TOGAF or not TOGAF: Extending Enterprise Architecture beyond RUP

Vitalie Temnenco

January 15, 2007

from The Rational Edge: This article contrasts the disciplines of enterprise architecture, solution architecture, and business architecture; compares these disciplines to the IBM Rational Unified Process (RUP) and suggests ways to combine them; and advocates applying The Open Group Architecture Framework (TOGAF) in combination with RUP to advance the implementation of enterprise architecture within organizations.

When provided with a problem statement or pointed to a specific user need, a project team equipped with the IBM Rational Unified Process® (RUP®) approaches a solution by creating a Business Case, a Vision statement, and a Software Requirements Specification among other artifacts. These work products and the activities that produce them are well understood within both the technical and business communities. However, the ways in which we conceptualize, prioritize, and select which business problems and user needs to implement in software remains a highly variable process throughout our industry.

This article explores the maturing and increasingly important role of enterprise architecture (EA) frameworks for today’s software development organizations. I begin by contrasting the discipline of enterprise architecture with the solution architecture and business architecture disciplines, while relating them to RUP. Then I explain how The Open Group Architecture Framework (TOGAF) advantageously expands the boundaries for enterprise architecture set by RUP to include enterprise business and IT planning, implementation governance, and other activities. Finally, I propose ways to apply TOGAF in combination with a few other EA frameworks.

Contrasting different architectural frameworks

In terms of their scope, there is a certain amount of overlap between enterprise, solution, and business architectural frameworks as they are commonly understood. So, how do they relate?
Solution architecture

Solution architecture frameworks take a variety of forms. IT professionals are already accustomed to dealing with Application, Data, Technology and other solution architecture forms (often called domains) during information systems development and maintenance projects. New (and overall more specialized) solution architecture forms, such as Security and Testing, are quickly becoming mainstream also. The most widely recognized solution architecture domains, their major subjects, and the dependencies among them are shown in the Figure 1.

![Figure 1: Domains and subjects of the solution architecture discipline](image)

All the solution architecture domains shown in Figure 1 are considered "technical," as their scope includes various elements of technology, such as software, data, and IT infrastructure. These domains are typically dealt with by technologists -- i.e., people who have backgrounds in systems engineering, software engineering, or IT administration.

Business architecture

Business architecture emerged as a separate domain in the 1990s, when many organizations embraced the business architect role as they attempted to optimize their business processes.

The business architecture discipline is concerned with the "terms of reference" of the business and with describing how it operates. Although not everyone agrees about what components should be included within business architecture frameworks, there is a general consensus that Process and Information, Organization, and Performance aspects are relevant. Each of these components is, in itself, fairly significant and incorporates multiple subject areas, as shown in Figure 2.
Figure 2: Components and subjects of the business architecture domain

The Process and Information component is, arguably, the focal point for business architecture activities because (among other things) it defines, describes, and classifies business processes and supporting structures that comprise the organizational business model. This component also comprises a host of related subjects, such as usability and accessibility. Every organization will, of course, have different business processes, structures, workflows, etc. Likewise, one organization might have that "extra something" in its business process model that gives it an edge over the competition while another organization struggles in this area.

The Organizational component is concerned with the structure and design of work practices, and with the operational style of an organization. The subjects dealt with by this component include the organizational structure, the products and services the business produces, its business units and their locations, and so forth.

Business Performance is a management-oriented component that encompasses subjects that define and quantify organizational efficiency and effectiveness. Concerns about productivity, business risks, and other related subjects belong here.

The enterprise architecture domain

Before I discuss enterprise architecture, I want to sketch the problem this discipline attempts to address. The majority of existing implementation methodologies include techniques for creating the solution to a well-defined business need. These methodologies, however, are not directly concerned with how and why a business need has emerged. Instead, they focus on the relative importance of responding to a given need versus whatever other needs the organization might have. Let's consider RUP, for example. The RUP delivery process takes the decision to begin an implementation largely for granted, and no RUP artifact is directly concerned with evaluating the relative urgency of the business need being addressed by the solution implementation.

