Simplifying data center complexity

Do more, faster, with less

business white paper

As data center environments grow increasingly complex, CIOs and data center managers must balance new tools and technologies with increasing uptime, controlling costs and enhancing their capabilities. Now, there is an essential new standard that can help organizations realize the promise of utility computing and meet critical data center management objectives.

dcml Data Center Markup Language

Table of Contents

Introduction

Over the past decade, explosive growth in the availability and use of distributed computing technologies introduced unprecedented complexity into data center environments. A fundamental shift in computing architecture occurred: essential business applications rapidly migrated from client/server architectures running on a few, large servers to Web-based architectures running on thousands of smaller servers. The challenges created by this important shift included a dramatic need for IT reinvestment, retraining and organizational redesign. Unfortunately, the rapid pace of change left IT organizations struggling to manage the diverse skill sets needed for a fragmentation of operations technologies – including Web servers, application servers, database servers, network devices, storage fabrics, security systems, environmental management tools and others.

In most companies, IT specialization became necessary to ensure the continuity of daily business. The massive growth and commercial importance of server technologies forced specialization between the administration of desktop computers and servers. Data centers were specialized to handle large numbers of expensive, power-consuming servers. Accelerating complexity in Web-centric applications exacerbated the difficulty of software management chores. Rapid advancement of the Internet and Web-based communication further segmented IT personnel: Divergent technology platforms required different toolsets, different expertise and different management dialects.

Growing challenges in data center operations

Today, IT is no longer merely an internal supplier computing power. Instead, IT has been asked to become a highly specialized provider of application availability, delivering reliability, security and uptime. As Internet technologies presented unparalleled opportunities for competitive advantage, customer service and revenue generation, CIOs have been required to balance the mastery of new, often unproven, tools and technologies against three competing objectives:

- *Be fast* – Units of measure for development cycle times have moved from years to quarters or even weeks. IT organizations are expected to enable – and certainly never hinder – the ever-increasing pace of business.
- *Be cost effective* – IT is now a strategic weapon for most companies, but only when the outcomes are worth their cost. Facing sharp curtailments in corporate IT spending, CIOs must constantly produce new capabilities with decreasing total costs of ownership.
- *Be high quality* – Despite the relative newness of many distributed computing technologies, they form the underlying infrastructure for an enormous scope of business-critical applications. Degradations of quality or performance have become largely intolerable.

As an internal service organization, IT's methods, policies and physical data center arrangements have become less scrutinized than the ultimate delivery of the promised application – on time, on budget, reliable, scalable and highly available. CIOs and data center managers have been given fixed or shrinking budgets to build environments and to refine whatever processes, policies and legacy system links will ensure application uptime. IT is consistently and unflinchingly asked to do more, better, faster, with less.

The promise of utility computing

Corporate demand for reductions in the total cost of application ownership caused a significant reallocation of IT resources. IT executives and vendors are investigating and investing in "utility computing," "real-time infrastructure" or "organic IT" – the newest terms applied to systematic efforts to control IT costs and to do more with less. When taken to their logical conclusion, solutions based on the utility computing model can yield many benefits: minimize labor expense; virtually eliminate overcapacity; ensure the assignment of granular, consumption-based operating costs to business applications; and improve the overall agility of any IT organization.

Recognizing the vast potential of this promise, many IT teams are exploring a variety of projects and experiments that fall under the new heading of utility computing:

1. *Internal scripting* – IT specialists are producing a smorgasbord of tricks, tips, best practices and automation to reduce costs and eliminate human error.
2. *Automation systems* – A growing number of vendor-produced tools can enhance the efficiency and quality of key data center processes.
3. *Resource virtualization* – Expensive computing resources, including servers, network devices and storage systems, can be dynamically provisioned, torn down and re-provisioned as needed.
4. *Intelligent monitoring* – Data center monitoring tools can deliver best-practice issue resolution techniques at the same time they deliver system alerts.
5. *Policy enforcement* – CIOs are expending considerable effort to build systemic enforcement mechanisms for key IT policies and procedures.

