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

JOIN THE COMMUNITY

# ARCHITECTURE & GOVERNANCE

magazine

VOLUME 10 ISSUE 1

## MAKING THE DIGITAL TRANSFORMATION

Part 1: How to Make Enterprise Architecture Strategic?

Lock-In: The Cloud's Hidden Menace

Engineering Why: What Every Architect Needs to Know

Best of Blog: Emerging Technology and Disruptive Technology — What's the Difference?

ARCHITECTURE &GOVERNANCE
magazine

**PUBLISHER**
Ben Geller & Ashlee Motola

**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







# A Big Deal

## BY GEORGE S. PARAS

Digital transformation is the latest big deal. While there have been many "next big things" over the last two decades, this one is different. It has the potential to fundamentally change things for IT and business alike. Made up largely of what has come to be known as SMAC (Social, Mobile, Analytics, Cloud), this transformation is less about the underlying technology and more about changes in the way we work and interact.

Digital transformation can touch every corner of a business and the IT environment. It's a lot to invest in all at once. An incremental approach makes the most sense, but with business demand and a high sense of urgency at play, there often isn't time to be strategic. It doesn't take long before project silos result in a new portfolio of incompatible processes, information, solutions, and infrastructure. And that is on top of the less-than-optimal portfolio many already had. Having a vision for the end game, an appreciation for the breadth of change, and insight on how much to do and when to do it provides coordination and sanity to the journey. That's a role for enterprise architecture.

In this A&G, our contributing authors offer a few of their perspectives on the issues surrounding transformation. First up, Jan Gravesen emphasizes the need to make enterprise architecture strategic. He looks at strategy processes through the years and describes how they each drive differing forms of EA. Ron Ross reveals his imaginary "why button," a way to think about how business rules help us make sense of day-to-day business operations.

Karen Tegan Padir looks at cloud computing from the business consumer perspective and suggests that businesses must take care to understand and address lock-in. And finally, Chris Curran, our guest blogger this issue, offers his thoughts on the context for evaluating emerging and disruptive technology.

We'd also like to encourage our readers curious about digital transformation to attend Forrester's "Unleash Your Digital Business" event. Details are inside. Thanks again for being an A&G reader! **A&G**

**GEORGE S. PARAS** is editor-in-chief of A&G and managing director of EAdirections.

**THIS ISSUE**


PART 1

# How to Make ENTERPRISE ARCHITECTURE STRATEGIC?

By Jan K. Gravesen, CTA, Executive Architect, IBM

**ABSTRACT**

*Creating impactful and strategically aligned enterprise architecture is an increasingly complex task. Information technology is opening up a historically wide and growing range of options for investing in technology to underpin strategic business goals and strategy continues to evolve. This article provides a framework that explains how enterprise architecture has evolved to be able to contribute to varying and evolving strategy views and how enterprise architecture can be described in terms that are more aligned to the set of strategies selected by the firm or organization.*

Although the concept of strategy in warfare can be traced back to ancient times, its application in business is surprisingly recent. Multiple theories and definitions regarding business strategy has emerged since the 1960s, but to this day little consensus exists on the meaning or importance of strategy in a business context.

MORE ON PAGE 4

Business strategy, technology strategy, and enterprise architecture are highly amorphous fields. Consequently, there is a need for narrowing the subject. One can distinguish between at least two approaches to enterprise architecture:

A: A design school which tries to influence the structure of the organization and the design of its capabilities with special emphasis on its information-based capabilities in order to better support strategic behaviors that align with its objectives and its competitive and political environment.

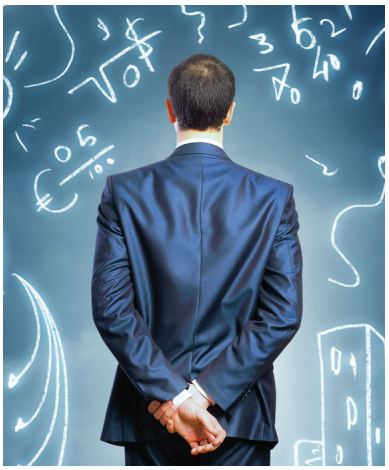
B: A learning school which redesigns business processes and the firm's structures as a series of smaller steps where each step is an adjustment to an overall guidance or strategic path as well as a response to changes that occur in the external environment and where each step builds on the results from earlier steps.

While early approaches to strategy were structured and design-oriented, newer forms of strategy are more dynamic and learning-oriented, and they force enterprise architecture to be approached differently to be effective.