In contrast with RUP and other disciplines that focus primarily on implementation, the principle concern of the enterprise architecture domain is the enterprise-wide identification, specification, and prioritization of business needs. The viewpoints discussed and the models drawn in the
context of an enterprise architecture framework address a range of problems both current and potential.

An EA roadmap is likely to include more than a single proposed solution (as illustrated in Figure 3), which may result in multiple, simultaneous implementations.

![Figure 3: A sample enterprise architecture roadmap](image)

**Envisioning the enterprise of tomorrow**

The Institute for Enterprise Architecture Development (IFEAD) sums up the key guiding principle of the enterprise architecture discipline this way: "No strategic vision, no EA." In other words, today’s enterprise architecture is about tomorrow's business systems. An important aspect of this assertion is that enterprise architecture is a holistic discipline that unites business and technology elements based on a single, strategic enterprise vision (see Figure 4).

![Figure 4: Elements of the enterprise architecture domain](image)
Now is the time

Although enterprise architecture concepts have been around for more than two decades, the EA discipline began gaining momentum more recently. This can be attributed to the acceleration of environmental changes in organizations of all sizes across most industries. Business agility and, in particular, the ability of the technology infrastructure to respond to change in a timely manner have reached critical importance.

Another factor contributing to a growing appreciation for the enterprise architecture discipline has been the more stringent regulatory climate of recent years, both in the US and elsewhere, which is driving organizations not only to improve their accountability and reporting practices, but also to make compliance organic to every business process.

In response to a sharper focus on EA principles, several robust enterprise architecture frameworks have recently emerged, such as TOGAF.

Similar subjects, different perspectives

The enterprise architecture discipline touches on virtually all the same subjects as the solution architecture discipline, but from a different perspective and in a distinctly different context. The EA context is holistic and its perspective is organizational, while the solution architecture is implementation-specific.

By the same token, the enterprise architecture discipline is more than a superset of solution architecture domains. While the subjects of solutions architecture (see Figure 1) are meant to be applied to the solution implementation, the EA subjects (shown in Figure 5) are largely used for enterprise analysis, planning, and architecture governance. For example, to a solution architect, the "Applications" subject area can refer to a group of linked components and classes, whereas for an enterprise architect it can entail a node running a group of business processes or providing a range of services.

Note that some (typically low-level) subjects from the solution architecture discipline are not included in the scope of EA, while several additional (mostly higher-level) subjects are added. Also note that the key business architecture subjects are subsumed intact within the EA discipline.
Figure 5: Components and subjects of the enterprise architecture discipline

Scope of the enterprise architecture

Enterprise architecture activities start long before projects driven by solution architecture commence, and are ongoing continuously throughout the lifespan of the enterprise. As a continuously running process, the enterprise architecture lifecycle has a single entry point that coincides with the creation of a practice, along with multiple process inputs that are distributed along the enterprise architecture timeline. As described in Figure 10 below, some EA frameworks introduce cycles or iterations, which must align with major phases in the organization's business development. In these frameworks a cycle consists of multiple phases with multiple feeds into EA activities, especially in the initial phases of a cycle.

Enterprise architecture inputs can differ in terms of how the enterprise transformation ideas are generated and validated; the classification of these inputs provides a way to measure the maturity of the organization's EA practice (see Figure 6). In some cases an idea is a by-product of an elevator talk between a CTO and an architect. In others, it is a coordinated response to requests escalated by the business planners, end users, or other stakeholders. In the case of the most mature organizations, it is an output of the strategic architecture analysis and planning process that also includes input from sources like those just mentioned.

Figure 6: Inputs to the enterprise architecture process

Enterprise architecture activities last as long as the enterprise is operational and has a vision. Yet, there is always a need for an EA perspective in the organization and EA activities must never cease.

