DITY™ Weekly Newsletter

itSM Solutions®
IT Experience. Practical Solutions

The Emergence of the Constituency-Driven CMDB System - ITIL's CMS in a New Context and What it Means for You

By Dennis Drogseth
Vice President – Enterprise Management Associates

The concept of the Configuration Management System (CMS) in ITIL v3 is potentially a game-changing catalyst in shaping Configuration Management Database (CMDB) deployments. If properly understood, the CMS can promote flexibility and innovation; if misapplied, it can lead to overarching complexity that can stall even the hardiest CMDB initiative.

ITIL v3's CMS opens the door to a systemic approach, helping CMDB planners, architects and process owners get beyond the implausibility of ITIL v2's single, monolithic database. It can enable a more direct approach in achieving time-to-value, thus simplifying the CMDB System effort.

This article suggests taking a "constituency" or "role/goal' approach to CMS design, based on pragmatism, readiness, and resource.

It is important to keep in mind that while the interest in CMDBs has become something of a tidal wave, CMDB System adoption, though well ahead of what it was two years ago, is still in its preliminary phases in most environments.

These CMDB programs are generally less than two years underway (many in the first six months), still somewhere in their first phase of deployment, generally without well-defined metrics, and most focused on seeking initial-phase proof of value. For these deployments, getting the CMDB to work sufficiently to demonstrate clear contributions in one or two areas is a primary goal.

Wisely, most CMDB deployments have plans for federation over time – usually starting somewhere quite specific. Example common starting points are a unique asset repository to complement the core CMDB, a Definitive Media Library (DML), or domain-centric repositories targeted at real-time information to reflexively look at the impacts of change from a performance perspective.

Why federation is beginning to take off

There are lots of reasons why federation is beginning to take off. One is simply performance. Depending on a single physical repository to house multiple hundreds of thousands of CIs and their attributes will eventually create performance issues.

Another is that most CMDB architects already recognize the difference between mapping the most critical devices and their application interdependencies, and being able to assimilate less critical infrastructure components, such as asset and inventory audits.

Or, in contrast, federation may reflect the need for a more granular component of the CMDB System to support change, configuration and release management for unique constituencies, such as switches and routers, or telecommunications devices, or, conversely, for in-depth change management of systems, or even desktops and remote devices.

Even vendor offerings are expressing the notion of federation implicitly, if not explicitly, in how some are packaging their CMDB solution. You can now find CMDB solutions for a service desk focused on incident and problem management, or asset management, or systems and network management configuration solutions, or service impact management solutions, or in some cases solutions more targeted at governance, compliance and security.

A prime reason for optimism in pursuing the federated CMS vision is that it increasingly reflects the two "parents" for CMDBs. The first is of course ITIL itself, with its focus on process and best practices. The second is largely architectural, as management technologies evolve to become more intelligent, automated and modular – with new types of integration across domains.

After all, the need for reconciled and consistent visions of "truth" is as old as IT, and CMDB-related technologies are now allowing IT architects to approach this problem in radically different ways. The sudden growth in CMDB System deployment represents a confluence of both of these ideas (process and architecture).

This is borne out in real-world deployments that typically must pair architectural with process expertise and evolving support for unique constituencies that, over time, will invariably lead beyond an all-or-nothing single repository and toward a more successful federation.

Getting ahead of the curve

The key point to make about this diversity is that there is not a single generically "right" place to start. There is, by contrast, a "right place for you" to start – and you will not find the answer for that in ITIL or from a vendor trying to sell you a CMDB solution. You need to assess your own operational and business objectives, dialog and communicate with key stakeholders, and judiciously decide where to begin based on readiness, resource, and impact.

The notion of a constituency-driven CMDB System may sound threatening and complex, but it can also be liberating – as long as you do not make the mistake of trying to do it all at once. You can start small (for example, several successful, germinal deployments have focused on life-cycle desktop and end-station asset management), or you can start with your most strategic core but keep the objectives finite, well defined, and still associated with clear objectives and a clear constituency.

An additional liberating thing about a constituency-driven CMDB System is that it will open up the market to multiple design points, optimized to support different processes and different roles within and even outside of IT. The challenge, of course, and the price of admission, is a normalization and reconciliation capability to support navigation across these sources with consistent CI and CI-attribute definitions.

But, I believe that there is good evidence to expect that the market will honor this requirement – there has already been some progress made through the CMDB Federation Workgroup and the DMTF.

Summary

If I am right, then, the coming years should witness a substantial amount of industry innovation as vendors with different strengths, constituencies, and design points contribute to the variety and depth of CMDB offerings – making it easier for you to target more specific, finite objectives with faster time-to-value.