As strategy has become an increasingly eclectic field of practice, most large and complex organizations today apply several different approaches to strategies: traditional diversification and partitioning may be applied at a corporate level, dynamic capabilities may be central to how the organization's web channel operates, and certain parts of its product strategy may adhere to strategy as ecology. Consequently, enterprise architecture must be able to reflect and respond to not just one but typically a set of strategies before it is possible to claim that it operates on a strategic level.

This two-part article explores how enterprise architecture could or should look under a set of different

MORE ON PAGE 5



Strategy as a deliberate managerial discipline dates back to the 1950s where Harvard professors developed the design-based approach to business strategy, which remains highly influential today. Successful strategies are those that achieve a two-way strategic fit between the environmental opportunities and threats that exist and the resources possessed by the firm.

strategies as they have emerged from the 1950s up to current times. In doing so, it provides a guide to how to express the benefit of enterprise architecture and to recognize the different forms enterprise architecture often must take to respond to the full set of strategies selected by an organization.

The article will be organized by a set of strategies starting with strategy as structure and design in the 1950s and up to more contemporary forms such as dynamic capabilities and strategy as ecology in the 2000s. In this first part, the emphasis is on classical forms of strategy from the 1950s up to and including the 1990s, while the second part considers the modern forms of strategy from the 1990s and up to current time. The second part concludes with an overview of a set of common themes that enterprise architecture seems to address irrespective of the specific strategies selected by the firm.

## STRUCTURE AND DESIGN (1950–1960)

Strategy as a deliberate managerial discipline dates back to the 1950s where Harvard professors developed the design-based approach to business strategy, which remains highly influential today. Strategic choices are based on an interdependent analysis of the external environment of the firm and the firm's internal distinctive assets, practices, processes, and resources. Successful strategies are those that achieve a two-way strategic fit between the environmental opportunities and threats that exist and the resources possessed by the firm.

The general idea of reflecting external and internal factors in the strategy is distilled in the popular SWOT framework, which continues to be used by firms and consultants alike.

Igor Ansoff specified two basic strategic questions for driving the strategic agenda of the organization, namely:

1. "Which business are we in and what business should we be in?"—in essence, what is the purpose of the organization?
2. "How should we compete in these businesses?"—i.e., the choice of appropriate strategic agendas in the chosen business areas.

Ansoff proposed to compare the different growth trajectories of the organization based on options for developing new products and serving new markets. The "Ansoff matrix" with its four product-market strategies (market penetration, product development, market development, and diversification) describes different strategic options that must be fitted to the different stages of the organization's life cycle.

It was especially the early management consulting firms such as the Boston Consulting Group that began to analytically address competitive advantage with the concept of the "experience curve." The rationale for the experience curve was that for cumulative doubling of experience (e.g., calculated as units produced and sold) the average total unit cost would fall 20–30 percent. The reduction in unit costs could be attributed to organizational learning and technological improvements in the manufacturing and distribution apparatus, and

MORE ON PAGE 6

economies of scale.

The strategic role of technology scale benefits and organizational (competence) learning was thus established as important levers for seeking competitive advantages and became linked to economical rational theory.

The environmental conditions for strategy were characterized by: (a) low organizational, product, and process clock speeds; (b) low information density, which only allowed products to serve mass markets; (c) functional, hierarchical, and stable organizational forms; and (d) a strategic thinking that focused on cost advantages and internal fits with relatively stable external conditions.

Enterprise architecture focused on building and sustaining a production apparatus (process) for serving mass markets and on designing the organizational structure and its capabilities so it maximized scale benefits (experience curve). Technology became an integral part of the manufacturing process, and its key contributions were toward process automation, waste reduction, and manufacturing asset utilization.

**EXAMPLE:** Enterprise Resource Planning (ERP) systems are a significant example of this thinking. Another example is often referred to as global single instance architecture (GSIA) where global applications serve an organization's functions on a global scale. Strategic enterprise architecture will often focus on the analysis and modelling of business activities across organizational units that is required to implement GSIAs or global ERP packages.

Many private and public sector organizations still maintain a diversified and decentralized structure, and today enterprise architecture is concerned with creating cohesion and collaboration among the different business units that often operate with different IT governance and investment schemes. The main objective is to improve scale benefits across the business units by fostering better resource and knowledge sharing.

### DIVERSIFICATION AND PARTITIONING (1960–1970S)

During the 1960s and 1970s, organizations were diversifying away from their original core businesses. The Boston Consulting Group's growth-share matrix provided an analytical framework for a balanced portfolio view of the firm. With time, the portfolio view led to the strategic specialization of firms into multi-business organizational forms consisting of strategic business areas being served by strategic business units.

Diversification and decentralization led to a situation where strategic business units, rather than the corporation, became semiautonomous profit centers and the units of strategic planning. Strategic planning increasingly became focused on the resources and capabilities of the individual business units, and enterprise architecture became a matter of building and sustaining manufacturing technology with a strategic fit to the business areas in which the units competed.

