Evolving standards for IT service management

M. W. Johnson
A. Hately
B. A. Miller
R. Orr

In this paper we describe standards and widely adopted best practices that facilitate the deployment of information technology service management (ITSM). We cover the Information Technology Infrastructure Library® (ITIL®) framework of best practices for delivering information technology (IT) services. As part of ITIL we discuss the central role played by the configuration management database (CMDB). Then we describe the CMDB federation specification, an emerging standard for federating data repositories in support of a CMDB. We discuss two standards for representing management data and constraints on those data: the Service Modeling Language (SML) and the Solution Deployment Descriptor (SDD). Finally, we describe how related but incompatible Web services standards are being unified into a consistent set of standards.

INTRODUCTION

In recent years businesses have become more responsive to customer demands and more adept at seizing new business opportunities. This evolution has been enabled by advances in information technology (IT), and enterprises worldwide are increasingly reliant on IT services to address requirements of both external clients and internal users. These services are built on an IT infrastructure that incorporates advanced technologies. The resulting web of technology and services is complex and dynamic, and changes to update services or refresh the technology are frequent.

Changes to IT services or the supporting infrastructure must be closely managed in order to avoid disruptions. Several administrators and operators may share the authority to plan and carry out changes to the same set of IT components. If they do not coordinate their activities, they may inadvertently interfere with each other. Thus, changes to service and infrastructure configuration must be controlled through a change management process. When failures occur, the administrators and operators must record, analyze, and address the incidents in a timely manner.

To support new business processes and to address the challenges of cost, complexity, and compliance with governmental regulations, many IT organizations are implementing a comprehensive approach to management—information technology service

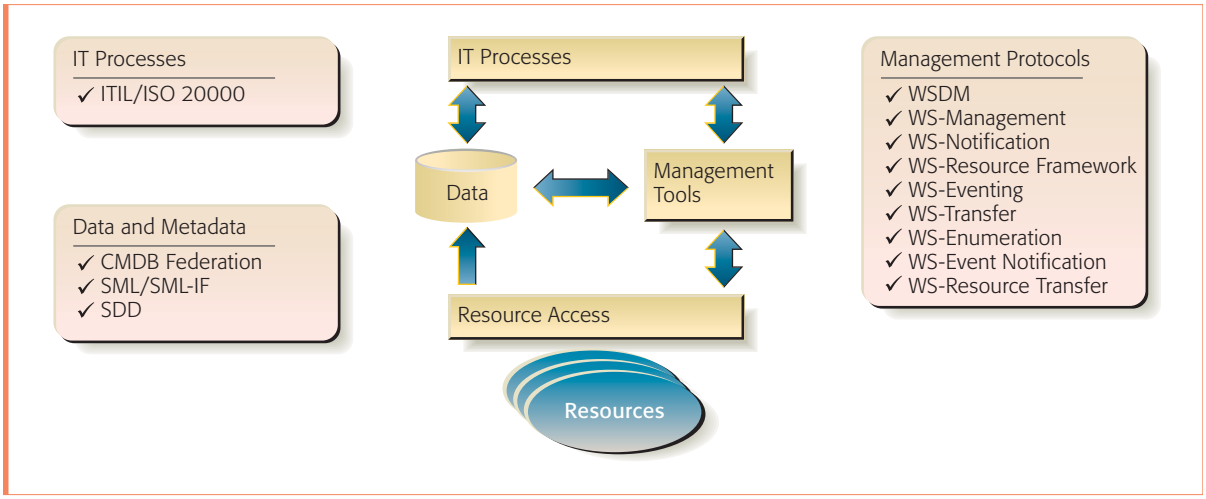

Figure 1
ITSM standards

management (ITSM). In this approach, instead of focusing on technology and IT systems, we focus on aligning IT services with business objectives and strive to optimize the performance of the entire business organization.

IT organizations are examining how to transform their existing IT infrastructure and processes to ITSM. Typically, an increased use of well-defined processes, integration of those processes, and an architecture that supports the transformation is required. In support of the transformation, IT organizations recognize that standards-based solutions enable them to improve the interconnectivity of IT components and exploit new technology with the associated cost savings. Basing solutions on standards helps achieve goals sooner and with less risk; standards facilitate the interoperability needed to connect internal and external applications and data; standards provide the ability to rapidly integrate new hardware and software into existing infrastructure. A comprehensive approach to ITSM leverages standards for information, processes, and services so that people and technology can interact effectively and efficiently. As such, standards are essential elements of IT.[1]

Not every standard that is relevant for IT management is necessarily produced and ratified by an accredited standards body. *De jure* (literally "by right") standards are produced by bodies that have assumed authority to issue standards. Whether this authority is granted by government, international agreements, or industry agreements, it is widely acknowledged that the organization has the authority to issue standards within its domain. Examples of de jure standards bodies relevant to ITSM include the International Organization for Standardization (ISO), the Internet Engineering Task Force (IETF), the Worldwide Web Consortium (W3C**), the Organization for the Advancement of Structured Information Standards (OASIS**), and the Distributed Management Task Force (DMTF). *De facto* (literally, "by fact") standards are those that acquire the attributes of standards by virtue of becoming widely used. They have wide industry acceptance and represent significant investments by companies. A de facto standard may later be adopted as a de jure standard.

In this paper, we focus on existing or emerging standards that are likely to impact ITSM. We present these standards in the context of an architecture for ITSM shown in ***Figure 1*** (see Reference 2 for an in-depth description of this architecture). The standards we discuss cover three areas: IT processes, data/metadata, and management protocols.

The Information Technology Infrastructure Library** (ITIL**),[3–5] a set of process-based best practices for the management of IT services, was developed in the United Kingdom Office of Government Commerce. The International Organization for Standardization published ISO/IEC 20000-1:2005,[6]

commonly known as ISO 20000, which formalizes the ITIL best practices by establishing certification requirements. ITSM solutions benefit greatly from using a coherent and robust process framework such as ITIL. ITIL defines processes that enable IT organizations to efficiently and reliably manage services and to satisfy performance, availability, and cost objectives. For example, ITIL defines a change-management process that starts with a user's submission of a request for change (RFC) and includes the steps required to analyze the change and plan its implementation so as to avoid unacceptable impact to other services and to ensure that all changes are properly authorized.

ITSM solutions use one or more data repositories (labeled Data in Figure 1) in which data shared by processes and other management software are stored. The data (and metadata) stored may include not only the *observed* data related to IT components and their relationships but also the *authorized* (expected) version of the same data. We discuss three data-related emerging standards: the configuration-management-database (CMDB) federation specification, the Service Modeling Language (SML), and the Solution Deployment Descriptor (SDD).

The CMDB federation specification[7] is an emerging standard describing how management data repositories can interact with each other to appear to external clients as a federated CMDB and how clients may access this data. The CMDB federation specification defines the interfaces to combine data from multiple sources into a single view based on reconciling resource identities or relating management data or both. For example, multiple management tools may manage the same resource, each assigning an identity to the resource. IT processes that do not understand that the data are about the same resource may inadvertently interfere with each other, leading to problems and instability in the IT infrastructure. Through use of the CMDB federation specification, different identities can be related to one another, with this single view leading to more reliable and predictable processes.

