Sign up at **www.architectureandgovernance.com/subscribe**

JOIN THE COMMUNITY

ARCHITECTURE &GOVERNANCE

VOLUME 6 ISSUE 5

magazine

CURING THE FEDERAL EA HANGOVER

- An "EA hangover"—a lingering state of discomfort driven by the repercussions of past failures and looming future obligations—exists in many organizations throughout the government.

By Ted Reynolds

Long regarded as a key thought leader in the evolution of enterprise architecture (EA) and strategic decision support, the U.S. Federal Government has made significant progress in advancing best practices to create wholly new and improved ways of addressing decades-old problems. Federal policies, mandates, and other guidance have collectively served as a governance playbook for our government as well as many other governments around the globe. In addition, advances in EA management, capital planning, and performance management have been formally adopted and institutionalized by industry. However, despite all of the hard work and external

MORE ON PAGE 2

ARCHITECTURE
&GOVERNANCE
magazine

PUBLISHER
Matt Price

EDITOR-IN-CHIEF
George S. Paras

MANAGING EDITOR
Holt Hackney

ART DIRECTION & DESIGN
BNewton Associates

Architecture & Governance
8601 RR 2222
Building 3, Suite 300
Austin, TX 78730
512-536-6270
www.ArchitectureandGovernance.com

validation, the majority of federal agencies have struggled to deliver on EA's great promise—to deliver measurable, repeatable, and transformative value to their stakeholders.

> While there's no single antidote or quick fix to eliminate all the headaches, there are common patterns that can be identified and mapped to past successes and failures.

Historically, one of the main inhibitors has been the imbalance between external demands on agencies and their resources to meet those demands. As time progressed, the importance of EA grew, and compliance requirements became more demanding. Agencies were unable to scale their teams, tools, and governance practices to keep pace. Consequently, strategic agency-specific transformation initiatives suffered and were deprioritized. Any cycles remaining from addressing data calls were typically used by agencies to take a reactive, "bottom-up" approach to creating architectures, focusing on the tangible elements of their enterprise (e.g., IT) and leaving the critical strategic and business elements for the future. These factors catalyzed a self-perpetuating disconnect in which an agency's increasing need to deliver value from its EA program became inversely proportional to its ability to do so.

In recent years, policy makers have noted these issues and addressed them by providing more focused and tangible EA program guidance. The federal enterprise as a whole was deliberately segmented into more manageable "slices," and the concept of *segment architecture* was born. The concept of "value" was reassessed, and foundational mechanisms to quantify and track performance were refined through specification of the Model for Performance-Driven Government (MPG). Current administration initiatives have highlighted the importance of *transparency*, *accountability*, *collaboration*, and *open government*, prompting agencies to deliver even more tangible EA program value to the business.

All of these signs suggest that we're moving in the right direction, yet anecdotal evidence suggests that an "EA hangover"—a lingering state of discomfort driven by the repercussions of past failures and looming future obligations—exists in many organizations throughout the government. Many new CIOs and EA program managers try to minimize the discomfort with slow-rolling EA initiatives for fear of repeating their predecessors' mistakes. Where prior EA program efforts have failed, the new leaders are left without the clout to successfully battle for budget dollars and resources. This perspective trickles down to the EA practitioners, who quickly realize they, too, don't have the charter or resources to take their creative ideas to fruition and, accordingly, take a path of least resistance. Over time, this collective mindset and corresponding inability to deliver real value serves to perpetuate the belief that EA is just a "low-value, check-the-box exercise."

MORE ON PAGE 3

CURING THE HANGOVER

It's reasonable to assume that the gap between demands and resources will continue to widen. Given this expectation, how can agencies ever gain the momentum to ultimately cure the EA hangover? While there's no single antidote or quick fix to eliminate all the headaches, there are common patterns that can be identified and mapped to past successes and failures. Given this information, agencies should be able to identify ways to navigate around some of the impediments to success.

As a vendor in the federal EA marketplace for many years, we've seen these patterns unfold and repeat themselves. This has enabled us to identify critical success factors and common impediments that agencies run into. Here are the most common impediments and our recommended mitigations.

★ ★ ★

IMPEDIMENT 1:

Failing to understand key stakeholder needs

This is a simple concept; yet due to the reactive nature of compliance-driven EA, many agency-specific EA initiatives fail to proactively identify important agency beneficiaries or their needs—i.e., they start as a "solution in search of a problem." These EA teams spend significant cycles in a trial-and-error discovery process until key value drivers are identified (or not!). For the fortunate projects that don't run out of time and money, the perceived ROI of the EA initiative diminishes over time with stakeholders, and the projects increase their risk for cancellation.

