ITIL And The Evolving CMDB

Hank Marquis

Why is the Configuration Management Database so often misunderstood?

Where there is the heat of hype, there is fire and, without a doubt, the Configuration Management Database (CMDB) is hot. Just look at IT service management vendor advertising and promotion: It seems as if the vendors spend more on the term "CMDB" than on any other element of the IT Infrastructure Library (ITIL). And perhaps with good reason, as database and related resource management are aspects of ITIL that rank among the top issues facing business and technology managers today.

Apart from the hype, however, there is a real need for CMDBs, and for ITIL in general. ITIL is the world's most appealing IT management framework precisely because, by following its process and operational guidance correctly, one can reduce costs, improve quality, align IT with business and comply with ever increasing legislation and regulation—achieving many of the goals of every business and IT manager.

Configuration Management is one of the most important ITIL processes, and the CMDB is critical to success with virtually all the other ITIL processes. This means you simply cannot succeed at ITIL without a true CMDB.

Despite its importance, the CMDB remains misunderstood. Few vendors currently offer a real CMDB, and still fewer customers operate one. The reasons lie with both customers and vendors, and within the technical limitations of today's database management technologies.

Practitioners who see the term "configuration management database," tend to start thinking of a traditional database, and try to build their own CMDB. Vendors who initially decide to jump onto the ITIL bandwagon tend to try marketing their existing IT asset management (ITAM) products as if they were a CMDB solution.

But a CMDB is not an ITAM or a configuration database, and it is nearly impossible to build your own CMDB. Finally, there are real technological differences between these customer and vendor efforts and a true CMDB as described by the ITIL.

All of this is not to say that vendors aren't working to deliver real CMDB solutions. They are—and these new CMDB solutions promise to combine the latest in process theory, database technology, open systems architectures and even interoperability between competitors.

Based on my experience over the last 6 years working with dozens of clients using a variety of IT Service Management (ITSM) software products from CA, IBM, HP and others, this article describes what it takes to be a true CMDB, as described by ITIL, and how vendors are moving to produce real solutions.

Will The Real CMDB Please Stand Up?

A CMDB is much more than a database storing information about equipment configurations. A true CMDB also stores the dependencies and relationships among these assets, which are called configuration items (CIs) by the ITIL. And CIs are not just hardware and software assets, but also the documentation, processes and people that compose, support and consume IT services.

The chief point of confusion here has been the lack of understanding that a true CMDB is first and foremost a meta-database or "database of databases." It provides the context for mapping the IT infrastructure content descriptions (e.g., CIs) that are stored in other databases.

Put another way, a true CMDB is an n-dimensional "cube" of current and historical relationships between CIs. CMDBs must federate multiple definitive data sources, and manage the issues of CI reconciliation, synchronization, mapping and visualization.

By comparison, most of the so-called CMDBs in use today are actually Excel spreadsheets, homegrown relational databases, diagrams in Visio, and the like. The ideal ITIL CMDB is definitive—that is, after a lookup you don't need to confirm or verify the information. The very point of a CMDB is to be the authoritative source of CI information. Its definitive nature drives its benefits:

- Better assessments of the impact of proposed changes and fewer failed changes.

Hank Marquis is a managing partner and the chief technology officer at itSM Solutions LLC, a firm specializing in IT Service Management coaching and education. Contact him at hank.marquis @itsmsolutions.com

- Increased IT efficiency, more effective planning and decision making
- Fast, accurate answers to day-to-day questions.
- Reliable resource for technicians, service desks and change management personnel.

All IT activities require definitive data, but while IT organizations may find the concept of the CMDB appealing, they know that mapping CI relationships and dependencies is a manually intensive process. Pressed, they will also acknowledge that a CMDB which is not maintained will not be definitive, and a non-definitive CMDB is more a liability than an asset.

One study by ITSM software vendor Managed Objects found that only 33 percent of respondents reported using software to map relationships between elements in the IT infrastructure. The same survey found that just 21 percent claim to update their CMDB daily or in real time.