The Service Modeling Language (SML)[8] is an emerging standard that specifies extensions to Extensible Markup Language (XML) schema to describe IT resources and their interrelationships. A companion specification, SML Interchange Format (SML-IF),[9] describes how to represent an SML model in a standard way for interchange. Use of SML and SML-IF helps to integrate management tools and processes, even though their underlying technologies differ significantly. Decoupling the implementations gives IT organizations more flexibility to choose components that offer the best solution without sacrificing the integration and consistency goals in ITSM implementations.

The SDD[10] is an emerging standard from OASIS for representing installable software packages and their configuration, dependency, and life-cycle information. This information is used to automate manual tasks in the deployment of software solutions. Without a standard such as SDD, the IT staff must understand how each software package needs to be connected and configured. This is a significant burden and is often not reliable because there is insufficient knowledge about the structure and requirements of the software package. ITSM solutions achieve greater reliability and efficiency when developers who best understand each software package provide the necessary information explicitly by using SDD.

A number of standards based on Web services[11–13] have been developed for managing resources in the IT environment. We focus here on proposed standards that will harmonize similar but incompatible families of standards related to Web Services Distributed Management (WSDM) from OASIS and Web Services for Management (WS-Management) from DMTF. These protocols can be used independently or together to manage resources in heterogeneous environments. For example, a component that implements an access layer to a resource may implement the WS-ResourceTransfer and WS-EventNotification specifications. A management tool could use WS-EventNotification to subscribe to resource state changes. After receiving a notification from the resource access component, it could use WS-ResourceTransfer to retrieve detailed resource properties or to reconfigure the resource. Because they can improve interoperability among components without constraining the component implementations, management protocols based on Web services are popular in ITSM solutions.

The rest of the paper is organized as follows. The next five sections deal with specific standards: ITIL, the CMDB federation specification, SML, SDD, and Web-services-based protocols. In the next-to-last

section we present some thoughts on the reviewed standards. The last section is the conclusion.

ITIL/ISO 20000

A fundamental goal of ITSM is the management of IT services and infrastructure with the same kinds of quality control that enterprises strive to use for all business processes. When this is achieved, businesses have the confidence to deploy new and updated services that are critical to their missions. A well-accepted way to achieve this is to manage through a process framework.

The most widely known and used process framework for managing IT services and infrastructure, ITIL is a set of best practices for aligning IT management with business requirements. Adopting ITIL is likely to lead to improvements in service quality and to lower costs for provisioning and managing IT services.

The current ITIL Version 2 consists of a set of publications, each of which describes best practices for some aspect of IT management. In this paper we focus on service management, which consists of service support and service delivery. These are the two most widely used parts of ITIL; they are the basis of ISO 20000, and they are the basis of the syllabus used for the most widely sought personal certifications.

A recent initiative, known as ITIL v3 or the ITIL Refresh project, is underway. ITIL v3 will improve the usefulness and applicability of ITIL "by addressing the changing needs of users as the technology base and business requirements continue to evolve" and by applying and improving its applicability to small organizations.[14]

ITIL is a framework that describes best practices, but it does not stipulate or constrain solutions. For example, several installations could all establish processes that are entirely consistent with ITIL but that do not interoperate with each other. Implementors are encouraged to adopt best practices that meet their business needs. Where, then, is the value? The value comes from two primary sources.

First, the best practices themselves have proven their value in many diverse IT environments. An organization that implements these practices builds on the cumulative experience that led to the current practices. It avoids the trial-and-error approach that typically occurs without the guidance drawn from ITIL.

Second, although different process implementations might not directly interoperate in an automated fashion, the shared concepts, including common terminology and approach, can lessen interoperability problems. For example, suppose two IT organizations are merging, and the first step is to consolidate to a single service desk, even though the back-end incident processing will remain separate for several months. If both incident management processes have similar concepts, definitions, and descriptions of an incident and how it is processed, then a straightforward conversion of either incident record could enable this integration. If different approaches are used, then such integration could be costly and difficult.

ITIL service support

Service support is the most common starting point for organizations adapting best practices based on the ITIL framework. Service support has five process areas (configuration management, change management, release management, incident management, and problem management) and one function (service desk):

1. *Configuration management*—This process is the foundation of service support and an integration point for other ITIL processes. A configuration management database (CMDB) is the central component of the configuration management process. A CMDB contains configuration records, which document the life cycle of a single configuration item (CI) or the relationship between CIs. A CI is any component that needs to be managed in order to deliver an IT service. Examples of CIs are hardware, software, buildings, people, and formal documentation, such as process documentation and service level agreements (SLAs). Commercial implementations of a CMDB often contain other information linked to CIs; for example, incident, problem, or change records.

   The configuration management process ensures that no configuration record (and hence, no corresponding CI) is added, modified, replaced, or removed without appropriate controlling

documentation. Verification and audit processes verify the physical existence of CIs and ensure that configuration records are correctly recorded in the CMDB.

The CMDB represents a logical database. In practice, the CMDB often consists of several data repositories, particularly if it includes all the types of data that an organization would like to manage with the ITIL processes.

2. *Change management*—This is the set of processes that manage and control changes to the actual CIs and relationships among them. Contrast the role of change management with configuration management, which controls documentation about CIs and relationships. Together the change-management and configuration-management processes enable IT organizations to plan and control changes and verify documentation about changes and the state of the IT environment. The change management process is initiated when a client or IT staff member submits a request for change (RFC). The RFC describes the desired outcome, usually in business-relevant terms, such as "deploy the new order-processing service" or "increase capacity to support 25 percent greater transaction load during the upcoming holiday season."

3. *Release management*—This process is often used to implement approved changes, particularly if the implementation is complex. Several factors influence the determination of the complexity of a change, such as the number of affected components and the level of coordination required among implementors. Some IT organizations use release management to implement all changes, whereas others implement simple changes without using release management.

4. *Incident management*—This is the set of processes that deal with service disruptions. In an ideal world, technology would never fail, services and technology would be correctly designed, and all deployments of new and updated services would be planned and executed flawlessly. In practice, of course, none of these ideals hold true. The environment is too complex, the changes too frequent, the cost and time pressures too great, and the knowledge of the interactions among the services, business processes, and IT infrastructure inadequate. Things go wrong, or at least appear to go wrong, from the perspective of a client. The entry point into incident management is the *service desk* function. IT staff use the service desk to record and monitor the progress of incidents and manage incident priority.

5. *Problem management*—This process is responsible for identifying the root cause of incidents and tracking the problem until it is fixed or a permanent alternative is identified. Some problems are recognized during the investigation of an incident. Other problems are recognized during analysis of many incidents or during analysis by other processes, such as the availability management process. For example, a problem might not be severe enough by itself to cause clients to report incidents, but the cumulative effect of such a recurring problem could negatively affect service levels. Problem resolution can take many forms, such as requiring a software redesign by development, installing a software patch, or replacing or reconfiguring a component to enhance its performance.