In diversified and decentralized organizations, enterprise architecture fluctuates between a central and a decentralized orientation depending on whether the organization sustains high growth or low growth/economic subtraction. In high growth situations, enterprise architecture becomes decidedly decentralized and focused on the business units, while low growth situations called for scale benefits and reduced technology (process) unit costs and therefore for technology asset efficiency and sharing of core product and process assets across the strategic business areas.

MORE ON PAGE 7

Many private and public sector organizations still maintain a diversified and decentralized structure, and today enterprise architecture is concerned with creating cohesion and collaboration among the different business units that often operate with different IT governance and investment schemes. The main objective is to improve scale benefits across the business units by fostering better resource and knowledge sharing. Enterprise architecture is about how to organize, govern, and utilize common process and product platforms while still allowing business units or departments to maintain some autonomy.

**EXAMPLE:** In diversification and partitioning, enterprise architecture often focuses on creating "globalized" capabilities—i.e., global information-based capabilities or platforms that can be modified or enhanced by local units to meet their local objectives.

An example is online collaboration centers for upstream oil and gas. Such cross-disciplinary problem-solving centers are becoming commonplace in upstream oil exploration and bring together petrochemical engineers, geophysicists, drilling engineers, and geologists in a cross-disciplinary process supported by online information-sharing technologies that operate on a global scale. The global scale is important because upstream oil and gas increasingly diversifies its operations across tar sands, fracking, deep sea drilling, and onshore drilling and must draw on the collective experience of scientists that operate from widely different locations in the world. The local scope is important because the online information-sharing platform must be able to adapt to local specific problems.

## COMPETITIVE POSITIONING (1980–1990)

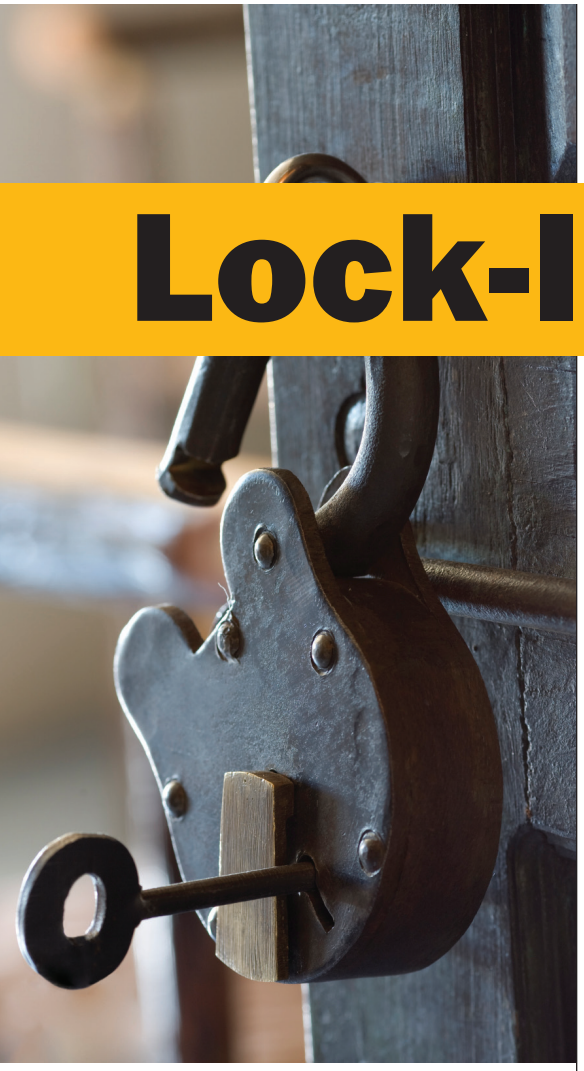
In the perspective of industrial economics and competitive positioning, which was most clearly described by Michael Porter, strategy is about comparing the attractiveness of different markets and then selecting strategies in each of these chosen markets, which would create defendable positions for the firm's strategic business units.

Two important characteristics that lend themselves well to enterprise architecture are the value chain and the ability to share resources across business units. The value chain is an analytical framework that allows the enterprise architect to focus on core activities and strengthen these through automation, by linking them up to create straight pass-through capabilities where data can be passed between process steps without human intervention, and where new information-based products can be configured within the activities offered by the value chain.

The ability to share resources is considered a way to make the organization more resource efficient. Especially in terms of knowledge and information-based business models, information sharing can be enhanced and automated by information technology. Supply chain management is an example of a process that cuts across both the value chain and the ability to share resources among business units, even extending beyond the boundaries of the organization to its suppliers, partners, and customers.