Promises of five- to tenfold improvement in administrator-to-server ratios, marked quality improvements and radically reduced capital requirements will ensure utility computing initiatives continue to receive attention and funding. In fact, within utility

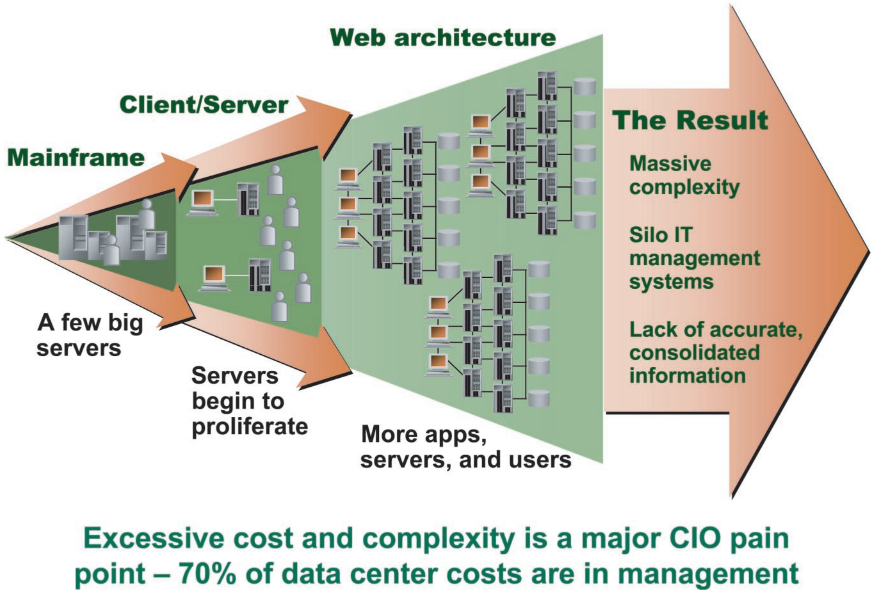

Figure 1: The evolution of enterprise computing

computing is the potential to transform IT into the low-cost, high-quality service provider needed by today's large companies.

However, the broad scope of the utility computing vision created a fundamental requirement that has not yet been fulfilled. For utility computing to make the leap from vision to reality, comprehensive interoperability and information exchange among its various components must be possible. Shared knowledge of the data center environment and its attributes is a prerequisite for the automated enforcement of policies, the nearly instantaneous provisioning of new equipment or the real-time delivery of intelligent alerts. For example, operating system utilities must be able to communicate with application server provisioning tools, which must be able to describe table requirements to a database, which must be able to request disk space from a storage array, which must be able to initiate an automated tape backup. All of these must interact with the workflow, billing, service initiation and management/monitoring systems, with security incorporated into the model.

Without a ready exchange of relevant data center information, the full benefit of utility computing will remain elusive. To date, the only method for producing comprehensive environmental integration is a massive, customized development effort – with commensurate costs and risks.

In every industry, standards arise when the value of a set of technologies is increased by each component's interoperability with the others. For example, the value of investment in early railroad networks was fundamentally constrained until a standardized rail gauge seamlessly linked network end points. Corporations elected to standardize on EDI protocols when the benefits of internal transaction systems became limited by an inability to interact with the transaction systems of suppliers and customers. Similarly, only a standard can unlock the full value of utility computing, which in turn will transform how modern data center environments are managed.

The remainder of this paper introduces and describes a new standard for unifying and extracting maximum value from utility computing. The Data Center Markup Language (DCML) is proposed to address the urgent need for interoperability among disparate data center elements, management tools, automation systems, and the policies that govern their use.

Figure 1 represents the transformation from centralized/terminal-centric computing to distributed/departmental computing to highly capable, intelligent, network computing heterogeneous architectures. Although the evolution has provided mainframe-level capabilities in desktop/modular-level configurations, the complexity of management has devolved as technology has evolved, leading us to where we are today – disparate management constructs with little to no normalization, and with very little utility-readiness. DCML looks to provide the means by which the computing-to-utility management and readiness transformations converge.