ITIL service delivery

Service delivery focuses on planning for and improving IT services, whereas service support focuses on day-to-day operations. Service delivery is composed of five process areas:

1. *Service level management*—This process is a common starting point. The IT organization negotiates SLAs with its customers (internal or external). These agreements specify service levels such as end-user response time, allowable planned and unplanned downtime, and cost. The SLA provides the direct link between the business units that contract for services and IT.

2. *Availability management*—This process is responsible for planning, monitoring, and improving the availability of business services. It seeks to strike the best balance between availability metrics, such as mean time to failure and recovery time and the cost to deliver a service level. Analysts might plan redundancy to increase availability to the end user, but this improved availability comes at a cost. Availability management is also responsible for maintainability—planning the processes to restore services or components back to normal operations after disruptions.

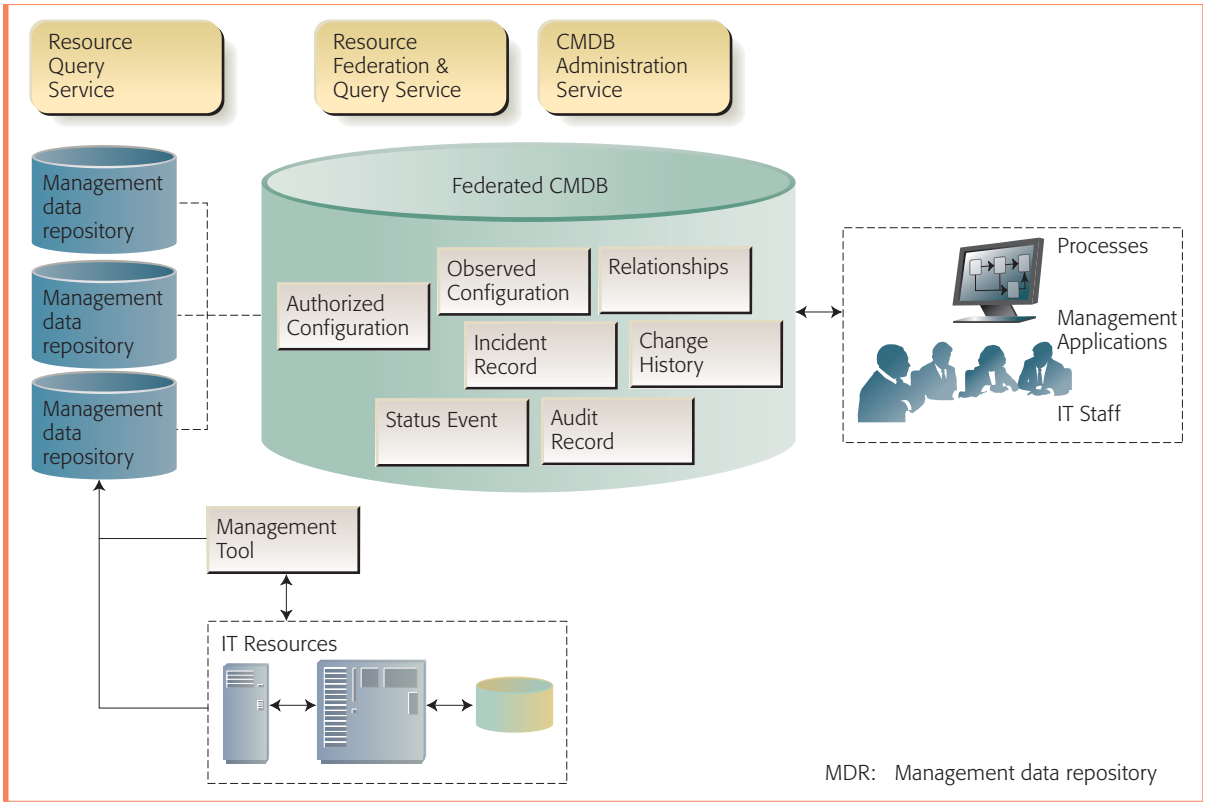

Figure 2
Federated CMDB architecture

3. *Capacity management*— This process is responsible for planning, monitoring, and improving the capacity to deliver business services. Capacity management seeks to strike the best balance between capacity and cost. Part of the capacity manager's job is to translate business-oriented requirements, such as the volume of cash register sales per hour, into IT metrics, such as the number of database transactions per hour that the projected volume of cash register sales will generate.

4. *Financial management for IT services*—This process is responsible for managing an IT service provider's budgeting, accounting, and charging requirements (capital costs and depreciation are part of another process called asset management that is not detailed in this paper).

5. *IT service continuity*—This process shares characteristics with availability management, but it has a different emphasis. Availability management focuses on specific services and components, often dealing in time frames of seconds or a few minutes. IT service continuity could deal with such time frames, but it is more likely to address massive recoveries that take hours or days. For example, if a flood causes a computing center to become inoperable, what is the plan for bringing an alternate site online, and what is the plan for providing at least partial services until it is online?

CMDB FEDERATION SPECIFICATIONS

IT management data (such as change, configuration, problem, incident, asset, and release data) is in every corner of the enterprise. Pulling together meaningful information from multiple, separately developed, distributed data sources is a difficult undertaking from the technical perspective, especially because of mismatches in interfaces and data models from vendor to vendor. The use of multiple sources for tools and services is so much a part of the landscape that the refresh to the widely used ITIL publications has identified support for multi-source environments as one of the significant

drivers. Many organizations are striving to base IT management on a CMDB that federates configuration and other data. This is consistent with the ITSM architecture, as shown in ***Figure 2***, which is built around shared data. IT processes and management tools use the data to coordinate with each other and as a basis for automated operations.

Several companies are working together to develop a specification that defines how to federate management data repositories (MDRs) into a virtual CMDB that spans all or part of the contents of each repository.[7] Typically, the federated CMDB normalizes data, reconciles resource names in situations in which different names refer to the same resource, and arbitrates among multiple sources that provide overlapping data. A federated CMDB is used to implement IT processes in an environment with multiple and often overlapping data sources and tools. It is often neither practical nor desirable to keep all management data in one data repository, although it may be practical and desirable to consolidate various subsets of the data.

A CMDB as defined by ITIL contains a record of the authorized configuration of the IT environment. The federated CMDB in this specification extends this base definition to federate any management information that an administrator configures, as long as the information complies with the patterns, schema, and interfaces of the specification. For example, as shown in Figure 2, the federated CMDB may include the observed configuration as well as the authorized configuration, the change history, incident records and audit records, status change events, and other related information (e.g., proposed or projected future states and process artifacts such as RFCs). In the specification, all of these data are called items.

