state transition. There are three recommended life-cycle semantic validations:

1. For a particular CI type, designating that there are requirements that selected fields be populated with information before a particular state is entered or exited.
2. Designating selected states as "protected" (as shown in Figure 7) so that any changes to protected states necessitate that an RFC be associated with them. This validation capability recognizes that there are life-cycle states in which a greater degree of control is required than in other states, as described in the next section.
3. Separating state-transition enablement from other attribute changes to provide greater control over the circumstances in which the life-cycle state can be modified. This validation capability provides a greater degree of assurance that the life-cycle state of a CI is changed in accordance with best-practice intent by making a different application programming interface (API) and user experience accessible for changing the life-cycle state of the CI.

Life-cycle state history

The life-cycle state history of a CI is important for activities such as planning, decision making, and managing changes to the defined configurations.

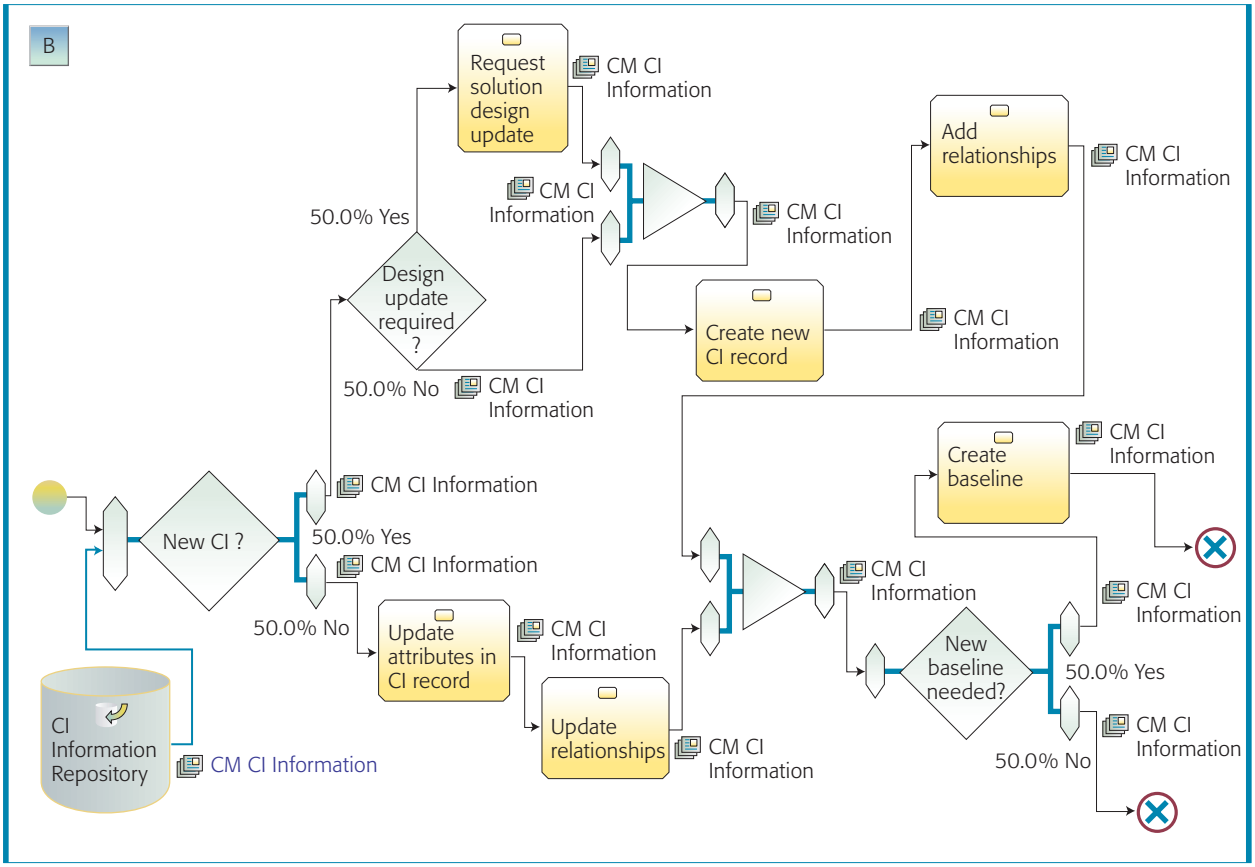


Figure 4B
Sample best-practice process to control CIs: step two, make CI updates or additions

Therefore, the entire life-cycle state history should be available for inspection for each CI. The life cycle history for a CI in a properly managed CMDB can be obtained from the change history for that CI because all authorized changes to the CI are recorded in the change history. As a convenience, it should be possible to review for a particular CI the life-cycle state history immediately without searching the change history. If the change history is periodically archived or deleted, then the change history cannot necessarily be relied on for a complete life cycle history.

Protected states

Given a state transition diagram, a set of the CI life-cycle states may be designated as protected states, affording a greater degree of control over the way in which they can be modified. The designation "Protected" implies that changes to the CMDB for CIs in this state must be associated with a change record (which results from an RFC), which serves as the controlling documentation; that is, for CIs in protected states, there necessarily exists controlling documentation. (In Figure 7, the life-cycle production and sunset states are designated "Protected".) Thus, for CIs in protected states, an association with a change record is required for any changes to the CI to take place. This includes making a transition from the life-cycle state of a CI into or out of a protected state.