Introducing the Data Center Markup Language

The Data Center Markup Language is the first standard to provide a structured format for describing the complete data center environment. It is a vendor-neutral data model governing the capture and exchange of information for system configuration, device interconnection and policy enforcement. Disparate utility computing components, as well as legacy enterprise environments within the data center, all can use DCML to achieve environment-wide interoperability and comprehensive knowledge sharing. The DCML standard will be the first to optimize and coordinate a view of the data center as a complex system of interacting elements.

The right time for a standard

Within the emerging category of utility computing is a considerable assortment of internal and vendor-developed tools. Each of these utility computing technologies performs a different function in the data center: some are complete platforms automating the full lifecycle of servers and software; some are solely focused on infrastructure virtualization; some enable intelligent system monitoring; and others are point tools performing highly focused tasks such as device provisioning or application patching. As IT organizations increasingly conform to the utility computing model, they will have to adopt more than one solution because of gaps in tool functionality. That is because no single vendor is building a complete system; currently, the vision and objectives of utility computing are unobtainable without multiple vendors and systems.

Regardless of originating vendor, all utility computing systems have a common dependence: a thorough understanding of the environment under management. However, one tool's definition of "environment under management" and the format used to capture its key attributes are likely to be entirely different from another tool's definition and format. Today's definitions of data center environments are usually narrowly scoped, and the format of the requisite environmental information is always proprietary. As a result, it is practically impossible to share knowledge between different utility computing systems.

With the industry only at the beginning of the utility computing adoption curve, interoperability is essential to the continuation of progress. The current economic environment will not permit protracted, expensive integration efforts. Corporate IT simply cannot afford to introduce disconnected utility solutions – the complexity of such efforts would ultimately degrade, rather than improve, the overall manageability of an environment. Today's utility computing systems are the first phase of a trend that can produce vital and timely benefits. However, for those systems to complete their transformation of modern data center management, there must exist a broader view, a unifying standard, a framework to tie together and facilitate the exchange of critical environmental information.

DCML is that standard. It is designed to remove the obstacles limiting the promise of utility computing by enabling the interoperability of its component systems, automation platforms and legacy management tools. CIOs are clearly pursuing a more holistic approach to managing server environments. IT organizations are aggressively investing resources in systems to streamline a variety of management tasks. Yet, without a technology like DCML, "local" optimization may ultimately prevent "global" efficiency. Without DCML, disjointed definitions and proprietary information may unduly restrict the long-term benefit to be gained from utility computing.

Benefits of a data center standard

The Internet and World Wide Web are obvious and extreme examples of the benefit of technology standards. Every Web server, network device and e-commerce application was fundamentally valueless (and therefore nonexistent) until standards such as HTTP, HTML and TCP/IP became sufficiently widespread. These standards were adopted across the breadth of technology vendors; nonadoption was simply not an option. Ultimately, the market viability of all Internet technologies was enhanced by the comprehensive applicability of these standards.

As a result, a connected server is able to maintain an awareness of, and communicate with, all application software, network devices, storage systems and counterpart servers required for its assigned function. Today, the weaker link in data center management is not the interoperability of its resident technologies, but rather the lack of a standardized information exchange between those technologies' management elements. For example, well-defined protocols exist for an application server to query and extract records from a database. However, standards do not exist for the rapid reconfiguration of the database after an application server upgrade. Expensive, manual intervention – or nonscalable, idiosyncratic automation – is required. There is no uniform language to establish, manage or maintain the critical relationships of a complex data center environment.

Only a standard can unlock the full value of utility computing, which in turn will transform how modern data center environments are managed.

DCML is modeled on today's Internet communication standards. The success of those standards illustrates three parameters necessary to capture, codify and disseminate contextual management parameters:

- *Machine readability* – To reduce reliance on human intervention, DCML-compliant information about the state and configuration of data center elements must be emitted and consumed by software rather than by administrators.
- *Vendor neutrality* – DCML must be useful to management technologies from a variety of vendors; no single vendor produces all utility computing and management components in a modern data center.
- *Comprehensive scope* – The breadth of technologies used in server-centric data centers requires that DCML communicates with all logical and physical components of environmentals, servers, application software, security/network devices and storage arrays, as well as all policy-enforcement mechanisms.