The chief advantage of a federated approach is that a uniform view of the data can be created and maintained without requiring the replication of all data to a central data store. Further, management tools may continue to use the existing data stores, and the organization has more flexibility for deciding when and how it evolves to use data in the federated CMDB. This approach to creating a federated CMDB accommodates diverse MDRs, management tools, and IT processes.

There are two primary elements in the architecture, the federated CMDB and the MDR. The federated CMDB implements administration, resource federation, and resource query services. The MDR implements resource query services.

- *CMDB administration*—The CMDB and each MDR use the administration services to register their services, the schemas they support, and their capabilities. Clients may query the administration services to locate services that satisfy their requirements.
- *Resource federation*—Each MDR uses the resource federation services to register resources (both items and relationships) that it manages. The services reconcile the resource identities and optionally other resource data.
- *CMDB resource query*—Clients use the CMDB resource query services to access any data stored in or accessible through the CMDB and MDRs. This includes identification data and optionally resource data. Depending on the implementation, the queried data may be stored locally or it may be federated from MDRs.

The CMDB is not a proxy to interact with managed resources. For example, creating a relationship instance in a CMDB does not change the configuration of real resource relationships. Separate agents that are outside the scope of the CMDB monitor and change resources. Similarly, the mechanisms used by each MDR to acquire data are outside the scope of the specification, as are the mechanisms and formats used to store data. The federated CMDB specification is concerned only with the exchange of data; it does not dictate how to manage the data, although enabling the use of ITIL processes is also an objective.

SERVICE MODELING LANGUAGE

An enterprise typically has many different sources of authoritative data that form the nucleus of its CMDB. Exchanging information between two data sources can be problematic if the formats are not compatible. Traditional technological approaches such as database keys provide one approach to uniquely identify and correlate the data, but they often result in undesirably tight coupling among the interacting components. Moreover, they may depend on each component having implicit knowledge about assumptions made by the other component because there may be no way to explicitly assert the constraints that apply to the data.

Different implementations from different vendors can make different scope and depth decisions that affect the type of information contained in a particular resource. The design of the software and its subsequent implementation reflect these decisions. For example, consider an asset data store contrasted with an incident data store. In this example, the asset management system places emphasis on coarse-grained machine attributes and focuses on physical location and value, whereas the incident management system places more emphasis on finer-grained machine attributes and logical location in the IT infrastructure. These two management systems are similar, but have different goals. The two systems both require information about many of the same resources, but each requires information that is different in scope and granularity. These differences can result in difficulties when the data between the two management systems are correlated.

The Service Modeling Language (SML) and its associated SML Interchange Format (SML-IF) are intended to address these requirements. They are expected to play an important role in the interchange of data among management systems, such as the data repositories that make up a federated CMDB.

Overview of SML

An SML model is a collection of XML documents used to describe a set of IT resources and their interrelations. In every SML model, there is a distinguished subset of the documents that comprise it, called the *definition* documents. There are two categories of definition documents: *schema* documents and *constraint* documents. Schema documents in a model are XML documents that conform to the SML profiling and extensions to XML Schema 1.0.[15] Constraint documents in a model are XML documents that conform to the SML profiling and extension of *Schematron*.[16] The definition documents provide much of the information that a model validator requires to determine if the model as a whole is valid. Because model validity in SML depends in part on dependencies among the documents that make it up, it is certainly possible that adding a document which is valid in one sense to an existing valid model could render the resulting model invalid.

The other documents in the model, called *instance* documents, describe the individual resources that the model portrays. Instance documents conform to the schema and constraints defined by the definition documents. Broadly speaking, an SML model is a graph of nodes connected to one another by arcs. The instance documents of a model form its nodes; explicit interdocument references form its arcs.

Schemas in SML

In the world of XML schema validation, SML schemas are like any other schema: A document that can be validated with respect to a set of schemas is valid with respect to them or it is not. SML modestly extends XML Schema 1.0 to provide schema authors with the means to impose additional constraints on the form and content of documents that conform to schemas they create. These constraints, called interdocument constraints, take the form of Schematron rules embedded in XML Schema. They provide authors with the ability to specify, for example, whether attributes of a particular element can cooccur. This ability to specify inter- and intradocument constraints (which is not available in XML) is expected to facilitate more robust implementations of a federated CMDB.

SML also provides schema authors with the ability to specify that instance documents in a model can have dependencies on other documents in the model. Unmet dependencies of this sort can cause a collection composed entirely of valid documents to be an invalid model. For example, to model a particular kind of Ethernet adapter installed in a server, the schema that models that type of Ethernet adapter might require that each of its document instances be related to an instance of a document that models a server. Any model that contains an Ethernet adapter document of this kind that does not have a relationship to a document modeling a server is invalid.

Constraints in SML

Constraint documents form a sort of "physics" for SML models. In addition to the constraints that the schemas of a model place on documents, constraint documents may be bound to particular instance documents. Constraint documents contain Schematron rule patterns that constrain the form of the documents in the model of which they are a part. The scope of a constraint document is the entire model, a capability that can constrain not only individual documents but also the references between documents. Constraint documents are useful

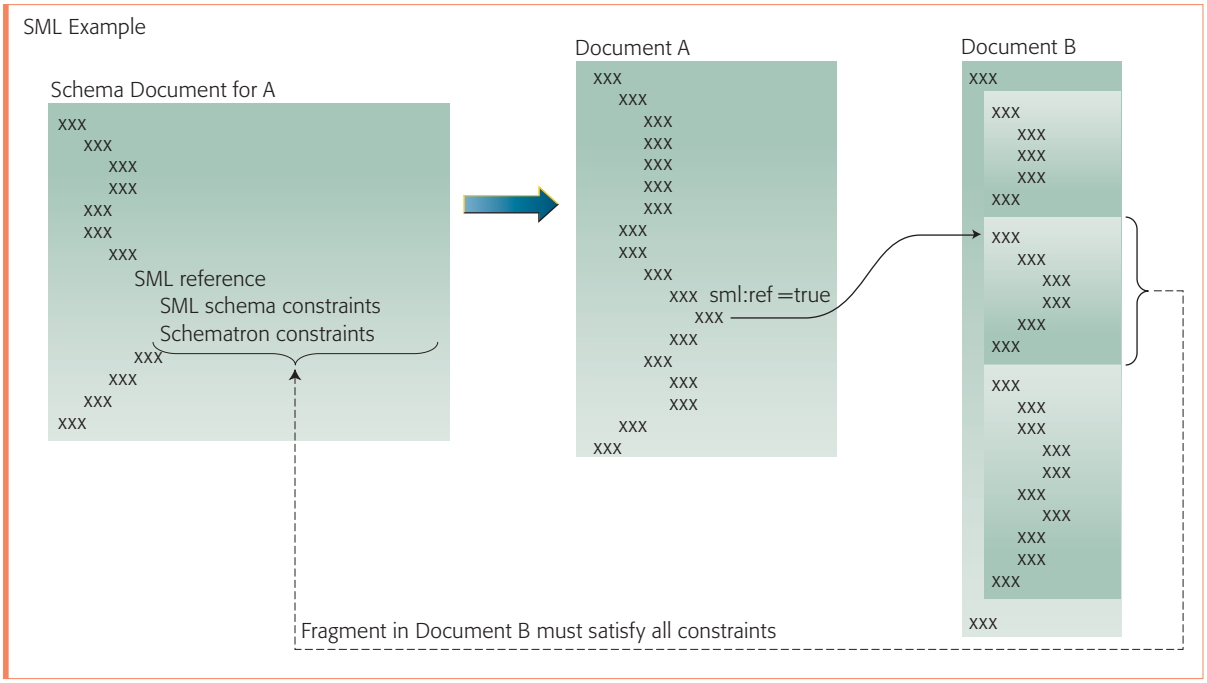

Figure 3
Interdocument reference in SML

for a variety of purposes, including setting policies for entire models.