**EXAMPLE:** Two general examples of enterprise architecture are business process re-engineering (BPR)—an attempt to optimize the firm's value chain and create defensible industry positions through differentiation in how activities were performed—and the creation of common information-sharing platforms that allowed business units to collaborate (for instance, on R&D activities) and join up their respective value chains. **A&G**

**JAN GRAVESEN** is an IBM executive architect specializing in enterprise architecture and public sector transformation and strategy.



# Lock-In

# The Cloud's Hidden Menace

By Karen Tegan Padir, CTO, Progress Software

So much is so great about the cloud that anyone who talks or writes about it is in danger of sounding like they are in the midst of some kind of religious rapture. Naysayers, on the other hand, are few and far between. The truth is that the cloud—by which most people mean the public cloud embodied in services provided by the likes of Amazon, Rackspace, GoGrid, and HP—is a tremendous opportunity that can benefit most kinds of organizations. It offers instant access and agility, infinite capacity and scalability, moderate costs, and the ability to "switch it on and switch it off."

Still, anything that comes close to being "too-good-to-be-true" has to have some downsides. For example, it is important to remember that cloud providers have no incentive to help you keep your costs low. Switching on instances is easy, but it is most often manual and sometimes complicated to switch them off. Not only do you have to remember to stop machine instances that are not being used, but you also have to manually remove unnecessary snapshots and release unused elastic IP addresses among other things. Who knows how many rogue instances are out there in the cloud, conceived for a specific project and then more or less abandoned, but still steadily incurring charges?

That's a problem in and of itself, but the even bigger downside of moving to the cloud is lock-in. Yes, lock-in—just like with the old proprietary operating systems offered by IBM, Novell, or Digital Equipment Corporation. Just like the "old days," once you are seduced onto the platform of any given cloud provider, your days of freedom are numbered. You may get lots of benefits and you may even love the arrangement, but should you ever decide to go elsewhere or bring your data and applications "back home" to your own premises, it could be problematic.

That's lock-in.

It's not the kind of problem you'd initially expect to find in the cloud—the paragon of a more flexible and even democratic IT space. Lock-in is a particularly serious problem when applications are developed within and for a particular public cloud environment. Unless you deliberately avoided using vendor-specific features, the chances of moving your application elsewhere become practically nil.

It's not that there isn't a role for the cloud. The cloud is going to be central to more and more IT activities. However, as you build an architecture for your organization, it is critical to avoid one-way streets that can lead to dead ends. Although the public cloud often integrates standards, each cloud Infrastructure-as-a-Service (IaaS) vendor brings specific elements to the table, such as ease of use features that end up becoming layers of lock-in. Many of these features are common to multiple IaaS providers' offerings, but the implementation details are platform specific and nonportable. And that's where the problems lie.

Cloud vendor lock-in isn't always quite as blatant or complete as, say, building an enterprise around Digital Equipment Corporation's VMS was in

MORE ON PAGE 9

the minicomputer era. Instead, the cloud usually acts as a giant sponge, happily absorbing whatever workloads you choose to offload. But "squeezing the sponge" to get your applications or data back from a given vendor isn't so easy because the cost and complexity of migrating off of the cloud providers are so high. You need to be conscious and deliberate in the choices you make about the services you consume from your cloud provider.

One option to help avoid lock-in is to choose a platform stack that can be deployed anywhere: on premise, private or public cloud. This provides better control and tooling at every stage of a move to the cloud and also reduces the leverage that a cloud provider has over you. You can also avoid cloud lock-in through adoption of Platform-as-a-Service (PaaS) technology. PaaS facilitates the deployment of applications without incurring the cost and complexity of buying and managing hardware and software or of endless provisioning cycles. It is today's version of middleware all bundled together and can be a critical "next step" for organizations as they move from on-premise to cloud. Many PaaS providers offer a private cloud option, which serves as an insulating layer that prevents cloud-vendor lock-in.

In the PaaS model, you can build functionality using the tools or libraries that are built in to the PaaS, as well as whatever is familiar from your own environment. And, critically, PaaS gives you the tools to control configuration and manage deployment, obviating dependence on most of the cloud-vendor specifics that lead to lock in.

There are many different PaaS vendors offering variations on this theme. Selecting from among them is beyond the scope of this article. However, if you are adopting PaaS to avoid cloud-vendor lock-in, you should make sure that your PaaS investment includes the ability to create, connect, and integrate applications and data and do so across a wide range of environments (another key way to prevent lock-in). In addition, PaaS should provide tools that are easy to use so that you are as productive as possible.