RECOMMENDED MITIGATION:

✔ **Define an EA capability road map.** Just like other agency programs, the EA program should have a strategic plan and define a capability-based road map that outlines how specific value will be delivered over time. To define the road map, EA practitioners will need to network with key stakeholders and determine the tangible benefits each stakeholder expects from the EA and in what time frame they expect to receive them. This will help identify EA project goals and the EA capabilities needed to achieve them, thereby focusing efforts on targeted, high-value areas. Additionally, this process will allow the EA team to establish and grow the base of support that will be critical to future success.

★ ★ ★

IMPEDIMENT 2:

Going too high or too wide, too early

Many of the failed federal EA programs we've encountered focus initially on achieving a laundry list of lofty goals, such as eliminating all risk and unaligned spending from an agency's technology portfolio. Such transformation is certainly a worthwhile long-term goal, but it takes significant time and effort, and most stakeholders expect a relatively immediate return on their investment. Trying to accomplish too much in too short a time is a significant risk to EA project success.

RECOMMENDED MITIGATION:

✔ **Right-size the scope of your EA road map.** Allow for realistic goals and delivery time frames. Consider a dual-track road map that balances achieving lower value, but quick win goals with ones that require longer time frames but deliver high-value sophisticated capabilities. Combined with outreach to the organization, this approach will ensure that the community is constantly aware of progress and quickly consuming the valuable capabilities the EA program is providing, while also addressing the long-term, strategic EA program road map.

★ ★ ★

IMPEDIMENT 3:

Waiting for policy makers (or the next contract) to define EA project goals

We have seen many agencies continue to take a reactive wait-and-see approach to defining EA project goals. This is mainly due to the risk-adverse demeanor of most EA program/project leads. They believe it advantageous to wait for guidance rather than assume the risks associated with innovation or changing course later. Contributing to this risk-aversion is the common misperception that EA demands long development cycles and the short-term nature of EA contracts. EA leaders are deterred from addressing high-value initiatives due to the belief that there won't be enough time to deliver value.

MORE ON PAGE 4

RECOMMENDED MITIGATION:

- ✔ **Identify long-range strategic agency initiatives and associate EA as a value driver.** These high-visibility initiatives will transcend policy and contract cycles and provide a framework around which to build EA project plans. If EA work items are left unfulfilled at the end of one contract, you should ensure these items are aligned to the performance criteria of the next contract. This tactic will ensure continuity of the EA road map and, ultimately, deliver aligned value more consistently.

★ ★ ★

IMPEDIMENT 4:

Confusing "activity" with delivering value

The well-known mantra "just because you can, doesn't mean you should" should be reinforced several times throughout an EA project. Many projects begin with an uncoordinated scavenger hunt for readily available and easily consumable data. While this method may occasionally yield worthwhile value, the approach is generally unsustainable in all but the most flexible and small-scale environments. Key stakeholders are given the false impression that the EA team and project are making progress, which only magnifies the impact of failure if/when it arises.

RECOMMENDED MITIGATION:

- ✔ **Focus EA teams by defining "value" granularly and regularly assessing what it means to achieve it.** It is critical to ensure that the EA road map defines tangible benefits that will be realized through the delivery of specific capabilities. Of equal importance is the regular assessment of how well capabilities align with expected benefits. This evaluation is typically facilitated by incorporating performance metrics (e.g., PRM Measurement Indicators), which identify benefits in quantitative terms. This approach encourages EA practitioners to stay focused on delivering aligned value via regular feedback on their progress.

★ ★ ★

IMPEDIMENT 5:

Forgetting to communicate success early and often

Just like a commercial product or service, it is critical that the EA team devise a structured communications plan that will immediately and consistently market the value being achieved by the EA program to key stakeholders and the wider community of beneficiaries. As soon as value is first realized, success should be declared and communicated to provide leverage for the next phase. This outreach process must be ongoing. If not, stakeholder support and resources may "dry up" or be redirected toward projects with greater visibility or perceived value.

RECOMMENDED MITIGATION:

- ✔ **Focus on creating "value guideposts" ASAP.** With any change in ways of working can come an initial period of "way finding" that delays early progress toward value. It is important to overcome this as quickly as possible through some form of "value guideposting." These are like trail markers identifying points of interest along the way. Each value guidepost provides a well-defined capability, such as a hands-on demonstration, that's *achievable* at that point on the trail. Just like trail markers offer handouts you can take with you, provide materials that communicate the guidepost capability benefits in a language the intended beneficiaries can understand. Defining such value guideposts will help the EA team get out of "firefighting mode" and focus on a stable path of delivery throughout the project.