References in SML

XML documents introduce boundaries across content that must be treated as a unit. XML Schema does not have any support for interdocument references. SML extends XML Schema to support interdocument references, as shown in ***Figure 3***, and adds a set of constraints for interdocument references. The example fragments in Figure 3 depict the schema definition for document A that contains a reference (indicated by the XML attribute `sml:ref=true`) to document definition B. For the model to be valid, the target of the reference to B must satisfy the constraints in the definition of A. Support for interdocument references includes the following:

- A new data type that represents references to elements in other documents
- Multiple addressing schemes for representing references
- Constraints on the type of a referenced element
- The ability to extend the XML `key`, `unique`, and `keyref` (key reference) constraints throughout a set of documents

SML, overall, helps unify the modeling of resources because a standard representation for resource information facilitates the exchange and reconciliation of information among management tools. SML and SML-IF are good matches for the requirements of the federated CMDB. In addition, validating constraints in exchanged documents leads to earlier detection of inconsistency, which, in turn, makes IT management more robust and more effective.

SOLUTION DEPLOYMENT DESCRIPTOR

Developers create solutions based on a set of assumptions about the operating environment and about how the solution will be configured for that operating environment. The task falls to IT administrators and operators to understand the documentation that accompanies the solution and the installation and execution requirements. Administrators must assess the compatibility of the solution and the environment and correctly configure the solution. This is an error-prone task for a variety of reasons. The documentation might not be sufficient or entirely accurate; the documentation could be ambiguous, and its interpretation by the IT staff might be different from what was intended; the environment may differ in a material way from the environment expected by the developers. The ITSM

process would be more robust if the assumptions and requirements about software, over its life cycle, were encoded in a formal manner by the developers, packagers, solution integrators, and others who participate in the software deployment and management processes.

The SDD is an emerging standard for representing metadata about installable software packages and their configuration, dependency, and life-cycle information.[10] Traditionally, software developers document installation requirements in an installation guide and expect users to understand and adhere to them. Because the software developers already know what the dependencies are, it is better if the software checks the system on behalf of the users. The ability to define and verify dependencies and other requirements is one of the main advantages of a solution installation technology based on standard software-deployment information; SDD provides a mechanism to describe the dependencies and other requirements, along with other important information about software deployment. Examples of checking for software requirements include verifying the operating-system type, ensuring that sufficient memory and disk space are present (during and after installation), and validating and satisfying any dependencies on other software packages.

In addition to deployment information, SDD also specifies software-packaging information. Today, many package formats exist, such as the RPM (originally Red Hat Package Manager) format used in many Linux** distributions and AIX* file sets that contain similar information, along with many others. However, packages typically are embedded in the artifacts in a proprietary format and are often consumed only by applications that reside in each target hosting environment. SDD externalizes metadata about packages, including relationships within and among packages, in a standard canonical format.

To address the challenges of multiple proprietary ways to express packaging and deployment information, OASIS has chartered the SDD Technical Committee to develop a specification and schemas to describe the characteristics of installable units of software. Processes for the deployment, configuration, and maintenance of software can take advantage of these characteristics. The Installable Unit Deployment Descriptor Version 2[17] is the basis for the SDD specification. The committee is also collaborating with the Open Grid Forum work groups for Application Content Services (ACS), Job Submission Description Language (JSDL), and Configuration Description, Deployment, and Life-Cycle Management (CDDLM). The ACS work group formally approved the use of the SDD Package Descriptor in September 2006.

Reference 18 describes SDD in more detail.

WEB SERVICES MANAGEMENT PROTOCOLS

Current IT environments are heterogeneous; many existing components do not use standardized interfaces and thus do not communicate well with each other. The integration of IT management components is enabled by a service-oriented approach, based on Web services.

Web services distributed management

Web services technology addresses the general problem of integrating applications, especially those built with a heterogeneous set of implementation technologies and platforms. Applying Web services technology in the systems management domain yields a common messaging protocol between a manageable resource and a manageability consumer.

WSDM is a set of specifications for management by using Web services and management of Web services. These specifications describe the use of Web services for managing resources and the use of Web services to manage other Web services. WSDM makes use of the Web Services Resource Framework resource-access specifications. Treating manageable resources as Web service resources provides a consistent set of interfaces needed to access manageable resource information. The WSDM standards specify this common messaging protocol for managed resources and their consumers. In contrast, WSDM does not prescribe a data or information model for the properties, operations, relationships, and events of managed resources. WSDM provides Web services interfaces for resources described by any resource generic model, such as Simple Network Management Protocol (SNMP)[19] and Common Information Model (CIM),[20] or proprietary models. Hence, a legacy application wrapped with a WSDM interface provides Web services access to the model already used by the application.