The IT industry's well-established approach to standards development will provide an essential degree of uniformity and openness to the ongoing evolution of DCML. Open working groups and independent evolution based on an open specification are proven tactics for ensuring that any standard meets current and future market needs. The collective effort of participating organizations, companies and individuals can be leveraged to maximize the value of DCML and, as a result, any investment in utility computing.

Today, the true benefits of utility computing remain obscured behind a tangle of disparate management and automation systems, provisioning tools, and vendor-specified software configurations. However, by relying on the described principles, DCML can unify the wide variety of available utility computing systems. It will provide a single format to describe, configure and manage virtually every component and attribute of a distributed computing environment. A data center can be quickly built, efficiently managed and easily replicated.

DCML defined

A DCML file is both a recipe and a blueprint of one or more data center environments. Much as a culinary recipe provides the list of ingredients and the instructions for successfully combining them, DCML provides an inventory of data center elements and the desired functional relationship between them. Just as an architectural blueprint establishes an easily understood, multidimensional plan for constructing or replicating a building, a DCML file can be used to provision or reproduce a complete data center infrastructure – with all of its component relationships, dependencies, configurations, operational policies and management processes.

Each utility computing system in a data center produces management instructions governing specific behaviors of a relevant subset of data center components. However, DCML compliance will enable these same utility technologies to transcend the relatively simple duties of awareness and governance. For example, by relying on a DCML "recipe," a server provisioning script will recognize the need to check a newly provisioned Web server into a load balancer. That same recipe, now updated to reflect the presence of a new Web server, also will trigger a DCML-compliant monitoring system to add the server to its list of monitored elements. Similarly, the "blueprint" aspect of the revised DCML file will provide instructions to an automation system for updating a disaster recovery site. Or, a security tool will be informed of the need to monitor differences between the new blueprint and the actual physical environment. In general, utility computing components require the blueprint and the recipe to establish and maintain full interoperation.

A DCML file offers a unique depth of valuable management and configuration information. Attributes of key environmental building blocks, such as software versions and application configurations, can be easily adapted to slightly different environments. Variations in hardware configurations, network connections and data center variables can be emitted and consumed by DCML-compliant systems. These, in turn, can facilitate rapid, accurate provisioning and change management. DCML also can describe the policies initially necessary to construct the environment. Those policies can then be compared to physical reality to ensure procedural compliance and to provide a snapshot of environmental state.

Ultimately, DCML-compliant systems will accelerate the rate at which IT managers can automate as much of their environments as is feasible and beneficial:

- First, a comprehensive DCML-generated blueprint of the environment will describe how it was constructed, including hardware requirements, software components and configurations, networking, and storage. All must be specified to facilitate automation.
- Second, a DCML-generated recipe will set the policies and relationships governing the interconnection of these environmental components. Approved and banned versions and combinations of various technologies, the order in which software is installed and started, the best-practice configurations to be used, and the required support from underlying hardware – all must be specified to ensure the resulting environment conforms to organizational policy and procedure.