CONCLUSION

While there's no quick fix to eliminating the EA hangover, once agencies identify and address these impediments, they will remove barriers to delivery of agency-specific value from their EA program. This requires a balanced approach, whereby EA teams continue to adequately meet compliance requirements while proactively addressing strategic agency initiatives. Whether the opportunity is datacenter consolidation, standards management, or other high-visibility initiatives, approaching it with these techniques in place will positively influence the outcome and get your EA program the recognition it deserves. **A&G**

TED REYNOLDS is the director of the Federal Solutions Group at Troux Technologies, Inc.

Strategic IT Planning software gives CIOs the power to drive business change. Fast.

☑ **Troux**

☐ **false**

You need to make important business decisions.

You need to understand and communicate how all aspects of IT support your business goals, strategies and capabilities.

You need to talk to Troux.

We've developed the world's finest software for aligning IT decisions to business strategy. You'll execute enterprise wide transformation initiatives and meet the strategic imperatives of your company with an unprecedented level of confidence. You'll drive agility and innovation across the entire organization with the support of your peers. Not only will you keep up with business priorities, you'll measure IT success in terms of business results.

That's huge. That's Troux. That's what we are all about. Transformation at the speed of business.

Find out for yourself. Let us show you exactly how your company can put its trust in Troux to meet business expectations. Visit **www.troux.com/jointherevolution**

troux
technologies

Take Charge of APPLICATION INTEGRATION CHAOS

By Carrin Tunney and Srini Sastry

At a high level, application integration means leveraging technical infrastructure to make diverse applications communicate with each other to achieve a business objective. The integration needs to be seamless and reliable, regardless of platform and geographical location of these applications.

The move toward service-oriented architecture, business process management, and software as a service has accelerated the recognition that application integration can increase business efficiency. To ensure that the integration is both beneficial and feasible, one should closely examine the business processes that are being supported *before* focusing on the systems and technologies themselves. Understanding this is the key to determining how to select the most suitable integration technologies.

This article examines the application integration challenges while considering the enterprise diverse technology landscape and architectural concerns. The presented "scenario-driven approach" describes how to successfully implement application integration standards at the enterprise level, leveraging the TOGAF methodology, and, ultimately, supporting the business capabilities.

CHALLENGES OF APPLICATION INTEGRATION

Making suboptimal or incorrect choices when selecting your technology toolset can lead to complex architectural issues, which in turn lead to tightly coupled systems and support and maintenance problems. Moreover, guidelines and best practices espoused by architecture groups are not consistently written down, nor are they consistently followed by application development teams at large. Without clear standards on when to use which technology, and for what purpose, one runs the risk of creating an unnecessarily complex technology environment.

Many enterprises have also suffered from organic growth and now have every integration technology of the last two decades in place: JDBC/ODBC, database links, remote method invocation (RMI), enterprise Java beans (EJB), Web services, Java message services (JMS), MQSeries, and flat files abound. This technology diversity has resulted in increased support costs and has adversely affected system performance.

MORE ON PAGE 7

Additionally, there are organizational barriers that may impact your application integration efforts as well. These issues typically arise from the fact that enterprise IT systems span multiple departments in the organization. Different development teams in the enterprise, if not properly guided or monitored, may tend to choose the path of least resistance or resort to technologies that they are familiar with, which can add to integration complexity.

Organizational issues in application integration can be tricky in a larger organization. However, these companies often have the greatest need for an effective application integration environment. Therefore, it is particularly important to clearly define standards, so that development groups can focus on business needs.

LEVELS OF APPLICATION INTEGRATION

Application integration can occur at many different levels of a system's architecture, including the data layer, the application layer, the service layer, and the presentation layer.

Integrating applications at the data layer can sometimes be the quickest implementation approach, due to the simplicity and power of data layer integration technologies. Data layer integration includes the use of database links, shared database catalogs, and direct database queries. However, this approach leaves many internal application details exposed, and upstream and downstream application changes result in significant impact, rework, and testing.

With application style integration, low-level implementation details may or may not be exposed or accessible from other systems. Application layer integration includes the use of flat files, message queues, and remote procedure calls. An "API" of some level is leveraged. Although this approach gives a better degree of separation, there are still problems with this approach.

