

Enterprise-architecture and the service-oriented enterprise

Tom Graves, Tetradian Consulting

TOGAF Glasgow, April 2008

info@tetradian.com / www.tetradian.com

21-Apr-08

(c) Tom Graves / Tetradian 2008

1

Tetradian Consulting
Unit 215, Communications House
9 St Johns Street
Colchester
Essex CO2 7NN
England

www.tetradian.com
info@tetradian.com

© Tom Graves / Tetradian 2008

SOA and SOE

- **Service-oriented architecture (SOA)**
 - SOA still viewed primarily in terms of IT
 - common interfaces for heterogeneous systems
- **The service-oriented enterprise (SOE)**
 - extends SOA metaphor to all aspects of the enterprise
- **TOGAF and SOA**
 - IT-architecture underpins IT-service architecture
- **Adapt TOGAF ADM for use with SOE**
 - extend IT-centric methodology to whole-of-enterprise

Service-oriented architecture, or SOA, is well-known as a way to structure the links between different IT systems.

So it has obvious connections with enterprise-architecture. A main theme of TOGAF conferences has been about how IT-architecture underpins the IT aspects of SOA.

Yet SOA can – or *must* – be about more than just IT. So the aim here is to extend SOA ideas to the whole of the enterprise – in other words, the service-oriented enterprise – and adapt the TOGAF ADM to suit.

The service-oriented enterprise

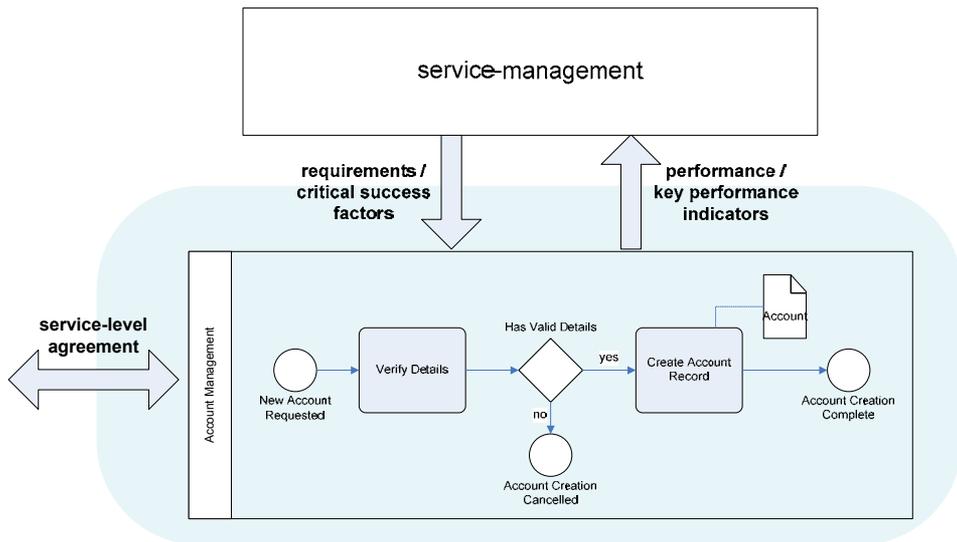
- ‘Service’ concept as unifying theme
- SOE analogy with UNIX
 - in UNIX, every device is treated as a file
- In SOE, *everything* delivers a service
 - use SOA for architecture of entire enterprise
- Products are ‘services’ too
 - deliver capability for client ‘self-service’
 - vacuum-cleaner → self-service of ‘cleaned floors’
 - computer → self-service ‘access to applications’

There’s an analogy here with the Unix principle that every device is treated as a file.

In the service-oriented enterprise, we gain unity and consistency by treating every activity as a service. From IT-interfaces to board-level strategic assessments, and everywhere in between, everything is a service.

In a sense, even products are services – they give the customer a capability to deliver self-service via the product.

Services and business-processes



A simple example of a business-function / business-service

21-Apr-08

(c) Tom Graves / Tetradian 2008

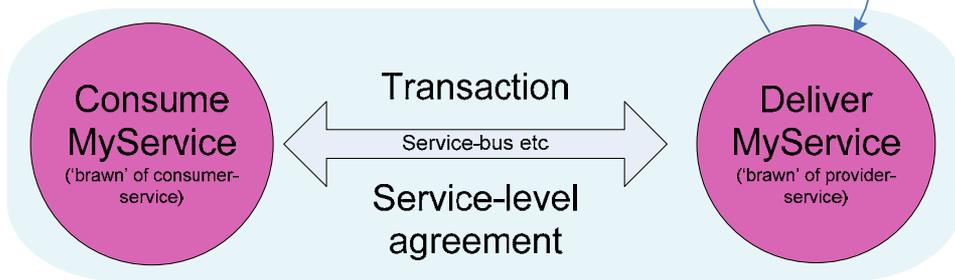
A business process is a network of transactions between services. IT-based SOA is concerned mostly with the low-level choreography and underlying mechanisms.

But there are also transactions that deal with service performance – monitoring service-level agreements (SLAs) between services; key performance-indicators (KPIs) that report on service performance; and critical success-factors (CSFs) that guide interpretation of KPIs. This information passes up and down through a web of other inter-linked services.