A lightweight version of protected states can be imagined in which only the life-cycle-state attribute itself is protected; that is, changes to the life-cycle-state attribute into, between, or out of a protected state requires association with a change record; changes to other attributes by means of discovery are not protected; that is, they can be made without an associated RFC. This lightweight version provides partial configuration control (in that state transitions and user interface updates to CIs are

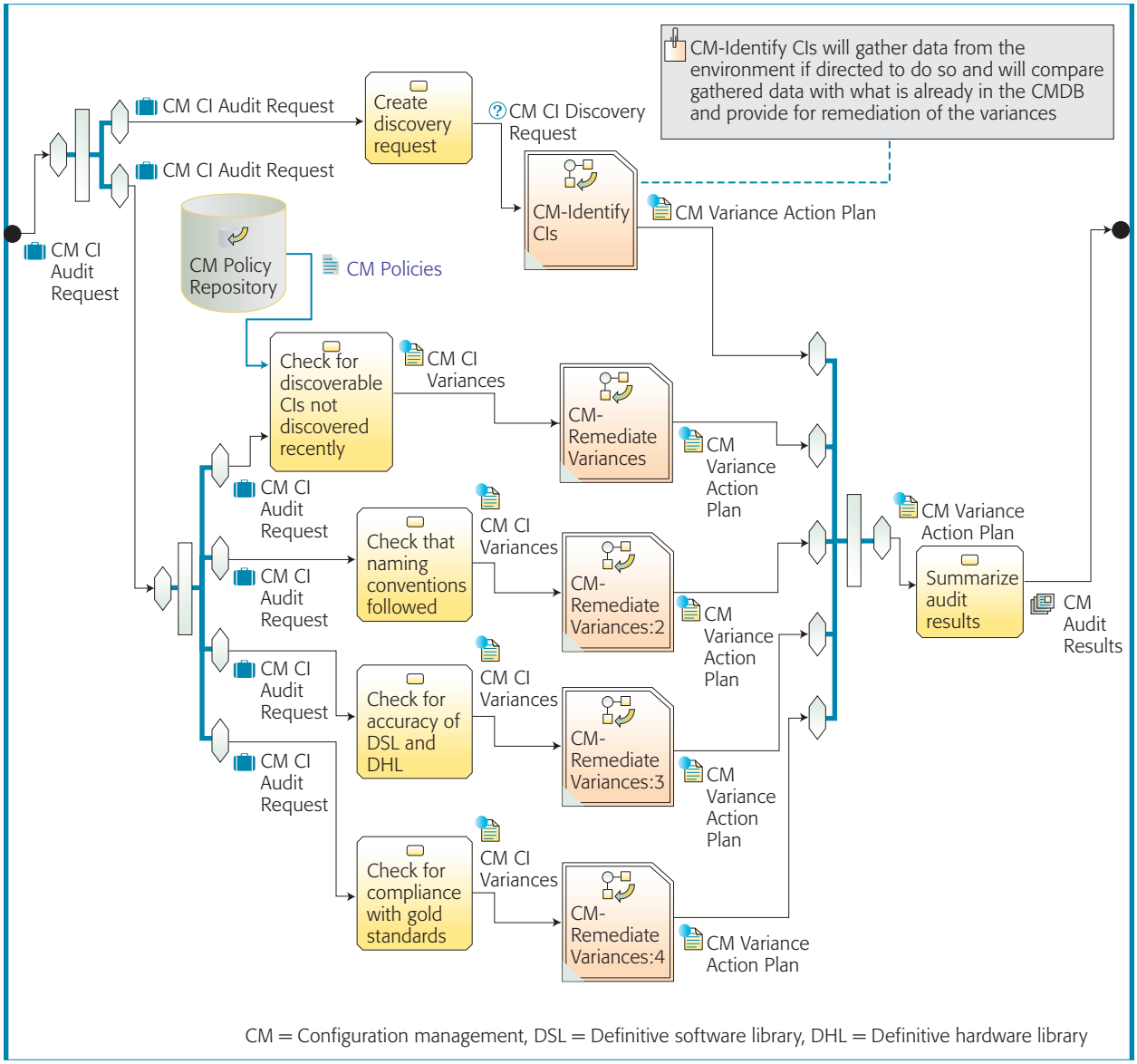


Figure 5
Sample best-practice process to verify and audit CIs

controlled), but is less desirable than full configuration control in which all attributes are protected.

Life-cycle state-diagram customization

The ability to customize the life-cycle state-transition graph of CIs, although not essential, provides significant benefits for configuration control and occurs in two contexts. First, each CI type may have a separate set of valid states and a separate life-cycle state-transition graph. This provides opportunities for rich semantic validation based on the type of the CI (e.g., the life cycle of a server can be described distinctly from that of a business application). Second, from a per-customer perspective, the life-cycle state-transition graph for each CI type may likewise be customized with valid states and life-cycle state-transition graphs (which can be called *multicustomer customization*). Additional details on the multicustomer aspect are provided later in the section "Service provider perspective."