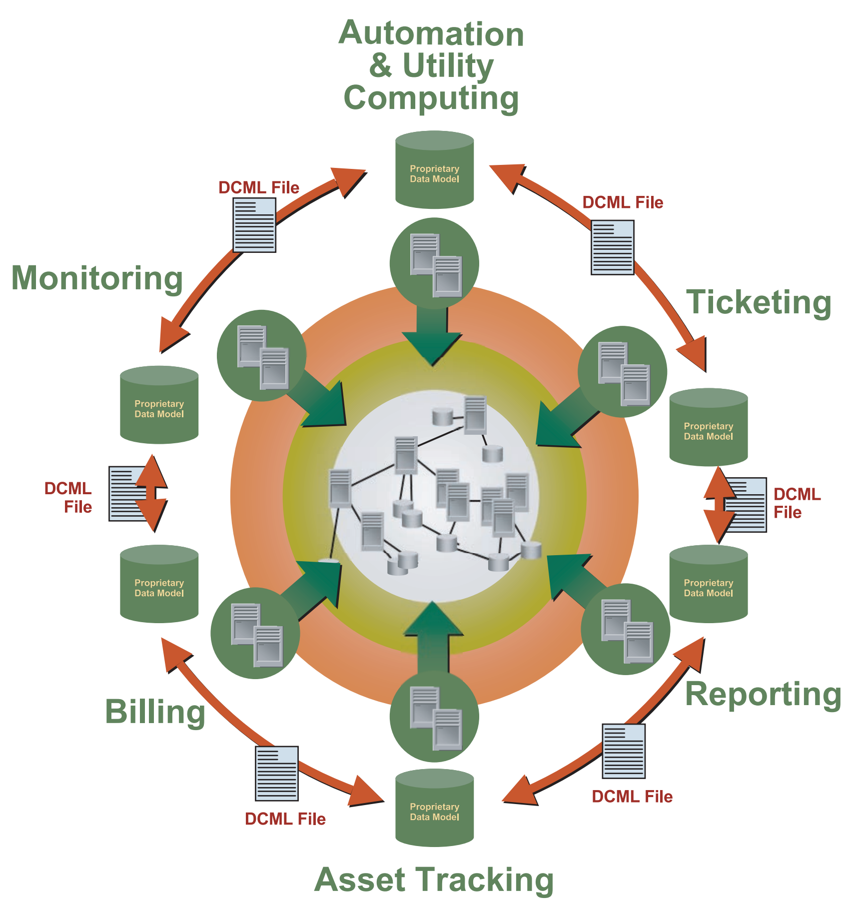

Figure 2: DCML in the data center

A DCML-based description of any environment can provide all of these data in a form and format useful to every system – new or legacy – operating in a data center environment. The result is a powerful hierarchy of critical management information that can unify virtually any combination of utility computing technologies.

Using DCML in the data center

The primary objective of DCML is to harness information in traditional, existing environments and to migrate/transition that information to utility/grid IT environments. Each of the following use cases illustrates the value and unprecedented functionality afforded by a vendor-neutral exchange of key environmental and component-level information.

Case 1: Promotion of code from development to production – The working relationship between software development and data center operations is critical to any IT organization's ability to respond to fast-changing business requirements. As development teams complete new application functionality, engineering will work with operations to ready the production environment for promotion of the new code. Operational requirements can range from the simple availability of disk space to the replication of an entire data center environment. Although most code rolls fall somewhere between those extremes, the importance of corporate change management processes demands a unique degree of formality and rigidity. Utility computing technologies can help ensure process compliance and a successful outcome, but only if those technologies are sufficiently interoperable.

DCML-enabled interoperability limits human intervention during code promotion to change management process oversight, quality control and the initiation of any necessary change rollback. The result is a faster process with fewer opportunities for human error. The application itself describes the environmental requirements; systems, rather than people, exchange information to establish suitable configurations. Engineering focuses exclusively on meeting the business requirements for the application, while operations can safely revert to a role of oversight and exception handling. The quality of the outcome – as well as the quality of the development/operations relationship – is vastly improved.

Without DCML, disjointed definitions and proprietary information may unduly restrict the long-term benefit to be gained from utility computing.

Code promotion *WITHOUT* interoperability	Code promotion *WITH* DCML-enabled interoperability
Development engineers produce an inventory of expected hardware and software requirements for the new code.	An application development tool emits a DCML description of the required production environment at the time the code is frozen and readied for promotion.
Administrators prepare the physical environment for the new code, provisioning OS-ready servers and configuring the network according to known requirements.	DCML-compliant systems managing the production environment consume that description to provision the required servers, configure network devices, update monitoring and security systems, provide utilization information to legacy financial systems, etc.
Engineers and administrators work together to configure the environment, methodically iterating through the various components needing modification to handle the new application functionality.	Web and application server configurations, load balancer configurations, database schema modifications, storage needs, etc. – derived from the parameters described in the DCML file – are all automatically generated and readied for installation.
Following successful installation and testing of the new application, downstream systems (such as security, monitoring and asset management) are updated.	