There are signs of hope, however. As more companies decide to base their operations on ITIL, they will become more educated about Configuration Management and more astute about their needs—perhaps painfully so, in some cases. These CMDB growing pains will, in turn, drive demand for the newer database technologies that vendors are developing to catch up to the ITIL-based requirements for a CMDB.

CMBD growing pains will help drive demand for newer database technologies

Vendors of all ilks are forming various consortia and marketing new products. Top vendors like CA, HP and IBM all claim to offer robust CMDB solutions. Other former network management vendors, such as Mercury Interactive (now owned by HP) and Managed Objects, as well as relatively recent entries like TouchPaper and TeamQuest, all now describe how they empower or deliver ITIL Configuration Management and CMDB functionality. One late entrant, Microsoft, is even trying to trying to patent the CMDB.

How CMDB Enables Dimensional Modeling

As mentioned above, a CMDB is not a database in the traditional sense, but a system for managing a meta-database, or a "database of databases." Gartner describes four key capabilities that make a CMDB different from other ITAM, asset inventories, device configuration files or relational databases: federation, reconciliation, synchronization, and mapping/visualization.

Taken together, these four capabilities enable creation of a dimensional database, something that few vendors deliver or even talk about. A dimensional database steps beyond the rows and columns of relational databases and allows many more types of data to be related to one another in additional dimensions (or planes).

Relational databases are optimized for on-line transaction processing (OLTP), but are weak with regard to *ad hoc* or dynamic queries. A dimensional database is optimized for on-line analytical processing (OLAP). A dimensional database enables dimensional modeling—the ability to quickly deliver responses to OLAP questions like "At what time of year do most users log on?" or "What locations have the fewest customer inquiries during the summer?"

This OLAP functionality is important in fulfilling the ITIL's absolute requirements for not only an audit trail, but also a baseline. An ITIL baseline is a snapshot of a CMDB configuration map or view at a specific moment in time. The baseline describes how a system is (and was previously) configured; it's also used to visualize what changed between subsequent versions of a configuration. As far as OLAP is concerned, a configuration at some moment in time is just the answer to a particular query.

Let's see how each of the four key CMDB capabilities contributes to federated dimensional modeling:

- **Federation—**Once it has been determined which data are definitive, that data must live in a particular, definitive database, the CMDB. Federation enables the communication between definitive and non-definitive databases. Helge Scheil, VP of development for Computer Associates, provided a financial record-keeping example. "You don't ever edit the General Ledger if you mis-pay a vendor," he said. "Instead, you make the change in the accounts payable system, which in turn updates the General Ledger."

In this example, the General Ledger (GL) is the CMDB, and the accounts payable (AP) and accounts receivable (AR) systems, among others, are federated to it. The point is that the GL is a view into the relationships between the AP and AR systems. Changes made in AP and AR get displayed in the GL.

Federation is not as simple as auto-discovering CIs or their relationships; it also must include an understanding of which data store is definitive, as well as which data reside in the CMDB store and which reside in the original data store. The CMDB must provide the means to store some data attributes locally and use this as the context to "glue" the federated data attributes into a cohesive configuration model or dimension.

A practical software product design trade-off here is one of speed versus accuracy. Many vendors cache local copies of data—possibly making it non-definitive. Other vendors don't cache, but rather link to the definitive data source—possibly slowing system response time.

As another example, consider a CMDB that ties together assets, people and projects. For their own parts, the asset management system, the human resources directory service application and the project management system have needs for updating and reporting that are confined to these original systems. However the relationships among assets, projects and users is a composite which does not exist in any of the federated data sources to which the CMDB connects; thus, the CMDB has to store keys and unique data that ties the federated data sources together. If the CMDB, as the system of record, is to remain definitive, it must “connect the dots” between federated systems.

- **Reconciliation—**Most IT organizations have lots of data, but they don’t know where it is, and they usually have the same data stored in more than one place. For example, one view of a router as seen by Cisco Systems’ CiscoWorks LAN Management Solution might be its name, another view from an Network Management Systems’ SNMP auto-discovery might be its IP address. Which is definitive? Which is the correct “name” for this CI?