The emerging role of enterprise architect

An enterprise architect is a thought leader, visionary, and industry expert. In most companies this is a new role that combines the skills of project manager, solution architect, and business analyst with the intuition of an executive.
A common limitation in perspective among many IT architects is that they are grown-up programmers and tend to be very inward-looking. While this is not altogether a hindrance for architecting and designing "boxed" solutions, it is a less desirable trait in the context of "architecting" the enterprise. An enterprise architect is likely to be more extroverted and well able to use professional, working, and even personal relationships with business owners, business leaders, colleagues, and customers to interpret, architecturally describe, and help to execute the enterprise vision (see Figure 7).

**Figure 7: The enterprise architect role**

The function of an enterprise architect is often compared to that of a city planner. In contrast, the function of a building architect is more readily associated with the IT architect role. The enterprise architect role often emphasizes the inductive skills of a detective over the deductive skills of a builder.

However, the high-level perspective of the enterprise architect does not mean that this role is disengaged from the user community. On the contrary, an enterprise architect must be involved in helping customers understand their real needs (as opposed to wants) and to work with them throughout the implementation of a solution. At the same time, an enterprise architect should be able to view his or her domain at a level of abstraction that forestalls direct involvement in the practical aspects of implementations. As David Jackson of IBM put it, an enterprise architect should "be able to understand the business problem and the business domain and explain it to the technical people and to be able to understand the technology domains and explain the technical possibilities to business people."

It is important that an enterprise architect play a pivotal role in architecture governance, a function that is often either shared between assorted business and technical roles or, even worse, simply
ignored. Architecture governance is the glue that provides both a context and a framework for all enterprise and project architecture activities.

**Enterprise architecture and RUP**

Although the core RUP does not incorporate the enterprise architecture discipline, RUP would definitely benefit from having a well-defined interface with EA. This is demonstrated in Figure 8, which highlights the serial nature of solutions implementation projects. As shown, a typical organization has a single instance of continuously running enterprise architecture process along with an arbitrary number of consecutive solution implementations.

![Figure 8: RUP versus enterprise architecture lifecycles](image)

It is worth mentioning that several attempts have been made to extend the scope of RUP toward the enterprise as a whole. The most recent is the Enterprise Unified Process, or EUP. EUP does not focus on the enterprise architecture function alone; instead, it provides a framework for performing a wide range of enterprise activities. EUP introduces as many as seven new disciplines, including the Enterprise Architecture discipline, and more than twenty-five new roles, and also provides guidance for their tailoring. Work is underway to integrate various EUP disciplines into the core RUP.

Meanwhile, RUP itself has evolved considerably since its inception and currently includes several enterprise processes, including Business Modeling and Change Management, in its core disciplines.

**Implementing an enterprise architecture program**

The relative complexity of enterprise architecture program implementation depends on such factors as the organization's level of commitment, availability of resources and guidance, size and complexity of the organization's business model, and organizational agility. The reality is that many organizations are simply not capable of implementing and maintaining an enterprise architecture program all at once, and would be better served to focus on less far-reaching process improvement techniques that emphasize efficiencies rather then effectiveness.  

TOGAF or not TOGAF: Extending Enterprise Architecture beyond RUP

Page 8 of 19
There are two general approaches to implementing enterprise architecture, which roughly correspond to the two distinct kinds of frameworks available.

The first approach projects organizational artifacts and processes onto the framework meta-structure. This approach works well for organizations that are very proficient at doing modeling. Organizations that favor this approach usually choose Zachman or an equivalent framework. One danger of this approach is that the framework structure may constrain creativity and add bureaucracy to the EA implementation process. The other problem with this type of framework stems from the serious shortage of implementation guidance.

The second approach is based on a belief that an enterprise architecture program has to be process-driven. Since this approach primarily focuses on activities rather than artifacts, it may be easier to understand and link with the existing enterprise and solution methodologies and techniques.