Message-based technologies such as JMS and MQSeries are examples of popular queuing solutions but are based on proprietary implementations. Error handling can be problematic as messages can be lost on "undeliverable mail" queues. The level of application details that is needed to expose for interfacing applications is one of the major disadvantages of integration using remote procedure calls (RMI and RPC style of integration). In short, integrating applications at the application layer is preferable to data layer integration, but still has its own issues.

Service-based integration includes the use of Web services. The advantage of Web services integration is interoperability, even though the integration can still be point to point. Each Web service client still has the responsibility of knowing which service to call, so the addition of new endpoints will result in additional coding.

An enterprise service bus (ESB) combines the strengths of existing service-based integration technologies but provides more abstraction and interoperability. Application integration using an ESB combines message-oriented processing and Web services, which is the foundation for an event-driven SOA. In our opinion, services-based integration, especially when paired with an ESB, is the preferred method for all application integration in an enterprise.

NOW WHAT?

To get your application integration under control, we recommend that you work within your existing landscape of systems and technologies and leverage the TOGAF methodology, instead of kicking off a large "EAI project."

We use a "scenario-driven approach" that focuses on aligning business scenarios with supporting technologies. This approach consists of the following steps:

1. Leverage TOGAF architecture principles.
2. Identify technology standards and create building blocks.
3. Identify usage scenarios and map scenarios to the technology standards.

Leverage TOGAF Architecture Principles

The main reason to integrate applications is to support a business process. Taking a technology-first approach can lead to inflexible solutions that can be costlier to maintain as the business environment changes.

Architecture groups do not typically create technology standards in a vacuum for idealistic purposes (although some development teams may have differing opinions). When developing standards, you need to make sure there is direct traceability to enterprise architecture principles such as:

- Creating loosely coupled interfaces.
- Setting platform independent, open standards.
- Developing reusable, shared services.

MORE ON PAGE 8

- Minimizing application impact.
- Promoting data consistency.
- Recognizing that business logic is a strategic business asset and should not be placed in closed vendor solutions.

These principles are based on the default TOGAF enterprise architecture principles; however, they may be too vague for some project teams and leave much room for interpretation. To address this vagueness, the architecture group needs to identify acceptable integration technologies and map usage scenarios to those technologies to control the complexity of the integration environment.

Identify Technology Standards and Create Building Blocks

Based on organizational strategy, solutions in place, and staffing and skill levels, you should then identify the technology standards that are preferred at your company. Determine if your strategy is to be a custom Java shop, a Microsoft shop, or an SAP enterprise. Regardless of the technology, set clear standards for technology usage and avoid allowing developers to choose their favorite technology to use for integration.

TOGAF promotes the use of building blocks to support effective enterprise architecture. Building blocks are simply a package of functionality defined to meet business needs. For example, an architecture building block is a high-level, abstract architectural pattern. A solution building block is a specific instance of a technology or product. Every organization must decide for itself what arrangements of building blocks work best for it.

For each architecture building block, create the corresponding and solution building block(s). For example, ETL is an architecture building block; it can be realized by a solution building block consisting of products such as Informatica PowerCenter or Business Objects Data Integrator.

A well-specified catalog of building blocks will lead to improvements in application integration, facilitate interoperability, control technical diversity, and provide flexibility in the creation of new systems and applications. Table 1 lists a sample of the approved architecture and corresponding solution building blocks.

Table 1: List of Architecture and Solution Building Blocks

Architecture Building Block (ABB)	Solution Building Block(SBB)
ETL	Data Integrator, Informatica PowerCenter
Message Bus	Tibco, MQSeries
JMS	WebLogic JMS
JAX-WS	Metro
ESB	Websphere Message Broker

Map Scenarios to Standards

As a final step, map the integration technologies to various usage scenarios. You will need to develop usage scenarios that can be used to identify types of application interaction.

Our set of usage scenarios includes:

- Perform domain entity validation.
- Synchronize business information in multiple systems.
- Notify of state change between applications.
- Notify other systems of event of interest.

Notice how our usage scenarios are not technology focused—the scenario is not “put a message on an application queue”; the scenario is “Notify other systems of event of interest.”

We use a matrix to map the preferred integration standards to various usage scenarios at various layers, i.e. data, business logic, and presentation. The goal of this matrix is to enable consistent architecture between system interfaces by establishing a common set of prescribed integration and usage patterns.