Field enforcement and key fields

One element of semantic validation on life-cycle state transitions is field-level enforcement for

designated key fields. Field enforcement provides a method to validate that a particular field (attribute) or set of fields conforms to designated criteria before a transition can take place. For life-cycle state transitions, this implies that the life-cycle state of a CI cannot be changed unless the key fields conform to the designated criteria (e.g., a server cannot be placed in the production state unless the host-name field is populated). The notion of *key fields* is related to field enforcement. Key fields are those fields that are essential to some activity, in this case, key fields for state transition.

CI ownership compared with management responsibility

The ownership of a resource is distinct from the management responsibility for the resource. Both ownership and management responsibility information for a particular resource should be recorded in a CI (as should subsequent changes). The owner of the resource is the entity legally responsible for possession of the resource (i.e., owns it). The management responsibility for the resource defines who is charged with the administration of the resource. To provide a concrete example, a resource in an outsourcing agreement may be owned by ABC Corp., but be managed by XYZ Inc. on behalf of ABC. For management purposes, XYZ may have a separate account team designated to support the resources owned by company ABC. Of course, it is also possible for an entity to both own and have management responsibility for its resources. This topic is a precursor to the area of multicustomer support described later in the section "Service provider perspective."

Impact assessment: A key CMDB service

Assessing impact is a key step in several service support processes. Change management assesses the impact of a change on business processes, IT infrastructure, users, and the availability of resources. Problem management determines the urgency of a problem and its impact on the business and users, as measured by service level agreements (SLAs) in order to establish a priority and severity that affects the time and resources allocated for problem resolution. Incident management determines the urgency of an incident and its impact on the business and users as measured by SLAs in order to prioritize the order in which incidents are resolved and to minimize impact to the business and users. Impact assessment for each of these processes relies on the semantically rich CMDB to provide information on the relationships between CIs.

CHANGE-MANAGEMENT INSIGHTS

In the course of building the change-management Process Manager and interacting with customers, we gained several key insights, as described below.

Loosely coupled change and configuration management

Some customers have been using a change-management system that is completely decoupled from their configuration-management system. The core idea is to use a source-code defect-management system for change management. This approach can satisfy the goals of providing a formal approval process for change management and making it possible to customize additional attributes on RFCs that are customer specific.

However, this approach falls short in several ways. First, the CIs that are in the scope of the RFC cannot be specified as structured data. The list of CIs needs to be specified in the RFC description. Second, because the RFCs have no link to CMDB, impact analysis of an RFC becomes completely manual and error prone. And finally, after a change has been implemented, the change-management process cannot update the CIs in the CMDB. This step needs to be performed manually.

Therefore, we strongly recommend the use of a system that provides integrated change and configuration management.

Adoption of change management

In our experience, customers start adopting both configuration management and change management in parallel. Their initial focus in configuration management is to configure the discovery tools, create necessary filters and mapping to identify CIs in their IT environment, and produce a layered topology. In parallel, they focus on defining their change-management process. According to Gartner Inc., one key problem in process implementation is the erroneous assumption that the process definition included in the technology will satisfy their goals.[29]

Each customer has a unique organization and unique control procedures. These factors significantly influence the process reference models used by a customer. A readymade process can provide a