While both approaches have their pros and cons, a compromise could be to use an activity-driven process overall, while applying a meta-framework as a supporting structure or for analysis purposes.

A sample enterprise architecture implementation checklist

Here are some of the basic activities that must take place as part of any EA implementation. This list should give you an idea of what the enterprise architecture work is really all about:

- **Study existing business practices.**
  Understanding the business model of your organization and, at a minimum, its high-level business processes is a pre-condition for commencing the enterprise architecture implementation.

- **Engage with senior management to understand strategic intent.**
  Obviously, senior management holds a key to interpreting strategic vision. Understanding a vision is critical for the purposes of drawing the roadmap of the enterprise architecture, as this artifact will drive the "to-be" part of the architectural work to come (more about this below).

- **Connect with the business community to uncover urgent needs.**
  While senior management may be envisioning the organization's future state, the business community holds more answers about its current state. Your goal is to extract those facts and level them with the expectations set by the strategic vision.

- **Build a panoramic understanding of the existing technology environment.**
  Technology is a major enabler of business processes (the other one being people), which implies that you won't go too far without proper understanding of your main tool.

- **Draw the improvement roadmap.**
  Having collected data from various sources, create a roadmap to advise the stakeholders -- including senior management, business, and technology leaders -- about how you intend to act upon the needs they have communicated to you.

- **Keep enterprise architecture models up-to-date.**
  Needless to say, after you create the roadmap for the enterprise architecture and receive stakeholder approval on it, you ought to make the best effort to update it over time.
For the activities just described to take place, strong methodological guidance is an imperative. The enterprise methodologies and frameworks that exist today vary significantly in the range of problems they solve and the approaches they take. Some of the most well-known frameworks are TOGAF, EUP, the Federal Enterprise Architectural Framework (FEAF), the Gartner EA Framework, the Department of Defense Architecture Framework (DoDAF), the Spewak EA Planning Methodology, and the Zachman Framework.

If you believe that the organization you work for is lacking in some of the areas discussed in this article, or you want to improve its effectiveness, or you personally have responsibilities relating to its enterprise architecture, then I recommend that you take a closer look at some of these enterprise architecture methodologies.

**EA framework choices**

When it is time to choose an enterprise architecture methodology/framework, most of the options available take the form of partially built "solutions" that can be adapted to specific organizational needs. In reality, the majority of these "half-solutions" are either not pragmatically reusable, or they require a very significant degree of tailoring to be of any value. Further, a serious problem with tailoring any of these frameworks is that the amount of guidance provided is woefully small while the level of detailed understanding required is very high.

Most, if not all, existing frameworks either extend other frameworks or recapitulate them for particular applications. For example, EUP is an extension of RUP that mimics its approach to describing process workflows and activities, while both FEAF and Spewak inherit from the Zachman Framework. TOGAF originates from the early, proprietary EA technical frameworks like the Technical Architecture Framework for Information Management (TAFIM), and builds upon ANSI recommendations for enterprise architecture (IEEE 1471-2000).

**Introducing TOGAF**

TOGAF is an enterprise architecture framework that emerged during the last two decades with the objective of becoming a standard for EA development. Created by members of the Open Group consortium, TOGAF has not always embodied a holistic EA focus. Initially, TOGAF only included technical architectures (versions 1 through 7); however, more recently the business architecture domain was added to the framework (v. 8, Enterprise Edition), which quickly propelled TOGAF to the front seat among today’s EA framework options (see Figure 9).
By the time the business architecture domain was added, TOGAF had already built a solid technological foundation with its Applications, Data, and Technology architecture domains. The addition of the business architecture domain gave TOGAF's growing popularity a further boost, while the other frameworks that started from scratch on the technology architecture had a difficult time embracing it. For example, EUP had to draw upon RUP's approach, techniques, and notation (UML), while Zachman, Spewak, and some others intentionally kept their guidance at a high level of abstraction, which negatively affected their acceptance because they failed to win the support of the technical community.