For example, if a usage scenario is to synchronize information in multiple systems, the preferred integration standard would be to use message bus or ESB as suggested by the matrix. Any deviations from preferred integration techniques would require an exception process to be followed.

The preferred technology for a particular usage scenario is indicated on the matrix with a + sign; a – sign indicates that this technology is acceptable but not preferred. Also, some technologies such as FTP and DB links are not included, meaning that they are not permitted and their usage would require an exception approval.

MORE ON PAGE 9

The matrix acts as a guide to architecture and solution delivery teams. Basing decisions on this matrix minimizes the decision variability across teams and provides a base for service orientation that ensures interoperability and true integration in a heterogeneous enterprise landscape.

CONCLUSION

The key to successful services-based integration is to focus on why applications need to exchange information. Our scenario-driven approach enables the application teams to use the matrix as a guide to identify approved integration technologies based on business requirements. This approach helps to control technical diversity as well as fosters consistent integration standards across the enterprise. **A&G**

CARRIN TUNNEY is an enterprise architect at DTE Energy. With more than 16 years in application development and distributed systems, she is TOGAF certified as well as a Sun Certified Enterprise Architect.

SRINI SASTRY is a technical architect at DTE Energy. He has more than 15 years of application development experience and also holds TOGAF certification and Sun Certified Enterprise Architect certification.

Figure 1: Application Integration Standards Matrix

Application Integration Standards

		Business Layer Integration						Data Layer Integration					Presentation Layer Integration
		Web Services (sync)	Web Services (asynch)	Message Bus	Enterprise Service Bus	Messaging JMS	Messaging Websphere MQ	Operational Data Store/Data Warehouse	Database View	Websphere Process Server/WBI	Stored Procedure	ETL	Portal Integration
	Scenario												
Data Integration or Exchange	Read Master Data for reference	+						+	–		–		
	Query Business Data for reference while processing transaction	+						–	–		–		
	Read reference data for analytics or correlating other data sources							–					
	Extract or load large amount of data											+	
	Synchronize business information in multiple systems			+	+								
Perform Business Transaction	Perform Business Request—Query	+			+				–		–		
	Perform Business Function—Update	+	+		+						–		
	Perform Domain Entity Validation	+			+						–		
	Perform Business Function with External Service	+	+		+								
Event Notification	Notification of state change between Java applications		+	+	+	+							
	Notification of change between heterogenous systems		+	+	+		+			+			
	Transform in-flight information between applications that use different message structures				+					+			
	Notify other systems of event of interest		+	+	+	+	+			+			
	Distribute information & data generated by business events in real time to other applications				+					+			
Process Integration	Coordinate business processes that span multiple applications				+					+			
	Integrate systems at user interface												+

Entries with a + or – indicate standard interface techniques.

Gartner Portals, Content & Collaboration Summit 2010

15 – 16 September 2010 | London, UK | **europe.gartner.com/pcc**

KNOWLEDGE MANAGEMENT
Enterprise Content Management and Web Content Management
SharePoint **Social Networking**
Customer and Employee Portals

Evolution in Real Time

The world around you is moving in real time. **Are you?**

With the explosive arrival of the real-time Web and the unprecedented spread of social networking, new technologies and cloud delivery models are storming the workplace—and offering unlimited opportunity. The upcoming **Gartner Portals, Content & Collaboration Summit, 15 -16 September in London, UK,** will address these developments head-on.

Gain big-picture perspectives and practical how-to guidance for your most important decisions in 2010 and beyond.

No matter where you are in your learning curve, there's plenty of insight, fresh ideas and practical advice to help you **bring your workplace, information and process into real-time**.

Hot Topics

- Social Web and Gen4 collaboration
- Cloud-based portal and content services
- Using social networking tools to drive business value
- Business process improvement
- Collective intelligence
- Managing risk and reputation in social networks
- Web 2.0 and Enterprise 2.0
- The Cisco, Google, IBM, and Microsoft showdown in the cloud

Visit **europe.gartner.com/pcc** for more details…

Twitter: #GartnerPCC

Register now

Online: europe.gartner.com/pcc
Telephone: +44 20 88792430
E-mail: emea.registration@gartner.com

Gartner
Portals, Content &
Collaboration
Summit 2010

THE CASE FOR APPLICATION MODERNIZATION

A How-To Guide for Modernizing the Application Portfolio

By Larry Acklin

The goal of many businesses is simple: Have the business and IT departments work together to drive growth and innovation, at an affordable cost. Applications play a critical role, and modernizing them is the key to achieving long-term sustainable results. Luckily, by applying a comprehensive and strategic approach to application modernization, organizations can create an application environment that is efficient as well as cost effective for the business. The first step in the process is to assess all of the applications currently in the environment in order to effectively modernize the application portfolio.