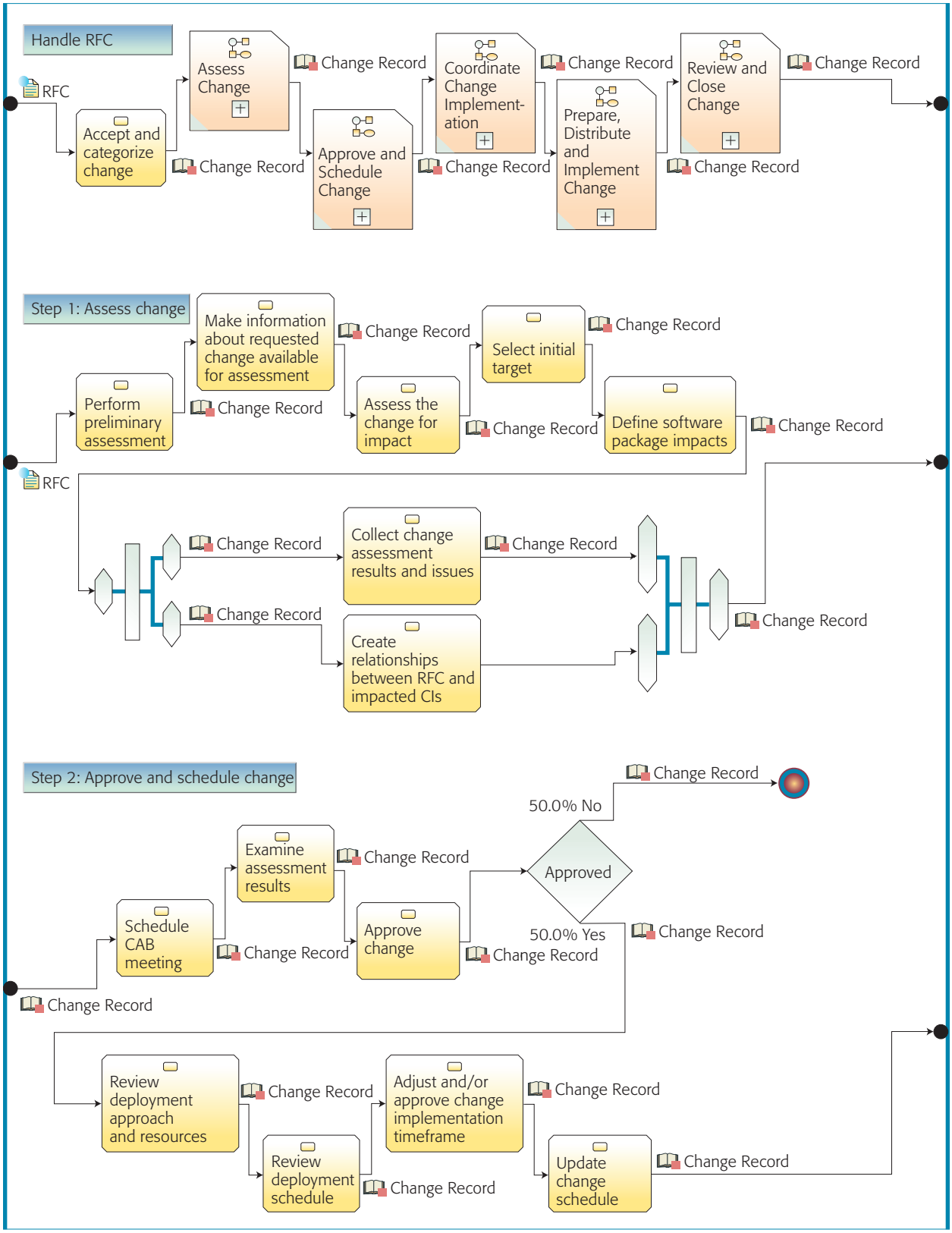


Figure 6
Overview of the sample best-practice process for handling a change and the five steps that comprise it (Steps 1-2)