Reconciliation is the process that ensures there are no duplicates and maintains the correct nomenclature for any given CI. Declaration of ownership and definitive residence of CIs and attributes also are critical to reconciliation.

In larger IT shops, reconciliation can’t be done solely in the heads of lead technicians

In some cases, particularly for the smallest shops, the reconciliation functions of a CMDB are performed in the brain and experience of the lead technician, who acts as the glue to pull all the disparate data sources together. In larger organizations, of course, this can’t be done in one person’s head, and the need for a sophisticated reconciliation engine increases as the number of federated data sources increases.

- **Synchronization—**To maintain synchronization with its federated databases, the CMDB system must discover changes made to the content or context of the infrastructure, then it must distinguish between approved and unapproved changes. An approved change is defined as one which is evaluated and pre-established, and for which there exists a Request for Change (RFC) in the CMDB.

When the CMDB system detects a CI change in a federated data source, it compares the change to its list of approved changes (e.g., RFCs) and synchronizes itself, if the change is approved; or it raises an alert if the change is not approved. Synchronization allows users in various ITIL-defined roles—such as Change Manager, Change Advisory Board (CAB) Member, Service Desk Agent, Problem Manager, Service Level Manager, etc.—to see and assess the risks of a change. Such oversight can help ensure an audit trail, and keep federated systems under management control.

- **Mapping and Visualization—**The CMDB has to be able to display its logical CI configurations visibly for its human users. Many systems use list hierarchies and folder and file metaphors to display their data. Some systems offer graphical representations of CI configurations. Although the underlying logic about relationships among CIs often can be learned from auto-discovery and application mapping tools, the finishing touches often require editing by hand.

The CMDB products from most vendors today usually offer at least one of these four key capabilities. While no vendor today offers all of them in a mature form, all claim that they will soon. When they do, customers can begin in earnest to explore and benefit from federated dimensional modeling. For example, they might use federated dimensional modeling for change impact assessment to understand the times customers use a system; and what locations have the highest number of users. Many other, common IT decisions require answers to questions about regions or locations, users and customers, projects, processes, dates, etc., and these could also benefit from federated dimensional modeling.

Another important use for a true CMDB is to enable development of Service Oriented Architectures (SOA), which rely almost totally on shared repositories for information about where services reside, their interfaces and so on. Bear in mind, however, that this federated multi-dimensional awareness (e.g., dates, locations, applications, users, etc.) makes the CMDB architecture quite difficult compared to the standard relational model, and it brings some unique issues.

For example, how do you handle a slowly changing data dimension? Consider an IT service where attributes for a CI vary over time—for example, a customer’s name. The CMDB system must have connections to the definitive source (federation), detect the change (reconciliation), determine the validity of the change (synchronization), store the old configuration, and make and display a new dimensional model for the new configuration (mapping and visualization.)

Each complete and approved configuration stored in the CMDB must exist for as long as required by the organization, so there may be several previous configurations stored at any given time. Each is a separate dimension that must remain intact, visible and accessible to users of the CMDB.

Industry Answers The Call

As IT organizations try to build their own CMDBs, they run straight into the wall of dimensional modeling and the four capabilities that make a true CMDB special: federation, reconciliation, synchronization, and mapping and visualization. Missing any of these converts any so-called CMDB into a simple ITAM, asset register or device configuration database, regardless of what the vendor claims or the builder believes.

The issues of federated dimensional modeling are just too large for most IT organizations to overcome. This makes it nearly impossible to build a true CMDB yourself—regardless of how good your team is—unless it is your core business function. In other words, unless you are in the business of selling CMDBs, it will probably be best to purchase one. Unfortunately, the cold hard truth is that there are woefully few vendors who even understand the issues required for a true CMDB, and fewer still with a functional solution.