An IT view of services

Emphasis in IT SOA is on
**service-to-service
transactions in near-real-time**

- monitoring often seems to be tackled only as an afterthought



21-Apr-08

(c) Tom Graves / Tetradian 2008


TETRADIAN
the futures of business
5

In abstract terms, each service has two layers: service-delivery; and sub-systems to guide and monitor service-delivery. We could call these the 'brawn' and 'brain' of the service.

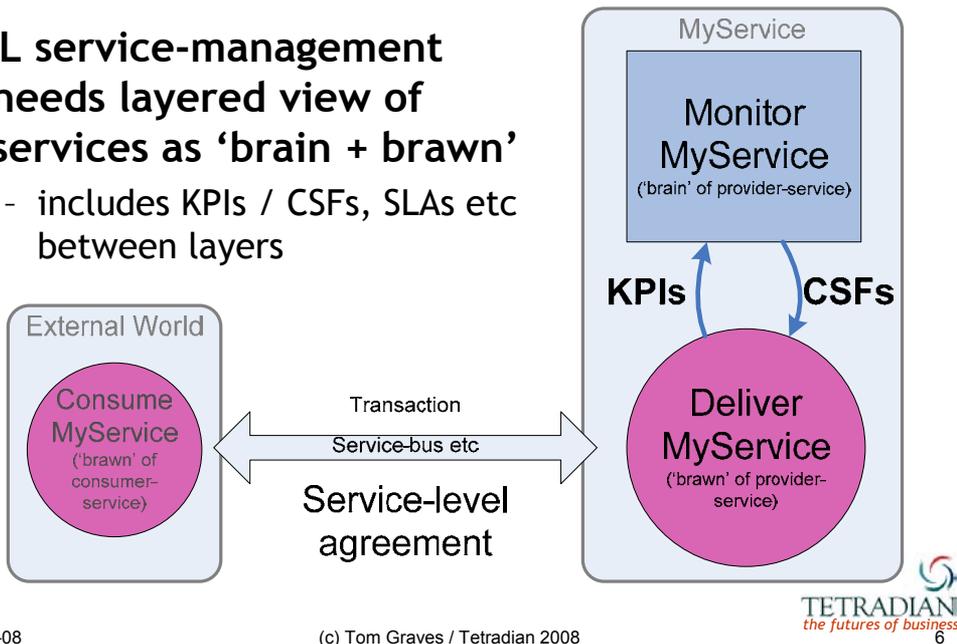
In most IT-centric SOA, the emphasis is on 'brawn to brawn' – the service transactions, and conditions and metrics to monitor the SLAs. In terms of time, this all happens in the 'now'.

KPIs and CSFs are almost invisible at this level.

An ITIL view of services

ITIL service-management needs layered view of services as 'brain + brawn'

- includes KPIs / CSFs, SLAs etc between layers



21-Apr-08

(c) Tom Graves / Tetradian 2008

TETRADIAN
the futures of business
6

But in service-management we want to know about the service's 'brain' as much as its 'brawn'. We need to keep track of what's going on, through all the KPIs and CSFs and so on.

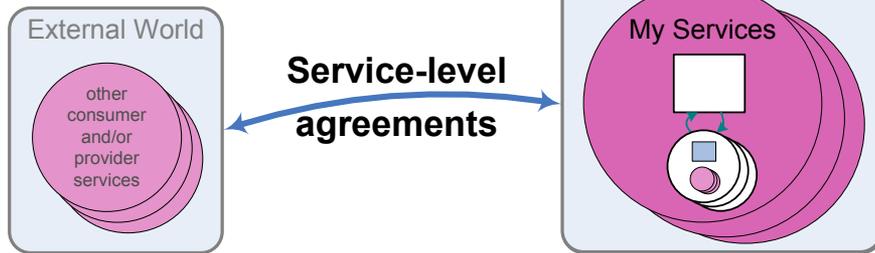
And we need a broader sense of time than just the 'now'. We need a picture of what's happened in the past, and prepare and plan for the future.

Even in IT service-management, many of these 'brain' services can be done only by people, not by machines. To make this work, we must expand our view of SOA beyond the 'IT-only' box.

A layered view of services

Hierarchy of services, each with their own KPIs, CSFs and SLAs

- each 'brain + brawn' pair becomes the 'brawn' of the next level upward



Viable System Model ("Brain of the Firm", Stafford Beer, 1972)

21-Apr-08

(c) Tom Graves / Tetradian 2008


TETRADIAN
the futures of business
7

The 'brawn / brain' pairs don't exist in isolation. It's useful to think of them as stacked in layers, services 'enclosed' within more abstract business-services and functions.

KPIs and so on migrate upward for a business view of service-delivery; we'll also want to 'drill down' to see the detail.

We could map this to Zachman, for example. Or the way that transaction-data becomes information, and then knowledge; we hope somewhere it becomes wisdom!