Case 2: Disaster recovery – Over the past two years, virtually every CIO has been asked to build or bolster a disaster recovery capability. However, the described growth in distributed, Web-centric computing has fundamentally changed disaster recovery economics. It has become financially and procedurally infeasible to duplicate an environment that will be maintained exclusively as an insurance policy against disaster, natural or otherwise. Instead, computing assets must be flexibly utilized and readily shifted from routine functions to disaster recovery purposes, whenever and wherever necessary. Only utility computing systems can produce such agility, and only fully interoperable utility solutions can bring the cost of disaster recovery capabilities in line with the benefit they provide.

In both cases, data snapshots are physically transported, usually by magnetic tape, to the secondary site for loading. However, in the case of DCML-enabled interoperability, the configurations of the site are also moved to the secondary site – not via magnetic tape, but contained within the DCML description of the primary site. Without DCML, the capture and replication of critical configuration information is inefficient, inaccurate and incomplete.

In addition, the utility computing components and automation systems in each site need not be identical. Less expensive management systems can be used in secondary sites. As long as DCML compliance is maintained, all systems can exchange relevant management information about the data center environment. Overall, with DCML, the value of disaster recovery capabilities relative to their cost becomes decidedly more attractive.

DCML will provide a single format to describe, configure and manage virtually every component and attribute of a distributed computing environment.

Disaster recovery *WITHOUT* interoperability	**Disaster recovery *WITH* DCML-enabled interoperability**
Server and networking assets are deployed to a disaster recovery site, but are typically given dual roles: (1) be available in case of disaster; and (2) provide computing power for noncritical or secondary applications.	Server and networking assets are still deployed to secondary sites with dual roles, maximizing their productivity.
In the event of disaster or prolonged outage at the primary site, the secondary site is wiped clean and manually reconfigured to match the latest specification of the primary site.	In the event of disaster, systems in the secondary site emit DCML descriptions to be used to return the site to its original functionality after the crisis has passed.
A massive testing effort attempts to identify discrepancies between the new site and last known version of the now-dark site. Problems are iteratively corrected until functionality subjectively approximates the predisaster state of the application.	An up-to-the-minute DCML "recipe" of the primary site is consumed by relevant systems in the secondary site. Server, networking and storage devices are automatically prepared for the deployment of the primary site's configurations.
	The DCML "blueprint" provides instructions for quick and accurate replication of all configurations, policies, dependencies and interconnections that governed the primary site. Manual intervention is limited to exception processing.

Case 3: Server consolidation – By reducing the number of computing assets dedicated to a given application, CIOs can reduce the total cost of ownership assigned to it. For example, an application with components distributed over two or more physical sites can be consolidated into one data center, eliminating the allocated fixed cost of the second (or third) site. In another example, several lightweight applications running on smaller, dedicated servers can be consolidated onto a single, larger server, sharing the resources of that server as needed. However, both consolidation examples – and any similar efforts – expose the consolidated applications to substantial risks of downtime and instability. Hardware virtualization and other utility computing systems were designed to support such efforts, but their interoperability is a prerequisite to a reduction in risk sufficient for wide-scale server consolidation.

As IT environments continue to expand, and as Web-based applications continue to power the newest IT initiatives, server consolidation will be an important and necessary element of infrastructure cost control. Technologies such as hardware virtualization will become increasingly mature and will provide increasingly robust toolsets to CIOs seeking opportunities to reduce per-application costs. Nonetheless, these tools – and all other utility solutions – cannot produce the promised returns on investment without the ready exchange of environmental information afforded by DCML.

Server consolidation *WITHOUT* interoperability	Server consolidation *WITH* DCML-enabled interoperability
Consolidation candidates are identified through *ad hoc* discovery and capacity-planning efforts.	Consolidation candidates are identified but are more easily targeted via DCML descriptors of resource consumption and service level sensitivity (for example, the tolerance to the risk of consolidation).
Destination environments are prepared for the introduction of the consolidated applications; hardware and software is purchased/reallocated and (re)purposed for the inbound applications.	DCML files are emitted to describe the applications' existing relationships and configurations in the current environment.
The applications are migrated, and the new environment is appropriately configured. Third-party virtualization technologies are configured to support the consolidation.	Virtualization technologies and other utility computing systems in the destination environment consume the DCML to prepare computing resources for the inbound applications.
Monitoring systems are updated and closely watched for indications of instability in the consolidated applications.	Monitoring systems consume additional DCML to identify adjustments necessary to compensate for the new configurations.
Internal billing systems are updated to reflect the new infrastructure costs.	DCML-compliant asset management links automatically ensure proper internal billing.