**TOGAF components**

A key component of TOGAF is the Architecture Development Method (ADM), which is the process used for organization-specific enterprise architecture implementation and tailoring. Besides ADM, TOGAF includes a collection of reference assets known as the "Enterprise Continuum." TOGAF envisions the Enterprise Continuum to act as a collection of reusable building blocks (patterns) that provide enterprise architecture teams with reference architectures, models, and processes that can be adopted in a Lego-like fashion.

**The TOGAF ADM**

The TOGAF Architecture Development Method (ADM) provides complete guidance for implementing and executing an organization's enterprise architecture. The process consists of multiple, consecutive phases enclosed in a closed loop (see Figure 10).
Figure 10: TOGAF Architecture Development Method (ADM)

The purpose of the Preliminary Phase is to identify implementation process stakeholders and get them on the same page about the enterprise architecture work. This phase delivers Architecture Guiding Principles that is based on the organization's business principles and describes process and criteria for monitoring the EA implementation's progress.

Phase A of the process is dedicated to articulating the EA vision. The Architecture Vision artifact draws upon the business drivers to clarify the purpose of the enterprise architecture effort and to create first-cut descriptions of the baseline and target environments. If the business objectives are unclear, then part of the work in this phase is to help the business to identify its key objectives and corresponding processes, which the enterprise architecture must support. The Statement of Architectural Work, which is also produced in this phase, delineates the EA's scope and constraints and presents a plan for the architectural work.

Phase B is dedicated to detailed work on the architecture of the business domain. Both baseline and target architectures that were outlined in the Architecture Vision are detailed to make them useful inputs into technical analysis. Business process modeling, business object modeling, and use case modeling are some of the techniques that are used to produce the Business Architecture, which in turn includes gap analysis of the desired state.

Phase C is concerned with delivery of Application and Data (Information) architectures. This phase draws upon on the baseline and the target architectures started in Phase A (Architecture
Vision) and results of the business gap analysis (part of the Business Architecture) to deliver Application and Data architectures for both current and envisaged environments, within the scope and according to the plan outlined in the Statement of Architectural Work.

Phase D completes the detailed architecture work of the TOGAF ADM cycle with the delivery of Technology Architecture. As in the preceding phases, gap analysis and draft architectures are used as a baseline, as are the architecture guiding principles agreed upon in the preliminary phase. Modeling notations, such as UML, are actively used during this phase to produce various viewpoints.

The purpose of Phase E is to clarify the opportunities presented by the target architectures and to outline the potential solutions. The work in this phase revolves around feasibility and practicality of the implementation alternatives. Artifacts produced here include Implementation and Migration Strategy, High-level Implementation Plan, and Project List, as well as an updated Application Architecture that acts as a Blueprint to be used by the implementation projects.

Phase F moves to prioritize proposed implementation projects and perform detailed planning and gap analysis of the migration process. This work includes assessing dependencies between projects and minimizing their overall impact on enterprise operations. In this phase, the Project List is updated, the Implementation Plan is detailed, and the Blueprint is handed to the implementation teams.

As the list of projects stabilizes, the focus shifts to formulating more specific objectives and recommendations for each of the implementation projects. During Phase G, connection between the governing architecture (TOGAF) and the development organization (which may be regulated by the combination of RUP and the Project Management Body of Knowledge (PMBOK), for instance, or some other project management methodology) is established and selected projects are implemented under the formal architecture governance. A phased deliverable is Architecture Contracts, which are accepted by the development organization. The ultimate outputs of Phase G are the architecture-compliant solutions.

The focus in Phase H shifts towards change management of the architecture baseline that is achieved with the delivery of the implemented solutions. The phase might produce a Request for Architecture Work that sets targets for a subsequent cycle of enterprise architecture efforts.