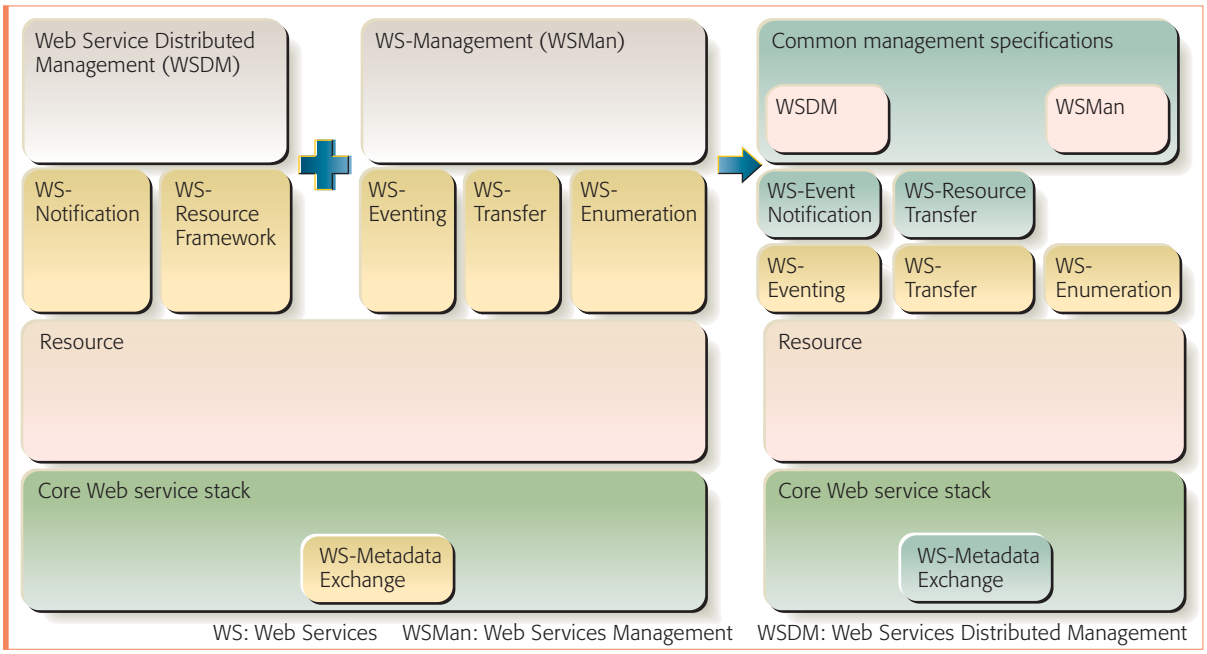

Figure 4
Harmonizing Web Services Management specifications

The focus of the WSDM architecture is the manageability interface that enables the manageable resource to be represented as a Web service. An endpoint reference (EPR), as defined in the WS-Addressing standard,[21] represents the means to access a particular manageable resource at a manageability endpoint. The implementation behind these manageability endpoints must be capable of retrieving and manipulating the information related to a manageable resource. The manageability consumer (management tool) directs messages to the location represented by the EPR. There is also a model for manageable resources to send direct notifications to the consumer, provided the consumer has subscribed to receive notifications. The WSDM capabilities for manageable resources are mapped in the standard to be accessible by using the WS-ResourceProperties specification for getting and setting properties and the WS-Notification family of specifications for carrying and distributing events to the manageability consumer.

Harmonization of Web services management specifications

WS-Management is a specification that is comparable to WSDM. As mentioned earlier, both WSDM and WS-Management refer to and incorporate many other Web services specifications that provide capabilities for resources, events, and management. IBM, Hewlett-Packard Development Company, Intel Corporation, and Microsoft Corporation have recognized the need to harmonize the Web Services Management specifications and are writing a common set of specifications that address the requirements currently satisfied by the specifications from the OASIS WSDM and DMTF WS-Management committees as described in a published roadmap.[22] This section presents an overview of the material and the status of the specifications that define this common set of capabilities for managing system resources using Web services. WSDM/WS-Man Reconciliation[23] describes a detailed mapping from the WS-Resource Framework specifications to the new reconciled specifications described in the roadmap.

Figure 4 illustrates the existing stack of specifications and the expected resulting stack of specifications. Both of the existing stacks provide a means to create, retrieve, update, and delete the XML representation of a resource by using Web services protocols. Both stacks also provide a means of subscribing to changes in a resource representation as well as a means to carry those events. The harmonization also modified WS-Metadata Exchange,[24] which is not a management-specific specification, to eliminate some technical overlap in the messages defined in resource specifications.

To ensure that a management specification covers all of the original use cases from both efforts, there may be specific functionality that remains in separate specifications (represented by the small pink boxes in the Common Management Specifications box in Figure 4). Ideally, these specifications would not be required. The function of the new specifications is divided into three major areas: (1) information management, (2) events and notification, and (3) management specifications and profiles.

Both WS-Transfer and WS-ResourceProperties provide methods to access and manipulate the XML representation of a resource by using Web Services protocols. By examining various use cases that guided the development of WS-ResourceProperties and WS-Management, the reconciliation effort created one unified specification, *WS-ResourceTransfer*. WS-ResourceTransfer extends an updated version of WS-Transfer with a means to access and manipulate both complete and partial representations (fragments) of a resource. In an operation, a fragment is identified with an expression that denotes the subset of interest. The specification allows implementations to specify a *dialect* (or language) to use when formulating the expression. The set of dialects supported by a resource can be discovered through the metadata of the resource. In addition to the dialects supported by a particular resource, WS-ResourceTransfer also defines metadata relating to the life cycle of the resource.

WS-Eventing is an existing specification that enables simple, interoperable publish/subscribe systems. A new specification produced as part of the harmonization effort, *WS-EventNotification*, builds on the WS-Eventing specification with capabilities from the WS-Notification specifications. The planned features that WS-EventNotification adds are a means to specify subscription, policy, richer filter languages, pausing of subscriptions, and treatment of subscriptions as manageable resources.

Where differences exist in current management specifications, many result from the use of different underlying resource specifications. New management specifications and profiles that are in development, based on the new WS-ResourceTransfer and WS-EventNotification specifications, should allow for the use cases from both the WSDM and WS-Man families of specifications to be supported by the new harmonized specifications.

DISCUSSION

The future of standards for ITSM is not entirely predictable. Standards generally evolve in response to market demands, and determining the value for new or enhanced standards takes time. The oldest standards related to ITSM are those that define resource interfaces, such as SNMP[19] and CIM.[20] These are being augmented or replaced by Web services standards, though their fundamental nature has not changed dramatically. It is likely that we will continue to see the evolution toward standards that support a service-oriented architecture (SOA), a preferred infrastructure for connecting components together. SOA leverages Web services to provide a vendor-, platform-, network- and protocol-neutral framework. This approach loosely couples existing components in order to integrate tools without massive revision. It also provides each IT organization the flexibility to select tools that best suit its business goals.

The future is more uncertain for areas in which standards are less mature, in particular IT processes and CMDB federation. ITIL is a widely used framework for defining processes, but it avoids defining specific process interfaces. Although in some sense this seems an unsatisfactory limitation, it is beneficial in that it allows diverse IT organizations to use ITIL successfully. Potentially, standards could develop to integrate IT process definitions at modeling or execution time. For example, an IT process workflow defined with one modeling tool could be integrated with an IT process workflow defined with a different tool. Moreover, an IT process workflow running in one execution environment could be integrated with an IT process workflow running in a different execution environment. The Web Services Business Process Execution Language (WS-BPEL)[25] standard provides underlying technology for this type of integration, and it could become a dominant workflow technology in the future.