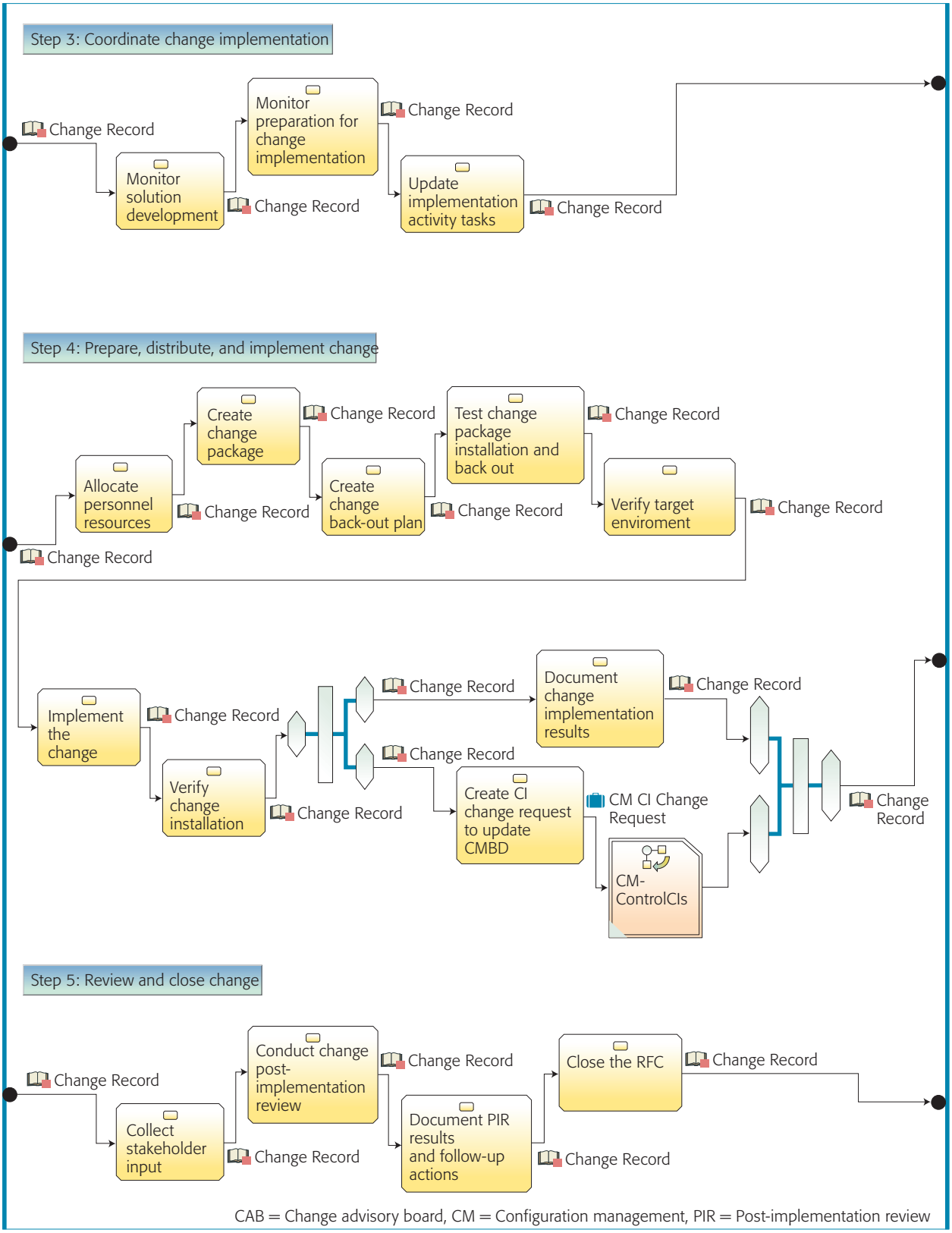


Figure 6 (Continued)
Overview of the sample best-practice process for handling a change and the five steps that comprise it (Steps 3-5)

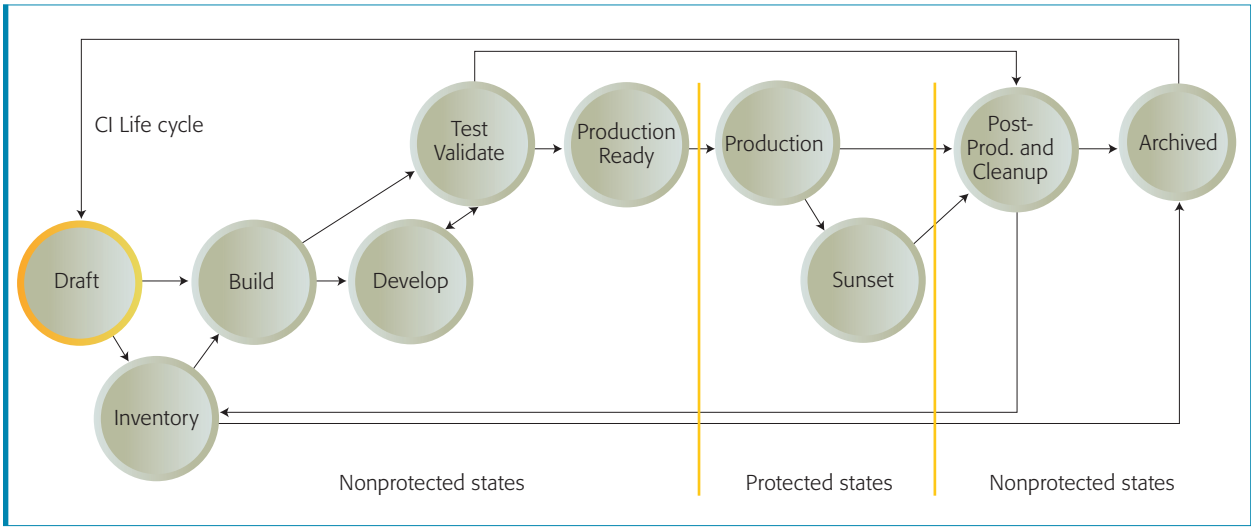


Figure 7
Life-cycle states

good start, but it rarely meets all customer requirements. The experience of one of our customers, a large insurance company, is an example. This customer had a complete process reference model for change and release management. However, the readymade process models had to be significantly customized in the field before they could meet this customer's requirements.

Some customers do not have a documented and approved change-management process. For these customers, creating new process reference models and gaining the approval of the IT staff can take a significant amount of time.

Compliance and automation

Every customer has a set of OMPs that are used to implement changes in its data center. For example, there are network-management OMPs, storage-management OMPs, and patch-management OMPs. Typically, the change-management system is used to approve and schedule changes, while the OMPs are used to implement changes.

Typically, integration between the change-management system and the OMPs is very limited. The change implementor needs to translate the work assigned by the change-management system into the OMP constructs. Depending on the level of manual intervention required, this translation can be error prone and inefficient. Additionally, it is harder to ensure that changes carried out by the OMP were indeed the changes approved by the change-management process.

In our experience, large banks are keenly interested in increased integration between the change-management system and OMPs. For example, they would like to automate the patch distribution step in the change-management process so that they can prove the authorized patch was distributed to the authorized set of CIs.

RFC data customization

Change process reference models typically require the addition of unique attributes to the RFCs. For example, some customers require new attributes, such as "Justification" and "Impact of not implementing the RFC." A change-management tool must provide an efficient way to support this type of attribute customization.

Another critical requirement is the ability to produce reports based on these custom attributes. For example, a customer who adds a new attribute to reflect risk may want to be able to produce a report on high-risk changes done to the billing application during a specific time period.

Automating remediation

Each IT environment or data center has several types of CIs, and each type of CI requires certain

types of changes. For example, RFC types for a CI server type might include such types as distribute patch, add storage, add hardware, install application, and upgrade application. The number of change types in a data center can be very large.

Each change type requires unique attributes in the RFC. For example, a distribute-patch RFC may require a patch-number attribute, and an add-storage RFC may require an amount-of-storage attribute. These unique attributes in the RFC can be used to automate the updating of the CMDB following the implementation of a change. For example, the amount-of-storage value in the add-storage RFC can be used to automatically update the authorized representation of the server CI after the change has been implemented.