**The TOGAF Enterprise Continuum**

The Enterprise Continuum is a pool of resources, such as models, solution patterns, and other assets that can be used as building blocks throughout the process of enterprise architecture implementation and tailoring. Essentially, the Enterprise Continuum is analogous to a reference library for EA practitioners and stakeholders.

Besides the ADM and the Enterprise Continuum, TOGAF provides additional resources and references pertaining to techniques, methods, and solutions that can be used for both planning and implementing the enterprise architecture.
Is TOGAF iterative?

As Figure 10 illustrates, TOGAF is an iterative process. TOGAF iterations (which are more often called cycles) are normally longer than RUP iterations, and span multiple phases of enterprise architecture implementation and maintenance. This is not surprising, as the scope of an EA is much larger than a single RUP project.

Requirements management and TOGAF

As shown in Figure 10, TOGAF is -- literally -- a requirements-centric process. ADM requirements management deals with all kinds of requirements, including most notably business drivers, concerns, and new functionality and change requests. Last but not least, since architectural requirements are subject to constant change, requirements management happens throughout the entire EA implementation lifecycle.

TOGAF and other methodologies

In the modern organization TOGAF has to co-exist with other methodologies, processes, and frameworks. With some of them, like RUP, it does not share a common scope; while with the others, like the EUP and the Zachman framework, there is overlap across part of the EA lifecycle.

Where TOGAF meets RUP

TOGAF is sometimes seen as an alternative to RUP, which it is not. Although many models and subjects are held in common between the two frameworks, they have different drivers and purposes.

The principle difference between TOGAF and RUP is that RUP is technology architecture-driven, whereas TOGAF is business architecture-driven. In RUP, business requirements are gathered to design and deliver a software-based system, whereas in TOGAF technology is seen as a way to realize a business vision.

The purpose of RUP as a Software Development Life Cycle (SDLC) methodology is to support timely and effective applications delivery. This contrasts with TOGAF's purpose, which is to support enterprise architecture implementation and maintenance. Figure 11 shows the extent to which the RUP and TOGAF lifecycles overlap.
There is a clearly defined point in the TOGAF implementation process where it intersects with RUP. The starting point of intersection is at the beginning of Phase F, when architecture blueprints (application model), business architecture, and high-level implementation plans are handed over to the implementation teams. The hand-over is considered complete when both sides agreed on the scope of the implementation. At the same time, Architecture Contracts are signed which detail groups’ responsibilities and principles of architecture implementation governance. Figure 12 provides details on the artifact flow between TOGAF and RUP.

**Figure 11: RUP and TOGAF lifecycles**
TOGAF or not TOGAF: Extending Enterprise Architecture beyond RUP

Figure 12: Artifact flow between TOGAF and RUP

TOGAF and the Zachman Framework

Although both TOGAF and the Zachman Framework (ZF) belong in the category of “enterprise frameworks” they differ in their approach, composition, and the terms of reference.

While the ZF is a structural (static) framework that is most effective when used as a model for analysis and classification of artifacts and meta-analysis of methodologies and frameworks, TOGAF is a process (dynamic) framework that also includes reference process models guidance for using them. Notwithstanding these fundamental differences there is broad potential for applying the two frameworks together:

- Use ZF as a dictionary and a delivery structure for artifacts, and use TOGAF as an artifact delivery process
- Use ZF for setting up the structures for transitioning artifacts between other solution methodologies, such as RUP and/or the IT Infrastructure Library (ITIL) and TOGAF
- Use ZF for gap analysis of the existing enterprise process and business models during or before implementing TOGAF
- Use ZF for meta-analysis of TOGAF, which may lead to identification of its weaknesses and to potential interfaces with other methodologies
- Use ZF as an aid to designing TOGAF enterprise architecture models (TOGAF phases B-D)

TOGAF and EUP

Although both EUP and TOGAF function at an organization-wide level, their scopes differ. EUP, which extends RUP into the enterprise, introduces seven new disciplines -- Enterprise Architecture being one of them -- and provides guidance for applying them. Although most EUP disciplines function at an organization level and act as inputs to enterprise architecture processes, EUP is not an enterprise architecture development framework per se. (Although EUP may change in relation to future implementations of RUP.)