Neither ITIL nor BPEL provides definitions of specific interoperable process steps, such as the verification and audit responsibility of the ITIL configuration management process. It might be worthwhile to define standards at this level to enable interoperability. For example, the verification and audit process from one vendor could be paired with the configuration status accounting process from a different vendor to achieve the same result that is achieved by using processes from a

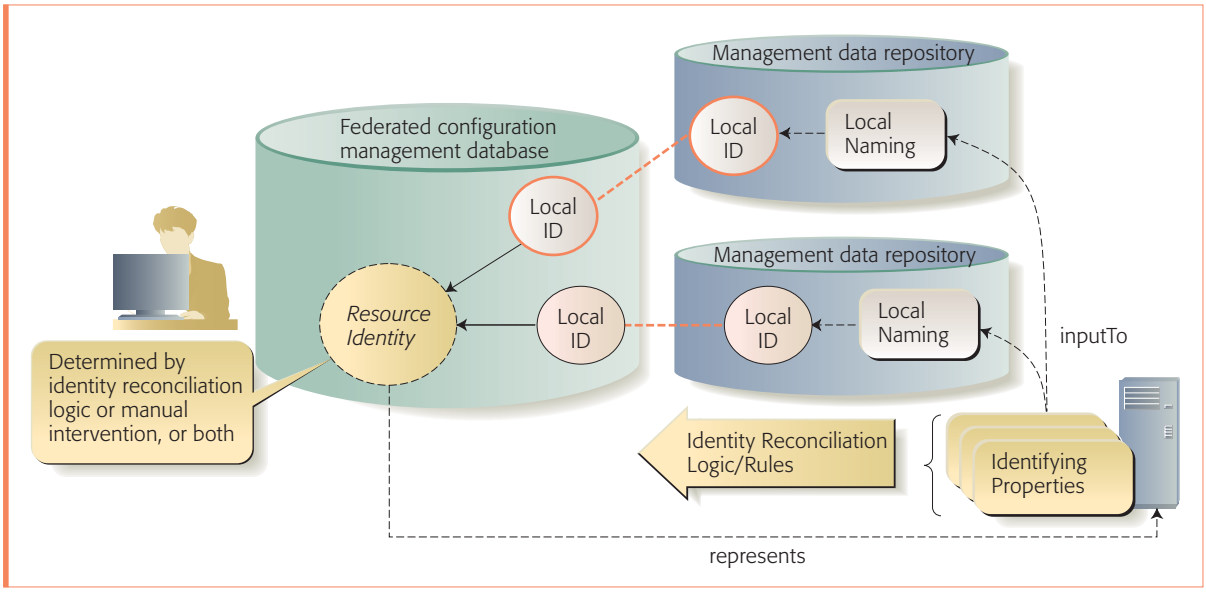

Figure 5
Identity reconciliation in CMDB data

single vendor. The current state of the industry is a long way from achieving this level of interoperability, or even determining if this interoperability has sufficient value to make the standardization effort worthwhile.

The CMDB federation specification is the newest and least mature of the standards described in this paper. The initial focus has been on defining common interfaces to register and query data in repositories with disparate data models. Additions to increase data consistency across repositories would further increase interoperability. One useful addition would be to define common data models by using SML. Such models would define widely used properties, much like CIM does, while allowing for extensibility. Consistent models make it easier for clients, such as management tools and processes, to interact with different implementations. However, converting implementations to use different data models takes time. An intermediate step is to define only the information needed to reconcile resource identities.

Reconciling resource identities is perhaps the most important responsibility of a federated CMDB. Each MDR knows at least one and often multiple properties that serve to identify each resource. By comparing the different properties known by each MDR, the CMDB often can determine when data of different MDRs relate to the same resource. For example, a computer system may be known by any of the following: its media-access-control (MAC) address; the combination of its machine type, model, and serial number; a globally unique ID permanently assigned to a system board; or an asset number assigned by the asset management process. As shown in ***Figure 5***, a federated CMDB can analyze the identifying properties that are presented when an MDR registers a resource. In many cases the analysis is performed successfully by machine; in others, manual intervention may be required. The envisioned standard would not dictate how these mechanisms work; it would describe how the identifying properties and local identifier maintained by the MDR should be presented to the resource federation service.

Other possible extensions to federated CMDBs are developing a common model for authorizing access to data, adding publish/subscribe mechanisms, and extending the data model to define common versioning constructs.

CONCLUSION

Businesses are demanding more from their IT organizations. They require better and more disciplined provisioning of IT services to ensure smooth

operation, predictable budgets, and satisfied customers. They require improved communications between IT and lines of business. They require controls on IT expenditures. They require IT to respond quickly with appropriate services to support new business opportunities.

Standards-based solutions ensure the interoperability needed to connect internal and external applications and data. Open standards also provide the ability to quickly integrate new hardware and software into the existing infrastructure and to adjust the infrastructure to changing business needs. The use of standards reduces risk. Standards-based solutions allow customers to find optimal solutions for their environment, with the assurance that the heterogeneous mix is interoperable with new and existing IT assets. IT organizations know standards-based solutions allow them to improve their business agility and exploit technology cost reductions. IBM has taken a leadership role in developing open standards by incorporating standards in its products and by supporting open-source projects.

We have discussed several standards that are expected to strongly influence the development and adoption of management solutions in service-enabled environments. Using the standards described here will likely lead to a smoother adoption of emerging ITSM solutions.

CITED REFERENCES AND NOTES

1. *Information Technology Standards*, National Institute of Standards and Technology, United States Commerce Department, http://www.nist.gov/public_affairs/standards.htm.
2. D. Lindquist, H. Madduri, C. J. Paul, and B. Rajaraman, "IBM Service Management Architecture," *IBM Systems Journal* **46**, No. 3, 423–440 (this issue, 2007).
3. *ITIL Planning to Implement ITSM*, Office of Government Commerce, United Kingdom (2002).
4. *Foundations of IT Service Management Based on ITIL*, J. Van Bon, M. Pieper, and A. van der Verrn, Editors, van Haren Publishing (2006).
5. *ITIL Service Support*, Office of Government Commerce (2000); *ITIL Service Delivery*, Office of Government Commerce (2001). These and other similar titles are published by the Office of Government Commerce, United Kingdom.
6. *ISO/IEC 20000-1:2005, Information Technology—Service Management—Part 1: Specification*, International Organization for Standardization (2005).
7. *The Federated CMDB Vision*, A joint white paper from BMC Software, Inc., Computer Associates International, Inc., FUJITSU, Hewlett Packard Development Company, IBM Corporation and Microsoft Corporation (January 2007, available from the author).
8. *Service Modeling Language, Draft Specification, Version 1.0* (March 21, 2007), World Wide Web Consortium (W3C), http://www.w3.org/Submission/sml/.
9. *SML Interchange Format, Draft Specification, Version 1.0* (March 21, 2007), World Wide Web Consortium (W3C), http://www.w3.org/Submission/sml-if/.
10. OASIS Solution Deployment Descriptor (SDD) Technical Committee, Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sdd.
11. The following documents were published by the World Wide Web Consortium (W3C): *Web Services Eventing (WS-Eventing)*, W3C Member Submission from BEA Systems Inc., Computer Associates International Inc., International Business Machines Corporation, Microsoft Corporation, Inc., and TIBCO Software Inc. (March 15, 2006); *Web Service Transfer (WS-Transfer)*, W3C Member Submission from BEA Systems, Inc., Computer Associates, Microsoft Corporation, Sonic Software, and Systinet (September 27, 2006); *Web Services Enumeration (WS-Enumeration)*, W3C Member Submission from BEA Systems Inc., Computer Associates International, Inc., Microsoft Corporation, Inc., Sonic Software, and Systinet (March 15, 2006); *Web Services Resource Transfer (WS-RT), Version 1.0*, from Hewlett-Packard Development Company (HP), Intel Corporation, IBM Corporation, and Microsoft Corporation (August 2006).
12. *Web Services for Management (WS-Management)*, Distributed Management Task Force, Document Number DSP0226, http://www.dmtf.org.
13. The following documents were published by the Organization for the Advancement of Structured Information Standards (OASIS): *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1*, Committee Draft (January 11, 2005), Document Identifier cd-wsdm-muws-part1-1.0; *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 2*, Committee Draft (December 9, 2004), Document Identifier cd-wsdm-muws-part2-1.0; *Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0*, OASIS-Standard (March 9, 2005); *Web Services Base Notification 1.3 (WS-Base-Notification) Committee Specification* (July 31, 2006), Document Identifier wsn-ws_base_notification-1.3-spec-cs-01; *Web Services Brokered Notification 1.3 (WS-Brokered Notification) Committee Specification* (July 31, 2006), Document Identifier wsn-ws-brokered-notification-1.3-spec-cs-01; *Web Services Topics 1.2 (WS-Topics)*, OASIS Working Draft 01 (July 22, 2004); *Web Services Base Faults 1.2 (WS-BaseFaults)*, OASIS Standard (April 1, 2006), Document Identifier wsrf-ws_base_faults-1.2-spec-os; *Web Services Resource Lifetime 1.2 (WS-ResourceLifetime)*, OASIS Standard (April 1, 2006), Document Identifier wsrf-ws_resource_lifetime-1.2-spec-os; *Web Services Resource 1.2 (WS-Resource)*, OASIS