However, in some cases change management is unable to know the actual changes made to the IT infrastructure by the OMP system. For example, if a new set of software packages has been installed on a server, a new discovery scan needs to be run on the server to identify the changes that were made to the server hardware and software. In such cases, the change-management process may explicitly trigger a verify-and-audit-CI configuration-management process on the server CI in order to discover the changes made and update the CMDB appropriately.

Automatic assignment of tasks and approvals

The volume of RFCs in an organization can be large. One customer, for example, reports 2000 RFCs every month. Therefore, customers are very sensitive about introducing new inefficiencies due to the change-management process.

One area of concern is manual assignment of approvals and tasks to people. A change-management system needs to provide a flexible mechanism to assign task ownership automatically. One pattern is to use information stored in the CMDB to assign tasks to individuals or groups of people. For example, the customer could preconfigure the individuals or groups who would perform the change-management roles (such as the change manager and change approver). The roles are associated with changes to each business CI (a billing application is an example of a business CI). Then, when a change is requested and the change requestor specifies a business CI, the change process could be instantiated for this change, with the process tasks assigned to the individuals or groups specified in the business CI.

Scheduling changes

Scheduling a change is a complex task that requires several pieces of information. One must know which CIs will be impacted, the SLAs on the impacted CIs, previously scheduled changes on these CIs, and the availability of skilled resources.

Few change-management systems provide a comprehensive but usable task-scheduling mechanism. An integrated change and configuration-management system is fundamental to achieving this goal.

Target CIs

An RFC may be associated with a large number of target CIs (i.e., the CIs that will be impacted by the RFC). For example, the target CIs for a distribute-patch RFC is the list of servers on which a patch will be installed.

A change requestor should be able to choose a CI collection to specify the CIs for an RFC. During impact analysis and change-implementation planning, the change manager needs a way to organize the CIs hierarchically. For example, CIs may be organized by location, by owners of a business application, or by the CI owner. This organization of target CIs evolves as the process moves along. For example, CIs specified during RFC creation may be refined during impact analysis. Some change process implementations choose to freeze the target CIs once the RFC has been approved

Leveraging an SOA

It is essential that the services provided by change management and those provided by configuration management fully exploit the concepts embodied in SOA.[22] In particular, customers expect that accessible change-management service-interaction points (both command and status related) be made available as SOA services. For example, an RFC may be received by a variety of CCMDB-supported entry points, such as a graphical user interface and including a well-documented Web Services interface.

SERVICE PROVIDER PERSPECTIVE

ITSM best practices have gained wide acceptance because they help the enterprise manage the provisioning of IT services according to business-

based objectives for quality and cost.[12] Furthermore, ITSM best practices are important to service providers to manage both their own business and the IT of their customers.[5] Similarly, outsourcing customers are also increasingly interested in best-practice IT services from their service providers, and in today's highly competitive marketplace, the outsourcing industry has been pressured to find ways to reduce cost in its key business functions.

When first pressured to reduce cost, service providers used standard cost-containment methods. However, it has become increasingly difficult to reduce the service-delivery and service-outsourcing operating cost without examining the hardware and software used to run the business to determine if there are gains in efficiency or reductions in cost that can be achieved in these areas. Advances in hardware and operating-system management have allowed hardware and resource sharing to help reduce delivery overhead somewhat, as exemplified by the IBM eServer* p690 series of AIX* servers or the IBM xSeries* Blade server technology, which help the service provider increase the utilization of the resources already available by allocating idle resources when and where demand requires. However, this type of idle utilization principle addresses only hardware sharing, while still requiring full software implementation. It also requires that the service delivery team maintain individual support contracts from each infrastructure instance with only marginal savings. Most enterprise-level applications require an individual infrastructure, which includes hardware, software, and service costs, and it must be created and maintained on behalf of each outsourcing customer.

What service providers require are applications for which a single instance can support multiple customers. This is especially true of those ITSM applications employed by a service provider to manage not only its IT environment and services but also those it provides to customers as well. However, the ability to support multiple customers in an ITSM application such as a configuration- and change-management system introduces additional challenges, such as absolute assurance on data segregation and security considerations. It also introduces the need to define repeatable processes to consistently manage the collection, maintenance, auditing, and availability of centralized data. When a service provider offers global solutions to a diverse set of customers, it needs to develop and manage global delivery methods and global processes, both of which must accommodate cultural, national, and regulatory factors. Thus, the service provider needs a common, flexible, and scalable tool to provide a transparent technology layer through which data and processes can be customized as required or as centralized governance allows. For example, a change- and configuration-management solution should provide default change- and configuration-management best-practice-based processes that can be customized at the customer level, or even at the level of the customer's organizations.

A truly multitenant change- and configuration-management solution that meets the needs of service providers has many challenges to overcome. These challenges will likely be addressed gradually over time. The ability to customize processes is important, but clearly less critical than the ability to provide flexible data segregation and access control. The features needed by service providers will also greatly benefit enterprises that choose to separately manage their enterprise internal divisions.