So, in considering a move to the cloud, make sure you are moving toward best practices and building or maintaining an architecture that is flexible. Consider these "rules of thumb":

1. Make sure you have controls in place so that you don't consume more cloud resources than you need, especially if it is just because "someone forgot" to throttle back a cloud activity.
2. Cloud is often but not always less expensive than on-premises. Be sure to have metrics in place and consider how you are evolving a given function. It might turn out that building your own capacity on premises or in a private cloud actually makes more sense financially.
3. Look for approaches that are modular in nature and allow you to mix and match functionality to meet your needs, while only paying for what you use.
4. Look at open source options. Open source can reduce your price of admission to mission-critical infrastructure. Open source offers direct access to the source code and those who wrote it—which can be a positive if you have sufficient in-house capability to make use of this information.
5. Look at PaaS as a potential stepping stone that can help you chart a path to hybrid and cloud deployments.

Then, as you look to the cloud to add capacity, cut costs, streamline dev/test, or add new functionality, be sure to keep the blinders off.

As a famous writer once warned, "Those who cannot remember the past are condemned to repeat it." We are on the cusp of a new, cloud-centric age in IT, but we should recall how unpleasant vendor lock-in was in the old proprietary era. The cloud should be your ally in moving to the future, not your enemy. **A&G**

**KAREN TEGAN PADIR** is chief technology officer of Progress Software, where she is responsible for defining and delivering the company's technology strategy, innovation, and vision.

# ENGINEERING WHY

## What Every Architect Needs to Know

By Ronald G. Ross, Principal,
Business Rule Solutions, LLC

Have you ever been confused about why you were not allowed to do what you tried to do? Been judged or evaluated in a way you didn't expect? Stumped by the result or decision a business system produced?

If so you are a *why victim*. In today's business world, all of us are why victims more and more often. The remedy is business rules. Not technical rules masquerading as business rules, but real business rules expressed of, by, and for the business represented purely in business terms.

Business rules are part of the broader solution: Why Engineering. This article introduces *Why Engineering* and its basic principles, along with the *Why Button*. Find out what it takes to be effective at Why Engineering in today's business environment.

### THE WHY BUTTON

Business rules are all about answering the question *why?*. Why things are disallowed. Why specific judgments or evaluations are made. Why certain decisions are reached.

Imagine you had a *Why Button* handy whenever you encountered some disconnect in day-to-day business operations. Hit the Why Button and presto—answers appear in the form of relevant business rules.

Not technical rules but rules of the business—statements of guidance you can read and understand no matter what your role—business manager, business product developer, operational worker, business analyst, or IT professional. A single representation accessible to all audiences that is:

- Precise enough to remove all ambiguity.
- Detailed enough to produce the same results no matter whether applied by workers "manually" or automated by machines.

What would that do for your business? For one thing it would keep know-how from walking out the door—i.e., make your business logic explicit, not tacit, so you can retain it. For another, it would eliminate semantic silos—people using the same words but not really

MORE ON PAGE 11

communicating. It would also go a long way in closing the gap between business and IT. Overall, it would mean stepping up to do business intelligently in the knowledge economy.

## THE WHO OF WHY ENGINEERING

To achieve this vision you need a *Why Engineer*™.

A Why Engineer is not a knowledge engineer in the sense of expert systems and not a technical wizard in ontologies. Rather, a Why Engineer is someone who uses rigorous discipline to capture, represent, and communicate business rules based on carefully engineered business vocabulary. A Why Engineer is an architect who:

- Takes great care—and pride—in how things are expressed and defined.
- Can probe deeply into the "why" of business logic never once using a term or structure whose origin lies in IT or system design.
- Believes basic operational know-how has huge value and therefore should be managed and leveraged in every way possible.

## THE HOW OF WHY ENGINEERING

What can a Why Engineer offer your requirements process? Business rules provide the "why"—the basic rationale—for business requirements and elements of system design. They provide a solid basis for motivating each part of the solution you envision.

As an example, suppose you are creating functional requirements for a truck-routing problem. Let's say you arrive at the four requirements illustrated in figure 1.

The obvious question is: What ties the requirements together? What's the underlying business rationale?

If you had started from a business rule, the business rationale would be straightforward—little or no further explanation needed. The focus shifts from the requirements to the business rule: *Is the rule right for the business?* Figure 2 illustrates the possible starting-point business rule (expressed in RuleSpeak®[1]) for the requirements in figure 1.

Today's system-driven approaches result in a lot of arm waving about the motivation for business requirements.