Standard (April 1, 2006), Document Identifier: wsrf-ws_resource-1.2-spec-os; *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, OASIS Standard (April 1, 2006), Document Identifier wsrf-ws_resource_properties_1.2-spec-os; *Web Services Topics 1.2 (WS-Topics)*, Working Draft 01 (July 22, 2004), OASIS.

14. *ITIL Refresh Statement* (December 13, 2005), Office of Government Commerce, United Kingdom, http://www.itil.co.uk/refresh.htm.
15. *XML Schema Part I: Structures (Second Edition)*, World Wide Web Consortium (W3C), Recommendation (October 28, 2004), http://www.w3.org/TR/xmlschema-1/.
16. *Information Technology—Document Schema Definition Languages (DSDL)—Part 3: Rule-Based Validation—Schematron*, ISO/IEC 19757-3, American National Standards Institute, http://webstore.ansi.org/ansidocstore/default.asp.
17. *Installable Unit Deployment Descriptor Specification Version 1.0*, W3C Member Submission, from InstallShield Software Corporation, International Business Machines, Inc., Novell, Inc., and Zero G Software, Inc. World Wide Web Consortium (W3C) Recommendation (July 12, 2004), http://www.w3.org/Submission/InstallableUnit-DD/.
18. P. Brittenham, R. R. Cutlip, C. Draper, B. A. Miller, S. Choudhary, and M. Perazolo, "IT Service Management Architecture and Autonomic Computing," *IBM Systems Journal* **46**, No. 3, 565–581 (this issue, 2007).
19. *A Simple Network Management Protocol (SNMP), Request for Comments 1157*, Internet Engineering Task Force (May 1990), http://www.ietf.org/rfc/rfc1157.txt.
20. Common Information Model (CIM) Standards, Distributed Management Task Force, http://www.dmtf.org/standards/cim.
21. *Web Services Addressing (WS-Addressing)*, W3C Member Submission from BEA Systems Inc., International Business Machines Corporation, Microsoft Corporation, Inc., SAP AG, and Sun Microsystems, Inc. World Wide Web Consortium (W3C) (August 10, 2004), http://www.w3.org/Submission/ws-addressing.
22. *Toward Converging Web Service Standards for Resources, Events, and Management, Version 1.0*, a joint white paper from Hewlett Packard Development Company, IBM Corporation, Intel Corporation, and Microsoft Corporation (March 15, 2006), http://devresource.hp.com/drc/specifications/wsm/.
23. *WSDM/WS-Man Reconciliation, An Overview and Migration Guide, Version 2.0*, IBM (May 2007), http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-wsdmmgmt/wsdmmgmt_v2.pdf.
24. *Web Services Metadata Exchange (WS-MetadataExchange) Version 1.1*, BEA Systems Inc., Computer Associates International, Inc., IBM Corporation, Microsoft Corporation, SAP AG, Sun Microsystems, Inc., and webMethods (August 2006), http://schemas.xmlsoap.org/ws/2004/09/mex.
25. *Web Services Business Process Execution Language Version 2.0*, Committee Draft, World Wide Web Consortium (W3C) (May 17, 2006), www.oasis-open.org/committees/wsbpel.

Accepted for publication April 12, 2007.
Published online June 27, 2007.

Mark W. Johnson
IBM Software Group, Tivoli, 11501 Burnet Road, Austin, TX 78729 (mwj@us.ibm.com). Mr. Johnson is a senior software engineer in the Tivoli Technical Strategy and Architecture group. He received a B.S. degree in electrical engineering from Cornell University. His major areas of interest are application and service management in which he has published papers and holds several patents. He was a co-creator of the Application Response Measurement standard. He has chaired working groups in The Open Group and the Distributed Management Task Force and is currently an editor of the CMDB Federation workgroup.

Andrew Hately
IBM Software Group, Tivoli, 11501 Burnet Road, Austin, TX 78729 (hately@us.ibm.com). Mr. Hately is a senior software engineer in the Software Standards group. He has led software development and reference implementation efforts relating to Web Services, registries, and repositories. His primary interest is in applying service discovery, network discovery, and service management standards to business problems. He was the co-editor of the UDDI Version 3 specification and has represented IBM in several specification efforts at OASIS, The Open Group, and other consortia. He has spoken at several industry conferences on the subject of standards, Web Services and service-oriented architecture. He holds several patents related to software design and has a B.A.Sc. in environmental engineering from the University of Waterloo.

Brent A. Miller
IBM Software Group, 4205 South Miami Boulevard, Research Triangle Park, North Carolina 27709 (bamiller@us.ibm.com). Mr. Miller, a Senior Technical Staff Member, is the lead architect in autonomic computing, addressing both technology and standards efforts associated with self-managing autonomic systems and ITSM. He has a B.S. degree in computer and information science from the Ohio State University. He is coauthor of *Bluetooth Revealed*, published by Prentice-Hall in 2000.

Robert Orr
IBM Software Group, Tivoli, 3901 South Miami Blvd, Durham NC 27709 (orrr@us.ibm.com). Mr. Orr is the manager of advanced technology and strategy for the Tivoli Chief Technology Office. His technical interests include computer architecture, system monitoring, and performance. He holds several patents in these areas. He is currently the project manager of the CMDB Federation working group. He has a B.S. degree in computer science from the University of Maryland. ■