CONCLUSION

Today's competitive business climate, the complexity of IT environments, and the criticality of IT to a company's success dictate the use of industry best practices, practices that enable an organization, service provider, or outsourcing customer to manage their IT environment according to business objectives for cost and quality. ITIL, developed by the United Kingdom Office of Government Commerce in collaboration with many industry leaders, provides valuable insights into the organization of a CMDB and the configuration and change-management processes that support it. These insights require additional reflection when applied to an ITSM platform, where the processes must be actually implemented in a service provider context; that is, for an enterprise whose business is the IT management of other enterprises. In this paper we introduced a change- and configuration-management process that conforms to ITIL, provided additional insights based on customer experiences, discussed how the processes relate to one another, and introduced additional considerations that must be addressed to implement such a solution in a service provider environment. Our experience from customer engagements shows that customers benefit by having a CCMDB that conforms to ITIL and

from the cost and efficiency improvements accrued by taking advantage of a service-provider-managed solution. Our experience also shows that customers demand a comprehensive solution to address their data segregation and customization concerns. Similar benefits can be realized by an enterprise that chooses to manage internal divisions separately.

As we look to the future, we see CCMDB functionality becomingly increasingly important to IBM Service Management. In addition to the natural and inevitable exploitation of CCMDB functionality within an SOA, there will be an increasing need for an extended set of managed elements (such as policies and managed-service artifacts) and ever more sophisticated relationship and gold-standard analyses, all integrated with policy-directed control, actions, and alerts. Given this, an enterprise might finally gain control of its IT infrastructure.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of the United Kingdom Office of Government Commerce, Information Systems Audit and Control Association, or The Open Group in the United States, other countries, or both.

CITED REFERENCES

1. Information Technology Infrastructure Library (ITIL), U.K. Office of Government Commerce, http://www.itil.org.uk.
2. U.K. Office of Government Commerce, "A Code of Practice for IT Service Management," in *Service Support, ITIL Managing Services*, Stationery Office, London, United Kingdom (2005), Section 1.9, http://www.tsoshop.co.uk/bookstore.asp?FO=1159966&Action=Book&ProductID=0113300158.
3. ISO/IEC 20000-1:2005, Information Technology—Service Management Specification, International Organization for Standardization and the International Electrotechnical Commission (2005).
4. R. J. Colville, *CMDB or Configuration Database: Know the Difference*, RAS Core Research Note G00137125, Gartner, Inc., Stamford, CT 06904 (March 2006), http://mediaproducts.gartner.com/gc/reprints/ibm/external/article5/article5.html.
5. M. Ernest and J. M. Nisavic, "Adding Value to the IT Organization with the Component Business Model," *IBM Systems Journal* **46**, No. 3, 387–403 (2007, this issue).
6. M. B. Chrissis, M. Konrad, and S. Shrum, *Guidelines for Process Integration and Product Improvement*, Addison-Wesley Professional, Boston, MA (2003).
7. Carnegie Mellon University Software Engineering Institute, *The Capability Maturity Model: Guidelines for Improving the Software Process*, M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis, Editors, Addison-Wesley Professional, Boston, MA (1995).
8. E. Guldentops, "Governing Information Technology Through COBIT," *Proceedings of the IFIP TC11/WG11.5 4th Working Conference on Integrity, Internal Control and Security in Information Systems: Connecting Governance and Technology*, Brussels, Belgium (2001), pp. 115–160.
9. Control Objectives for Information and Related Technology (COBIT), Information Systems Audit and Control Association, http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981.
10. F. Niessink and H. Van Vliet, "Towards Mature IT Services," *Software Process: Improvement and Practice* **4**, No. 2, 55–71 (1998).
11. A. Cater-Steel, W. Tan, and M. Toleman, "Challenge of Adopting Multiple Process Improvement Frameworks," *Proceedings of the 14th European Conference on Information Systems*, Göteborg, Sweden (2006), pp. 12–14.
12. A. Hochstein, R. Zarnekow, and W. Brenner, "ITIL as a Common Practice Reference Model for IT Service Management: Formal Assessment and Implications for Practice," *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*, Hong Kong, China (2005), pp. 704–710.
13. R. Conradi and B. Westfechtel, "Version Models for Software Configuration Management," *ACM Computing Surveys* **30**, No. 2, 232–282 (1998).
14. D. Lindquist, H. Madduri, C. J. Paul, and B. Rajaraman, "IBM Service Management Architecture," *IBM Systems Journal* **46**, No. 3, 423–440 (2007, this issue).
15. H. Madduri, S. S. B. Shi, R. Baker, N. Ayachitula, L. Shwartz, M. Surendra, C. Corley, M. Benantar, and S. Patel, "A Configuration Management Database Architecture in Support of IBM Service Management," *IBM Systems Journal* **46**, No. 3, 441–457 (2007, this issue).
16. C. J. Paul, "The Process of Building a Process Manager: Architecture and Design Patterns," *IBM Systems Journal* **46**, No. 3, 479–495 (2007, this issue).
17. V. A. Danciu and B. Kempter, "From Processes to Policies—Concepts for Large Scale Policy Generation," *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, Seoul, Korea (2004), pp. 17–30.
18. J. Kramer and J. Magee, "The Evolving Philosophers Problem: Dynamic Change Management," *IEEE Transactions on Software Engineering* **16**, No. 11, 1293–1306 (1990).
19. A. Keller, J. L. Hellerstein, J. L. Wolf, K.-L. Wu, and V. Krishnan, "The CHAMPS System: Change Management with Planning and Scheduling," *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, Seoul, Korea (2004), pp. 395–408.
20. A. Keller, "Automating the Change Management Process with Electronic Contracts," *Proceedings of the 7th IEEE International Conference on E-Commerce Technology Workshops*, Munich, Germany (2005), pp. 99–108.
21. *Sarbanes-Oxley Act of 2002*, Public Law 107-204 (116 Statute 745), United States Senate and House of Representatives in Congress (2002).
22. C. Mayerl, T. Vogel, and S. Abeck, "SOA-Based Integration of IT Service Management Applications," *Proceedings of the IEEE International Conference on Web Services*, Orlando, FL (2005), pp. 785–786.