THE MESS WE GET IN

Cost: Organizations are spending upwards of 70–80 percent of their technology budget on maintenance and keeping the lights on.[1] Whether these costs are infrastructure, application licenses, or support staff, they inhibit the flexibility as well as the ability to innovate and grow the business.

Complexity: Much of the code found in applications today is redundant, and it is common to see multiple instances of redundant applications. In many cases, the applications have been "patched" many times, becoming so bloated that changing them is a risk that threatens business continuity. In addition, the accumulation of applications through growth and acquisitions steadily increases the total number of applications in the portfolio and the level of complexity.

Security: As we move to a world that allows information to flow through many different devices and techniques, providing a security layer to match is difficult, especially if you have to change the applications to implement security. Tactical attempts to modernize applications have served as Band-aids that add to complexity, which only decreases overall security. New regulations on data security exacerbate the problems.

The combination of these issues makes it a struggle to keep applications aligned to the business. The good news is steps can be taken, which offer minimal risk and compelling IT and business benefits.

THE MESS CAN BE CLEANED UP, BUT . . .

What's ironic is that most organizations acknowledge that a formal application modernization program would benefit their organization. In fact, recent HP-commissioned research with more than 200 global organizations conducted by Forrester Consulting showed that 78 percent of respondents believe their organization would *greatly* benefit from an application rationalization effort. But in contrast, only 56 percent plan to do so. That's not the end of the contradictions.[2]

The study also addressed how applications *should be* modernized versus how they *will be* modernized. Eighteen percent of respondents believe that 21–40 percent

MORE ON PAGE 12

of their applications *should be* migrated to new technology platforms, but only 14 percent will do so. Forty-seven percent of respondents believe that they should *add new features but not modernize* up to 20 percent of their applications. Yet 44 percent will modernize. Forty-three percent of respondents believe that their applications *should be maintained but not modernized* up to 20 percent of the time. Yet 34 percent will modernize.[3]

Further, respondents were asked to rate their perception of modernization risk before and after employing various techniques. The percentage of respondents that perceived high/very high risk ranged from a low of 39 percent to a high of 50 percent—noting worry about requirements for external help, but most worried about the risk of cost overruns, business process impact, staffing, scheduling overruns, and technical issues. The good news is many organizations overestimated their exposure to risk. After modernization, 25–30 percent of survey respondents actually experienced *less* risk than they feared.[4]

The research further found that the top barriers preventing strategic modernization programs were cost, risk, business buy-in, and subject matter expertise. The study also asked respondents, "What would accelerate your plans to modernize/replace your applications?" The number one accelerator by a substantial margin is the ability to bring all parties to the table to reach agreement, followed in turn by increased staff/funding, the ability to make a proper business case, and the ability to lessen risk.

Why is that significant? It ties together many of the points—risks are actually less of an issue than they seem, modernization plans must be relevant to organizational plans, and cost is an ever-present issue. Still, enterprise leaders who do not understand our reasoning for modernization will not support it.

WHY MODERNIZE NOW

The most compelling reason for a formal application modernization approach is that other organizations, including competitors, are modernizing and gaining a competitive edge.

Secondly, there are many benefits that provide significant outcomes from modernization. Organizations see results that include:

- Better alignment of business and technology
- Increased agility
- Improved security
- Improved business continuity
- More balanced technology maintenance spend to 50 percent or less of the IT budget
- Improved understanding of how business processes are implemented in the applications

A COMPREHENSIVE APPROACH

Transforming the applications portfolio requires a comprehensive approach that treats each application appropriately based on its value to the business. The approach should also address the applications management processes to ensure that the applications portfolio will support the needs of the organization as they change and evolve over time.

Modernizing the applications portfolio itself can be achieved through assessment and a program road map designed to maximize benefits, accelerate ROI, minimize risk, all while identifying self-funding options. HP advocates an approach that consists of three phases: assess, modernize, and manage.

Assess: Assess the complete technology environment including applications, data, and infrastructure. Discover the assets available today through automation, inspections, and code samples. Analyze that information in comparison to business goals as well as priorities, and determine the strategic value of each application to the business. Look for redundancies in applications and identify opportunities for modernization. Finally, decide the best approach to reach the ideal future state and accompanying modernization road map.