DCML technical constructs

The DCML specification is based on an extensible markup language (XML) schema that describes the DCML constructs, syntax and semantics. DCML-compliant utility computing elements will emit and consume XML file descriptors encompassing a complete data center environment. As the common language of information exchange on the Internet, XML is a widely implemented, well-understood technology. XML parsers and generators are readily available from a number of sources, making implementation relatively inexpensive.

The DCML file itself will contain a machine-readable hierarchy of critical environmental information, including:

- *Server configurations* – Hardware specifications, basic input/output system settings, networking settings, operating system (OS) loads and patch levels, etc.
- *Software configurations* – Installed packages or images, required installation sequences and patch levels, functional configurations (for example, configuration as an application server versus a database server), hardware dependencies and others
- *Applications* – Descriptions of complete n-tier business applications composed of Web servers, application servers, database servers, and/or various code and content modules distributed across all tiers and ancillary systems
- *Environmental lifecycle* – Classification of groups of systems as development, build, test, staging/stress testing, production, and/or disaster recovery environments consisting of servers, software, application code and others
- *Networking* – Firmware versions, software, patch upgrades/changes, administrative protocols and configuration settings for switches, routers, bridges, access devices, load balancers, Web accelerators, and other network services/appliances
- *Security* – Firmware versions, flash configurations, software, patch upgrades/changes, administrative protocols and configuration settings for firewalls, intrusion detection/prevention devices, anti-virus technologies, and all facets of security in the enterprise
- *Storage configurations* – Disk space allocations and configuration files for storage arrays, network- and direct-attached storage devices, SAN switches, etc.

- *Data center* – Complete hierarchical combinations of previously described servers, software, applications, production environments, networking and storage
- *Environmentals* – Enterprise requirements of the data center itself (including power, cooling, floor space and storage systems physical configurations, including networking security and other physical systems). These also include the blueprinting constructs of all systems within an environment and their individual power and cooling requirements.

DCML forms an extensible data model and a corresponding vendor-neutral schema that describes the physical and logical components in data centers, their relationships and interconnects, how to build and configure those components, and key operational policies governing their use. DCML will unify a virtually limitless number of utility computing elements, automation tools, legacy systems and data center management techniques. The hierarchy of information provided by a DCML file will form a recipe for combining these elements into one or more cohesive, flexible, integrated and cost-effective data center environments.

DCML's impact on the IT industry

The enormous diversity of systems, tools, methods and policies employed to manage a data center makes creating a standard to govern them a daunting and potentially risky task. Yet the rewards of success are enormous for IT executives and IT vendors. For CIOs, DCML is the prerequisite to the tangible benefits of utility computing technologies. For IT vendors, DCML represents opportunities to bring unprecedented value to customers in the form of reduced costs of ownership, easier implementation, higher quality and previously unattainable leaps in innovation.

Why will DCML provide benefits not otherwise achievable? The opportunities and value contained in the DCML standard are derived from its reflection of utility computing principles as well as from its unique approach to describing the entirety of a data center environment.

Delivering on the promise of utility computing

Without DCML, environmental information and system management attributes are captured and disseminated via a variety of largely manual means. System administrators and subject-matter experts use their memory, expertise and knowledge of existing systems to construct an environmental map of the data center. Changes to the environment are manually captured and sporadically communicated. Disk images of machines are created and used *ad hoc* to reproduce system configurations. Disk contents are backed up to tape and stored offsite for later retrieval.

Problems with these approaches are numerous. Human fallibility slows execution time, creates inconsistency and reduces quality. Incompatibility between vendor platforms prevents comprehensive representation of required management information. Hardware, network and data center dependencies change, rendering previously established configurations nearly useless for provisioning or reprovisioning purposes. Durations for backup and restore processes, combined with unsolved technical difficulty obtaining a consistently usable backup image, significantly reduce the effectiveness of backup systems.