In contrast with EUP, TOGAF focuses on the enterprise architecture discipline alone. It was envisaged and created as a framework for enterprise architecture implementation from the beginning.
Despite the differences in perspective between EUP and TOGAF, the disciplines introduced by EUP are very important to an enterprise architecture implementation. For example, Enterprise Business Modeling is instrumental for "architecting" changes to the existing business processes, while Portfolio Management is a "must use" tool for an enterprise architect to use for analyzing potential and monitoring (or in EA terms, governing) ongoing implementations.

**Potential TOGAF implementation hurdles**

TOGAF ADM is not an end-to-end process; it is a framework and, as such, requires organization-specific tailoring. Tailoring assumes both strong methodological and business model knowledge -- qualities that might not be easy to find. The situation is complicated by the fact that TOGAF v 8.0, Enterprise Edition is a relatively new process, with limited street expertise available.

It may be hard to obtain buy-in from stakeholders to use TOGAF. It usually takes an influential employee to convince the senior management and staff that TOGAF processes and models ought to offset or replace those familiar to them.

Despite recent enhancements, TOGAF is not as detailed as some other methodologies, especially in its core area of business and technology liaison. For example, TOGAF does not include anything like Rational Method Composer or MyRUP, nor does it have as comprehensive periodical library available.

Mature organizations, especially those that routinely use project management and software lifecycle methodologies, such as PMBOK, RUP, Scrum, and ITIL are typically more successful at implementing enterprise architecture. I recommend that organizations get comfortable with a couple of techniques, notations, and SDLC methodologies before bringing TOGAF into the organization, as illustrated in Figure 13.

![Figure 13: TOGAF implementation roadmap](image)

Other potential threats to a successful EA implementation include lack of the standard tool for capturing and managing EA artifacts and lack of a standard notation.
TOGAF or not TOGAF?
The foremost choice that an organization has to make is whether implementing enterprise architecture in general, and TOGAF in particular, is the next logical step in its evolution. Often, organizations that successfully and consistently use solution lifecycle methodologies like RUP look at implementing EA as the next milestone in their evolution.

It is critical that enterprise architecture not be forced into the organization by decree. The demand for an enterprise architecture practice should, instead, arise organically from the understanding of ongoing complexities that surround the management of business processes and structures in combination with the supporting technology infrastructure.\(^9\)

If you are not sure whether TOGAF is for you, it might be a good idea to take it for a test drive in your organization. A trial might include a group of enthusiastic and trusted solution and business architects. It is important to have both business and technical roles in the mix to ensure a balance of interests. Also note that, to become successful, the EA implementation has to secure support in the organization. Support can be ensured by including a member of the executive team as a proxy to the EA team, or by securing a seat for the enterprise architect on the organization’s steering committee.

Acknowledgements
I wish to express my gratitude to Jason Uppal, the Opeg Group TOGAF certified master architect #1, who inspired me to write about the subject, as well as to David Bentley and John Reading for their support and feedback.

Notes
1


5 The Open Group website ([http://www.opengroup.org/togaf/](http://www.opengroup.org/togaf/)) is the most comprehensive source of information about the TOGAF framework.

6 See "UML, RUP, and the Zachman Framework: Better together" for more information on this bullet and the one following.

See http://www.agilemodeling.com/essays/realisticUML.htm for Scott Ambler's ideas on the sufficiency of UML for enterprise and solution modeling purposes.

The paper From Business Process Design to Enterprise Architecture from IDS Scheer is a great source of ideas on how enterprise architecture evolves within organizations.