Modernize: This phase includes realigning the applications portfolio so that it works for the business. It is designed to improve quality, security, information management, and performance. Multiple strategies are required to modernize an environment. Three basic approaches can help: keep, change, and retire. A modernization road map consisting of all three approaches will maximize benefits of the organization, while reducing costs.

- ❍ **Keep:** The focus is on making as little direct impact on the application as possible. We would use this approach for performing actions such as extracting business logic from existing applications or simply extracting knowledge for the purposes of documenting the

MORE ON PAGE 13

application. It could be used for rehosting applications to higher performance, lowering cost platforms, or wrapping existing applications as services to expose their functions to other systems. This approach is focused on cost savings.

- ❍ **Change:** This is where we will make fundamental changes in the structures of the applications so they meet the new demands of the business. Examples include replacing legacy applications with packaged ERP or industry applications. It could also be modernizing from legacy languages to new .Net or J2EE based languages. Where is makes sense, it could include going to SOA or cloud computing models. This approach is focused on increasing agility.
- ❍ **Retire:** For applications that perform duplicate business functions or no longer provide business value, this involves rationalizing the application's role and capturing and archiving data from the retired application. This process is critical to minimize disruption to existing users and for future business reporting and regulatory needs.

Manage: The third phase is all about gaining the greatest degree of visibility and control to maintain the optimized environment. By understanding the current investments in the application portfolio, IT can calibrate those investments for the greatest impact.

For companies that choose to not manage the application portfolio internally, one viable option to consider is outsourcing to an experienced service provider. Whether an organization chooses to outsource the management of the entire portfolio or only select applications, this phase will need to include the following:

- Clearly defined standardized service offerings based on ITIL framework
- Right-size support based on the criticality of each application by choosing from service package tiers
- Scalable, global delivery capabilities representing the highest industry standards backed by service level commitments
- Fixed, predictive spending levels that can adjust as the applications portfolio changes
- The ability to tune investments without renegotiation for quick and easy business alignment

HOW MODERNIZATION PAYS OFF

Each organization, including its process and benefits, is unique. By taking the steps to complete application modernization, businesses will be able to deliver impressive outcomes. Some of the specific benefits that applications modernization provides include:

- Increased responsiveness to business priorities and changing demands by eliminating the complexity inherent to legacy systems and server sprawl.
- Improved productivity for employees and reduced maintenance costs by streamlining complex, legacy systems to more modern, efficient technologies.
- Faster speed to market with modernized infrastructure and applications.

It is clear that organizations need to take action and truly understand the infrastructure in order to garner the business results that they desire. Applications modernization is one way for organizations to take control of the data center and uncover important information that would not have been made available without the process. Without applications modernization, organizations cannot effectively leverage their portfolios, which slows down innovation and business growth for companies all over the globe. **A&G**

LARRY ACKLIN is product marketing manager for HP Enterprise Services. He is responsible for global Application Modernization Services.

Notes

1. 2009 InformationWeek Analytics Survey of InformationWeek 500 executive.

2. Source: A commissioned study conducted by Forrester Consulting on behalf of HP, May 2010.

3. Ibid.

4. Ibid.

CIs and BBs: ITIL Meets TOGAF

By Eric Hendrickson

I have had some recent discussions about a tool that I produced which is meant to allow projects to be assessed based on their architectural reuse of a set of core building blocks (BB). This would allow architecture development within a project to demonstrate architecture building block (ABB) reuse through a BB-Requirements matrix, where each project requirement would be mapped to a standard set of core ABBs. The matrix would theoretically show that the more BB-Requirement intersection, the less risk of unnecessary and noncompliant components being introduced by that project's proposed architecture. This tool has been sent out to a few people, and, interestingly, one of the first questions that has come back has been related to IT management. Essentially, the question was this:

What is the difference between a TOGAF building block and an ITIL CMDB configuration item?

Since we are in the process of implementing an ITIL-compliant ITSM system, there is increasing interest in topics such as CMDB and configuration items (CIs), but the more I consider the question, the more I am drawn into this topic: Truly, what is the difference? Or, are there any differences at all? Let's look at some definitions:

- **A Configuration Management Database (CMDB),** according to ITIL v3, is a database used to store configuration records throughout their life cycle. The configuration management system maintains one or more CMDBs, and each CMDB stores attributes of CIs and relationships with other CIs.
- **A Configuration Item (CI),** according to ITIL v3, is any component that needs to be managed in order to deliver an IT service. Information about each CI is recorded in a configuration record within the configuration management system and is maintained through its life cycle by configuration management. CIs are under the control of change management. CIs typically include IT services, hardware, software buildings, people, and formal documentation such as process documentation and SLAs.
- **A Building Block (BB),** according to TOGAF v9, represents a (potentially reusable) component of business, IT, or architectural capability that can be combined with other building blocks to deliver architectures and solutions. Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached. For instance, at an early stage, a building block can simply consist of a name or an outline description. Later on, a building block may be decomposed into multiple supporting building blocks and may be accompanied by a full specification. Building blocks can relate to "architectures" or "solutions."