**Figure 1. Four functional requirements for a truck-routing problem**

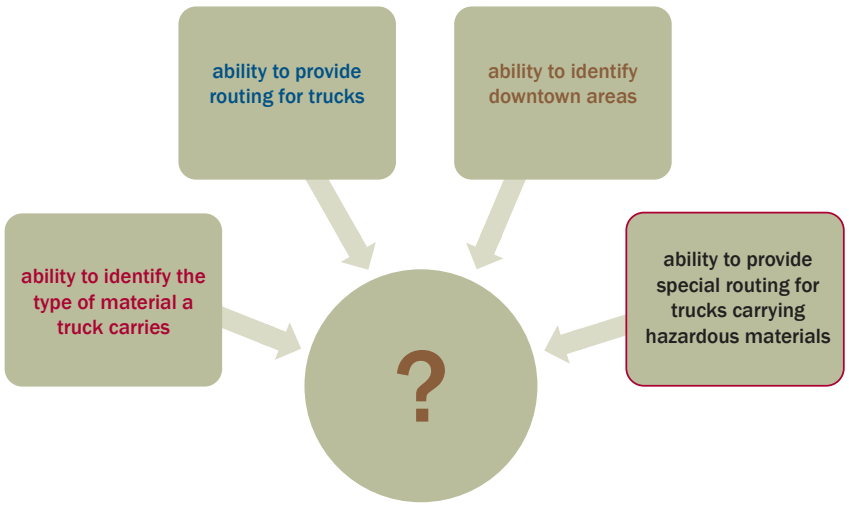
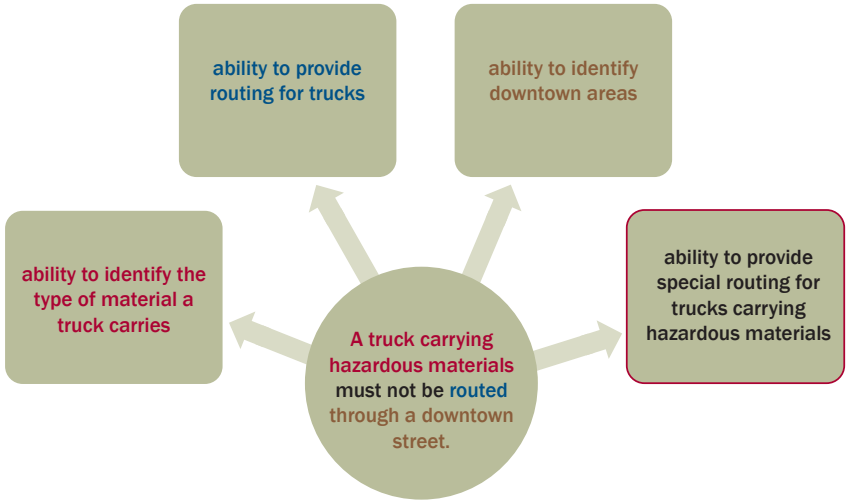



**Figure 2. Possible starting-point business rule for the four functional requirements**




1. RuleSpeak is a set of guidelines for expressing business rules in structured natural language. The guidelines are free on www.RuleSpeak.com.

MORE ON PAGE 12

A great many pages of generally formless documentation are often produced no one really reads.

All that is far from harmless noise. It detracts from delineating:

- The core business policies needed to actualize the business strategy.
- The elemental know-how that differentiates your product/service and provides the basis for achieving excellence in its delivery.

How have methodologies strayed so far from the very know-how that keeps you in business?! The Why Engineer puts things back on track.

## THE WHAT OF WHY ENGINEERING

*Why Engineering* is based on three fundamental architectural principles:

**PRINCIPLE 1.** *The same complete, intelligible, unambiguous, deployable meanings (business rules and definitions) should be presented to all key audiences in the business—managers, business product developers, operations, business analysts, and IT professionals.*

Looking back to the truck-routing example, none of these audiences is likely to have much trouble understanding the structured rule statement: *A truck carrying hazardous material must not be routed through a downtown street*. This example is a relatively simple one; reality, of course, is often more complex. Let me return to that point momentarily.

**PRINCIPLE 2.** *A Why Button should be part of every architecture and business solution.*

Consider the following scenario for the truck-routing problem. The local manager in a large city needs a load of automotive parts picked up on the docks and sent rush-delivery to a parts dealer across town. He assigns a driver to the job, tells her to get it there fast, then goes about his other tasks. Meanwhile, the driver requests optimal routing for the shipment and is surprised by the result, a route by no means the shortest or fastest. The driver is tempted to take a much more direct route right through town—after all, the manager said to get the load there fast—but first she hits the Why Button. The response she sees:

- *A truck carrying hazardous material must not be routed through a downtown street.*
- *This shipment includes air bag modules.*
- *Air bag modules are hazardous material.*[2]

**PRINCIPLE 3.** *The same form of "why" answers (business rules) created originally should be reused and provided to each audience*[3] *in identical form whenever the Why Button is hit.*

In the scenario above, the same business rule statement originally harvested before designing the system plays a direct role during its subsequent operation. And why not? It's a simply a *business* rule—a rule for running the business. That's its purpose; that's the role it should play—to inform and shape everyday behavior at the operational level. Every game has a rulebook for reference; putting the rules directly to work in automated systems is simply good architecture practice.