But this is changing. In April 2006, BMC Software, IBM, HP and Fujitsu formed the Configuration Management Database Federation Working Group (CMDBf WG) to establish a standard architecture for a federated IT configuration management system incorporating data managed by products from different vendors. Microsoft and CA have since joined as well.

Microsoft is trying to patent CMDB nomenclature as its own

The CMDBf WG aims to develop an open specification for sharing information between CMDBs and their federated data repositories. After meeting regularly for nearly a year, the technical group within the CMDBf WG is close to publishing a white paper on the architecture. (At press time there was no CMDBf WG website.)

Should the CMDBf WG be successful, the result would be an industry standard for CMDBs that would dramatically streamline what is now largely a manual process. Focusing on solving the problems of dimensional modeling and automating federation, reconciliation, synchronization, and mapping and visualization, may help deliver on the promise of ITIL—increased efficiency, effectiveness, economy and equity.

To The Future, And Beyond!

There is no crystal ball with which to predict the future of CMDB technologies, but some very interesting activities are taking place. For example, consider United States Patent Application number 20060004875, in which Microsoft has applied for a patent on "CMDB Schema" that "includes an entity to store information identifying configuration items and a separate entity to store attributes of the configuration items. The CMDB schema may also include a separate entity to track relationships between configuration items." Sounds like Microsoft at least understands where the CMDB is heading.

Aside from joining the fray and trying to patent the CMDB, Microsoft is pushing SML (Service Model Language, formerly Microsoft's Systems Definition Model.) SML provides a language for defining models of datacenter resources such as servers, networking gear and storage. CMDBf WG is considering it, and IBM is backing it.

Interestingly, one of the biggest issues facing CMDB 2.0 is not the technology, but rather the organizational change that true CMDB implementation drives. A true CMDB by its very nature builds bridges between organizations, work groups and departments. A CMDB lets IT communicate to its customers and business managers about service and value, not costs and technology.

A CMDB also ties the IT department to the business. Without a CMDB, IT is not often credible, but with a CMDB, IT can show audit trails, including who uses which systems, when and what they did (or did not do.) A CMDB also allows IT to provide service in accordance with service level agreements (SLAs), to document service costs, and to accurately bill users for services.

On the surface this seems like a good thing, but issues of transparency are not always welcome, and the light of transparency can be harsh. Establishing a transparent IT organization also shows areas of weak and inefficient management, poor decision making, uncontrolled spending, and unreasonable behavior.

Conclusion

CMDB troubles occur when customers want something tangible before (or without) thinking about what they need to achieve. They start with existing ITAM or configuration database systems and try to grow their own CMDB. This is problematic since by definition a CMDB must be federated. According to Mathew Schvimmer, director of product management and marketing for ITSM at HP, any vendor creating a CMDB must take the position: "federate or become irrelevant." Schvimmer said, "Real-time modeling, defining IT services, federation between related data sources and data reconciliation are keys to CMDB success."

A CMDB that implements true dimensional modeling is difficult if not impossible to create on your own, although one can manually create and manage some workable CMDB functionality with today's technology. New technologies like SML, and industry initiatives like the CMDBf WG hope to address and automate CMDB's tough technological requirements. On the horizon, new product offerings from all levels of ITSM software vendors point towards a new technology base and robust products that will address the needs of those who choose to base their IT operations on the ITIL framework.

CMDB's are not simply ITAM asset registers or collections of device configurations. But if that's all you have, then you make do with it and improvise the rest. The good news is that help is on the way. The really good news is that the processes which must be established, the controls that must be implemented, and the lessons to be learned on the road to establishing an ITIL CMDB will serve the organization beyond the introduction of any number of new CMDB products□

Companies Mentioned In This Article

BMC Software (www.bmc.com)

CA (www.ca.com)

Cisco (www.cisco.com)

Fujitsu (www.fujitsu.com)

HP (www.hp.com)

IBM (www.ibm.com)

Managed Objects
(www.managedobjects.com)

Microsoft (www.microsoft.com)

TeamQuest (www.teamquest.com)

TouchPaper (www.touchpaper.com)