23. N. Joshi, W. Riley, J. Schneider, and Y.-S. Tan, "Integration of Domain-Specific IT Processes and Tools in IBM Service Management," *IBM Systems Journal* **46**, No. 3, 497–511 (2007, this issue).

24. A. G. Ganek and T. A. Corbi, "The Dawning of the Autonomic Computing Era," *IBM Systems Journal* **42**, No. 1, 5–18 (2003).

25. M. W. Johnson, A. Hately, B. A. Miller, and R. Orr, "Evolving Standards for IT Service Management," *IBM Systems Journal* **46**, No. 3, 583–597 (2007, this issue).

26. U.K. Office of Government Commerce, "A Code of Practice for IT Service Management," in *Service Support, ITIL Managing Services*, Stationery Office, London, United Kingdom (2005), Section 7.8, http://www.tsoshop.co.uk/bookstore.asp?FO=1159966&Action=Book&ProductID=0113300158.

27. IBM Tivoli Unified Process, IBM Corporation, http://www.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html.

28. U.K. Office of Government Commerce, "A Code of Practice for IT Service Management," in *Service Support, ITIL Managing Services*, Stationery Office, London, United Kingdom (2005), Section 7.2, http://www.tsoshop.co.uk/bookstore.asp?FO=1159966&Action=Book&ProductID=0113300158.

29. S. Mingay and S. Bittinger, "Don't Just Implement CMMI and ITIL: Improve Services," Research Note G00136578, Gartner, Inc., Stamford, CT 06904 (December 2005).

Accepted for publication December 20, 2006.
Published online July 11, 2007.

Christopher Ward
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (cw1@us.ibm.com). Dr. Ward is a research staff member and manager in the Service Delivery department. He has a Ph.D. degree in computer science from the University of Florida. He joined IBM in 2000 and is most recently responsible for innovative functionality in the configuration-management process for the IBM Tivoli ITSM-based CCMDB product. Dr. Ward has published over 50 papers addressing a variety of computer science problems, is author or coauthor of numerous patents, and is a Senior Member of the IEEE.

Vijay Aggarwal
IBM Software Group, 11501 Burnet Road, Austin, Texas 78758 (aggarwav@us.ibm.com). Mr. Aggarwal is a senior software engineer. He is currently the design lead for a set of IT infrastructure library process managers. He received an M.S. degree in computer science from the Indian Institute of Technology, Kanpur, India. His focus is in building enterprise-system management products. He has provided technical leadership in the areas of network management, storage management, provisioning automation, serviceability, and process management.

Melissa Buco
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (mjbuco@us.ibm.com). Ms. Buco is a senior software engineer in the Service Delivery department. She has a B.A. degree in mathematics from Northeastern University and an M.S. degree in computer science from Columbia University. Her recent focus has been in the area of IT service management. She contributed to the configuration-management process for the IBM Tivoli CCMDB. She has also worked in the areas of SLA management, software engineering, emergency management, project management, and workflow. Ms. Buco received an IBM Outstanding Technical Achievement award and has received several IBM Research Division and Invention Achievement awards.

Emi Olsson
IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (emio@us.ibm.com). Ms. Olsson is a senior architect for server systems operations and lead architect for the global configuration-management solution. She has provided service for a diverse array of over 200 accounts. She has been working in the configuration-management field for six years with the goal of providing quality data to other disciplines. Most recently, Ms. Olsson was instrumental in bringing the IBM Software Group, Research, and Integrated Technology Delivery (ITD) organizations together to partner on the IBM Tivoli CCMDB product. Under her architectural direction, ITD began offering Tivoli CCMDB as a service in August 2006.

Steve Weinberger
IBM Integrated Technology Delivery, 4499 Fisher Road, Columbus, Ohio 43228 (sweinb1@us.ibm.com). Mr. Weinberger is an IT architect focused on the configuration-management sector of IBM Global Services, Strategic Outsourcing Services division. He joined IBM in 1997 and is most recently responsible for leading the technical architecture for the global configuration-management solution based on the IBM Tivoli ITSM suite of products. Global standardization of configuration-management processing and tooling has been a driving force in Mr. Weinberger's career for the past several years. For this work, he was recognized with an IBM Outstanding Technical Achievement Award. ■