## DEALING WITH COMPLEXITY AT SCALE

The truck-routing business rule is a relatively easy one to comprehend. Reality is generally more complex for at least three reasons.

1. Many (perhaps most) business rules are more highly nuanced (qualified). That's one reason for following careful guidelines in expressing them such as offered by RuleSpeak. Through extensive real-world experience, best practices have also been developed for handling lists[4] and decision tables[5].
2. The vocabulary for the product/services of many organizations is more abstruse. Solid business definitions of terms are essential. A reasonable comparison in that regard is to the legal profession. In one 22-page contract I reviewed recently, I found five full pages of definitions. That's 23 percent of the total content!

2. http://phmsa.dot.gov/staticfiles/PHMSA/DownloadableFiles/Files/Hazmat/Alpha_Hazmat_Table.pdf

3. Obviously, answers should be available only to those duly authorized—but authorizations are simply more business rules.

4. *Tabulation of Lists in RuleSpeak®*: A Primer—Using "The Following" Clause, free download on http://www.brsolutions.com/b_ipspeakprimers.php

5. *Decision Tables—A Primer: How to Use TableSpeak™*, free download on http://www.brsolutions.com/b_ipspeakprimers.php

MORE ON PAGE 13

3. The devil is in the details—organizations often have thousands or tens of thousands of business rules. So a glossary of business definitions is not enough. You need a specific kind of architecture—a blueprint—to organize the underlying concepts. That's the only way coherency in significant numbers of rules can be achieved. Such blueprints for business semantics come in the form of a *concept model* based on structured business vocabulary. We offer ConceptSpeak™ to guide professionals in developing this kind of structure.[6]

## LEARNING AND APPLYING WHY ENGINEERING

Why Engineering really has nothing to do with IT directly. It can be (and has been) used even where no automated system is being built. It's a very pure form of business architecture.

In a sense, Why Engineering is simply about highly precise business communication. Is that a skill every architect or IT professional possesses naturally? Unfortunately no—not even close. It must be learned.

Fortunately, effective techniques for Why Engineering are available and have been proven in practice. They consist of structured natural language tools such as BRS ConceptSpeak™, RuleSpeak®, and TableSpeak™. These notations—really *thinking tools*—are based on a rich standard, SBVR (*Semantics of Business Vocabulary and Business Rules*)[7], developed over many years by world-class experts in formal logic, linguistics, and software engineering. SBVR itself is based on ISO terminology standards.

Is Why Engineering hard? Yes and no. The organizing principles and thinking tools can be readily learned. As for any engineering discipline, however, there's a definite learning curve. It takes diligence and practice to become really good at it.

Actually, the hardest part of Why Engineering is getting at the buried assumptions and know-how in people's heads—or lost in the jumble of legacy systems. The thinking tools of Why Engineering simply offer professionals the essential means for discovery, representation, and validation.

6. *Business Rule Concepts: Getting to the Point of Knowledge* (4th edition), by Ronald G. Ross, 2013.

7. Refer to the *SBVR Insider* section on www.BRCommunity.com.

The ultimate prize—common understanding in explicit form—is something the Why Engineer must still work hard to achieve. If it were easy, everyone would already be doing it!

## SUMMARY

Why Engineering is engineering in the fullest sense of the word. All engineering strives to produce something useful for people or their organizations. In Why Engineering that product is business rules, explicit business logic.

In general, engineering requires two things: *source material* and *structural principles*. For Why Engineering:

- The source material is literally *words*—or more accurately the concepts and meanings behind the words.
- The structural principles indicate how business logic can be represented in an unambiguous, anomaly-free form that is free of any IT or system-design artifacts or bias.

Why Engineering is engineering at its best.

As in all good engineering, the product of Why Engineering is *highly* reusable. What you develop as business logic is directly reusable as the answers produced by the Why Button in operation. Nothing more is required. It is *exactly* the same stuff—unified and reused with pinpoint accuracy.