Without claiming to be an expert in ITIL or TOGAF, I have attempted to outline similarities and differences that I believe exist at first glance. There are definitely several ways to look at the nature of the data and metadata that populate an ITIL CMDB or a TOGAF meta model-based architecture repository, but since I'm still learning to apply both knowledge areas, I'll leave my assessment at only the surface level.

SIMILARITIES

- **Components:** CIs and BBs are both discrete components—hardware, software, locations, roles, services, etc.—each with a unique set of attributes.
- **Relationships:** Both are expressed not only in terms of their own attributes, but are most valuable when relationally modeled in respect to other components.
- **Abstractions:** Both make use of abstraction, composition, and decomposition to express "low level" components and their relationship to "high level" components.
- **States:** Some configuration management systems (CMS) are able to manage transitional states between the current state and previous transitions or even proposed future states. This is similar in concept to the transitioning of a BB from a current

MORE ON PAGE 15

to future state. Though, the implementation of this is dependent on the ITSM and CMS, CIs definitely support the idea of states.

DIFFERENCES

Single vs. Multiple Perspectives

One of the biggest differences between a CI and a BB is the framework that manages them. Since ITIL is primarily a service management framework, CIs are typically represented in a service model. Because of this, I would imagine that there is almost never a CMDB dedicated to managing the relationships between organizational goals and a process, nor is data typically modeled in a CMDB at all. Therefore, I believe you could consider ITIL's service models to be a single service management view of the broader set of views required to model EA. This means that only the CIs required to model this view are housed in the CMDB.

Operational vs. Strategic Functions

Since the CMDB typically only manages CIs related to service management, it is particularly helpful to those performing day-to-day service management activities. Consider a strategy map dashboard that shows strategic goals and their relationship to one another, and rolls up health information for each goal. This would be another operational view, supported by EA modeling, which would not fit into a typical CMDB and therefore is not a candidate CI.

Service Management vs. Enterprise Architecture Context

In summary, I think the biggest difference between CIs and BBs are their context. The systems that attempt to support this context do not take into account other uses. While the CIs in a CMDB are relegated to only those required to model the service management view, this does not have to be the case. Nor is it true that there shouldn't be some collaboration between CMDB and EA repository vendors to support a dual purpose system, where BBs are able to be made into CIs in support of the IT service management view. **A&G**

ERIC HENDRICKSON is an enterprise architect at Liberty University. He can be reached at enhendrickson@liberty.edu. His blog is at http://eametamind.info.

A&G CALENDAR

IASA World Summit 2010
September 23–24, 2010
New York City
www.iasahome.org/web/itarc/nyc2010

EAMKON 2010
October 5–6, 2010
Stuttgart, Germany
www.eamkon.de/

Troux Directions 360
October 6, 2010
London
www.troux.com/tdemea

Forrester's CIO Forum 2010
Business Innovation: Fueled By IT
October 6–7, 2010
Gaylord National Resort & Convention Center
Washington, D.C.
www.forrester.com/events/eventdetail?eventID=2483

MDM & Data Governance Summit
October 13–14, 2010
Furama City Centre, Singapore
www.enterpriseiq.com.au

Gartner Symposium/ITxpo 2010
October 17–21, 2010
Orlando
www.gartner.com/technology/symposium/orlando/index.jsp

Data Warehousing & Business Intelligence 2.0 Summit
October 26–28, 2010
Swissotel, Sydney Australia
www.enterpriseiq.com.au

Data Management, Information Quality & DW/BI Conference Europe 2010
November 3–5, 2010
London
www.irmuk.co.uk

Enterprise Architecture Management 2010
November 23–24, 2010
Hilton Berlin, Germany
www.marcusevansde.com/EAM-Europe2010

Cutter Consortium Summit Europe 2010 in association with IRM UK
December 1–3 2010
London
www.irmuk.co.uk

Submit your Calendar Events to
editor@ArchitectureandGovernance.com