Utility computing models and systems were created to solve these and similar problems. However, as discussed, most utility computing solutions have been tailored to address a narrow range of data center management tasks. There has been no standard for representing the exchange of information they need to succeed together. DCML provides the first data center standard that reflects the original objectives of utility computing to:

- Describe how to construct, replicate and recover systems under management – not only the state of those systems at a point in time
- Capture the best practices, operational policies and standards critical to reducing operational costs
- Disseminate information that helps extract additional value from existing management solutions, regardless of whether the solution was purchased or built in-house
- Specify the data model and semantics of a data center while enabling flexibility on communication protocols and end systems capabilities

Through these means, DCML can deliver IT executives the full promise and tangible benefits of utility computing.

Creating new opportunities for innovation

DCML opens important new avenues for IT innovation. The interoperability established by the DCML standard enables the IT vendor community to begin to address customer requirements with unprecedented creativity and far-reaching results. The examples below begin to illustrate the scope of DCML's potential impact:

- *Server and OS platform* vendors can significantly increase the simplicity and speed at which their technologies are provisioned. Widespread use of DCML-based specifications can enable scaling from 10 to 100 to 1,000 to 10,000 servers with equivalent ease.
- *Application software* vendors can improve the adoption rate of upgrades and new products by consuming DCML descriptions to configure upgrades automatically for seamless integration into an existing environment.
- *Network equipment* vendors can improve the affordability of disaster recovery and geographic load balancing by using DCML to establish and maintain configuration parity between primary and secondary sites.
- *Storage systems* providers can work to integrate their device management technology with DCML-compliant systems to lower the total cost of ownership for high-end storage systems.
- *Monitoring and management system* providers can offer more detailed and comprehensive environment visualization tools by consuming DCML-compliant descriptions from an increasing variety of data center elements.
- *Asset management system* vendors can reduce their customers' integration costs by establishing automated, DCML-compliant technologies that rapidly link financial systems and data center management systems.
- *Automation systems* vendors can deliver to customers an extensive collection of automation modules and can provide a centralized interface for managing heterogeneous DCML-compliant systems.

Overall, DCML will coalesce utility computing technologies and their vendors around a unified goal of helping IT organizations do more, faster, with less.

Conclusion

Utility computing holds the real promise of bringing order to the cacophony of tools and technologies found in today's data center environments. The explosion of complexity and scale in the data center, combined with ongoing corporate pressures for higher quality at a lower cost, has established utility computing as an important trend in IT – a trend that shows no signs of diminishing. As CIOs grapple with the competing goals of speed, cost and quality, they likely will continue to invest significant resources in investigating and implementing utility computing.

However, as utility computing technologies strengthen their hold on the data center, most CIOs will see their environments grow more complex rather than more efficient. A growing breadth of disconnected, nascent management technologies will present CIOs with a decision: Either kick off enterprise resource planning-style integration projects, or stop investing in utility computing. Given today's economic climate and the potentially enormous benefits of utility computing, neither option is tenable. Only a standard can provide the interoperability, cohesion and information exchange needed to prevent utility computing from simply becoming a resource consumptive pipe dream. DCML is the standard that can hasten the maturity of utility computing from an interesting trend to an archetypal technology.

About the DCML Organization

The DCML Organization is an open, independent, vendor-neutral, non-profit corporation being formed to create an open, freely licensed specification, Data Center Markup Language (DCML), and to encourage its broad adoption. DCML is the first standard that provides a structured model and encoding to describe, construct, replicate and recover data center environments and elements. DCML is designed to provide a mechanism to enable data center automation, utility computing and system management solutions to exchange information about the environment to make utility computing a reality. In addition to developing specifications, the organization intends to work with formal standards bodies, enable and administer certification and compliance programs, and perform user and market education. For more information about how to join the DCML Organization, or to learn more about planned activities and DCML, visit www.dcml.org.

dcml Data Center Markup Language