Good engineering is also always concerned with sustainability of the product. Point-in-time ("band-aid") solutions are avoided. In Why Engineering, sustainability can be achieved by business-level rulebook management,[8] something we believe should also be part of every business architecture. **A&G**

8. "What Rulebooks, Rulebook Management and GRBS Are About," http://goo.gl/iBwsrE

**RONALD G. ROSS is cofounder and principal of Business Rule Solutions, LLC. Twitter: Ronald_G_Ross. For more info: http://www.brsolutions.com/**

# Emerging Technology and Disruptive Technology: What's the Difference?

**BY CHRIS CURRAN**

*This post is syndicated from CIO Dashboard (Original Post)*

Some people question my use of the word "emerging" to categorize technologies such as social, mobile, cloud, analytics, sensors and others. They argue that these technologies are already here and aren't emerging anymore. Some suggest using "disruptive" to describe them instead. Now is a good time to distinguish between these two labels and make sure that we are using each to help us better understand technology's impact on our businesses.

Emerging means to come into view. New technologies, like those represented in the latest SMAC craze, crash into a painfully cramped marketplace virtually every day. Reaching for a shiny object in the distance isn't necessarily wise. Pouncing on an emerging technology that fizzles out is costly and can distract CIOs from focusing on maintaining reliable systems based on stable technologies that drive business goals.

In contrast, the word disruption can have two meanings: 1) to throw into turmoil or disorder; 2) to break or split apart. Business leaders who don't respond to the first fast enough can end up facing the second. Think about the behemoths that have been brought to their knees after they didn't get ahead of consumers shifting their buying behavior as a result of advances in technology.

The important difference in these terms is the context to which they apply when thinking about a new technology. For example, smartphone apps might have already emerged but could still have untapped disruptive potential in impacting business models in the personal healthcare domain. Sensors may already be pervasive in our phones, cars, appliances, and sports wearables but still be untapped in the disruptions they could offer in the insurance industry.

Emerging technologies might lead to disrupting certain aspects of an industry, business model, or customer segment while not impacting others. Disruptions associated with an emerging (or emerged) technology might be years away or weeks. Not every company needs to adopt the latest suite of technologies with the same sense of urgency.

We surveyed more than 1,100 business and technology executives and found that just under 50 percent are investing in mobile, social, cloud, and big data. Are these companies falling behind or being smart and not chasing after technologies that don't support their business goals? How do CIOs manage this delicate dance along the emerging/disruption spectrum? Not investing too early and not too late? How did Goldilocks know which porridge, chair, and bed was best? She was dedicated to the process of exploration.

In the same Digital IQ Survey, we asked companies how they explore and act on information technology innovations. Do they maintain dedicated innovation teams or throw a team together on the fly? Do they hire a third party or partner with a university? Why does it matter?

We parsed the answers of top performers, 50 percent of whom maintain a dedicated innovation team compared to 38 percent of the pack. Others assemble ad-hoc teams on the fly or hire third-party vendors.

Enlisting a dedicated team to evaluate emerging technology is becoming a must-have, given the velocity, volume, and intensity of emerging technology. You need to consistently evaluate the parade of options and keep your discerning skills sharp. A commitment to organized innovation will increase the odds that you don't chase after an emerging technology that never reaches disruption or fail to take calculated risks that generate revenue.

Effectively traversing the emerging/disruptive spectrum is a matter of survival for today's businesses. We need to develop instincts, skills, and

MORE ON PAGE 15

## BEST OF THE BLOG

teams that enable us to get ahead of disruption without losing our breath and sight of what is important to building the business. A&G

**CHRIS CURRAN** is a PwC principal and chief technologist for the US firm's Advisory Practice, where he is responsible for technology strategy, enterprise architecture and innovation, and the development of thought leadership reflective of PwC's point of view on trends and innovations in technology. He leads the development and analysis of PwC's annual Digital IQ Survey, which measures how well companies understand and capitalize on the value of technology and leverage it to meet their business and customer needs.

## A&G CALENDAR

**IRM UK's Driving Business Performance and Innovation with Enterprise Architecture**
March 6–7, 2014
London
For info: http://www.irmuk.co.uk/events/110.cfm?utm_source=iContact&utm_medium=email&utm_campaign=irmuk&utm_content=

**IRM UK's Understanding Enterprise Architecture: Structure, Domains, Disciplines, Value**
March 12–14, 2014
London
For info: http://www.irmuk.co.uk/events/102.cfm?utm_source=iContact&utm_medium=email&utm_campaign=irmuk&utm_content=

**Troux Worldwide Conference**
March 18–19, 2014
Austin, TX
For info: http://www.troux.com/resources/events/conference2014/

**Forrester's Unleash Your Digital Business**
May 5–6, 2014
JW Marriott Orlando Grande Lakes
Orlando, FL
Register now: forr.com/tm14us or call +1 888.343.6786

Submit your Calendar Events to editor@ArchitectureandGovernance.com