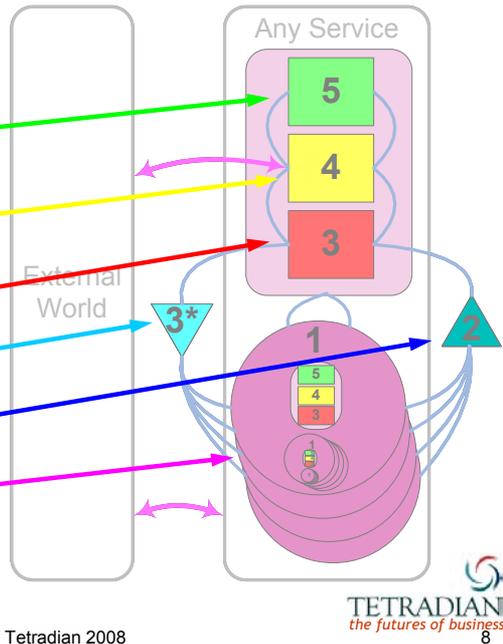
This structure is recursive: each 'brawn / brain' pair becomes a 'brawn' at the next layer. This is a core principle of a long-proven structure called the Viable System Model (VSM).

Viability System Model (VSM)

In the Viable System Model, every service contains a set of specialised sub-systems

- 5 - *policy / purpose*
- 4 - *'outside / future' [+ strategy]*
- 3 - *'inside / now' [management]*
- 3* - *sporadic audit / review*
- 2 - *sub-process coordination*
- 1 - *process operations*

These interact with each other to act on and with the external world



21-Apr-08

(c) Tom Graves / Tetradian 2008

TETRADIAN
the futures of business
8

VSM focuses on management and management-support – on guiding principles and the future view as well as day-to-day management. Each service has sub-systems to do specific tasks.

Note that choreography (system-2) and quality-audit (system-3*) exist partly outside of the main management hierarchy.

If the overall enterprise is to be 'viable' - especially in the longer term - all these sub-systems need to exist in every layer and every service.

xVSM and SOE - review

- **xVSM is map of interfaces needed for SOE**
 - interdependency implies need for interfaces
- **Use xVSM to model service-completeness**
 - all standard-VSM links must be present for service 'viability'
 - all xVSM links must exist for optimum service-performance
- **Links identify service interdependencies**
 - each interlink requires information-system support in some form

21-Apr-08

(c) Tom Graves / Tetradian 2008



Here's a quick summary so far of how the viable services model can support structure and 'completeness' for the service-oriented enterprise.

Every path between 'systems' also needs information exchange – relevant for whole-of-enterprise IT design.

The xVSM completeness-checklist

- **5**: what is the service's *purpose*? who/what defines *policy*?
- **4**: what current *strategy*? outside *relationships*? who defines this?
- **3**: how are its *tasks defined, managed and monitored*?
- **3***: what *random checks / audits* will *verify performance*?
- **2**: how is it *coordinated* with other services?
- **1**: what does it *do*? how does it do it? how does it support its 'downline' services (if any)?
- **X**: how does it identify and resolve any *run-time exceptions*?
- **C**: what *corrective-action* does it undertake for *causes* of issues?
- **M**: how does it *track* and manage *quality-issues* and other issues?
- **P**: how does it manage *improvement* of its *processes*?

21-Apr-08

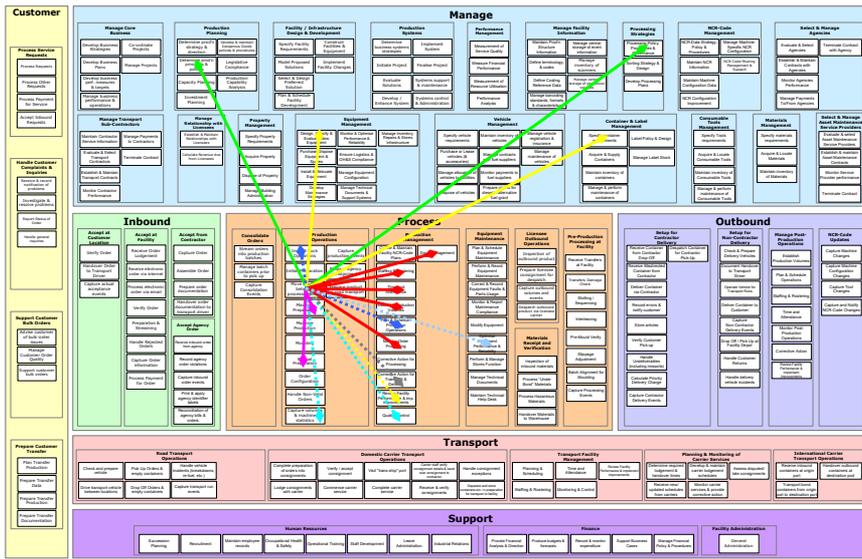
(c) Tom Graves / Tetradian 2008



This is the xVSM checklist we used to review 'viability' and quality-management in that transformation project.

We used the project's existing four-tier business-function model as the information-source for the review.

xVSM analysis: logistics [1]



Logistics organisation: mid-level function-model with example xVSM mappings

21-Apr-08

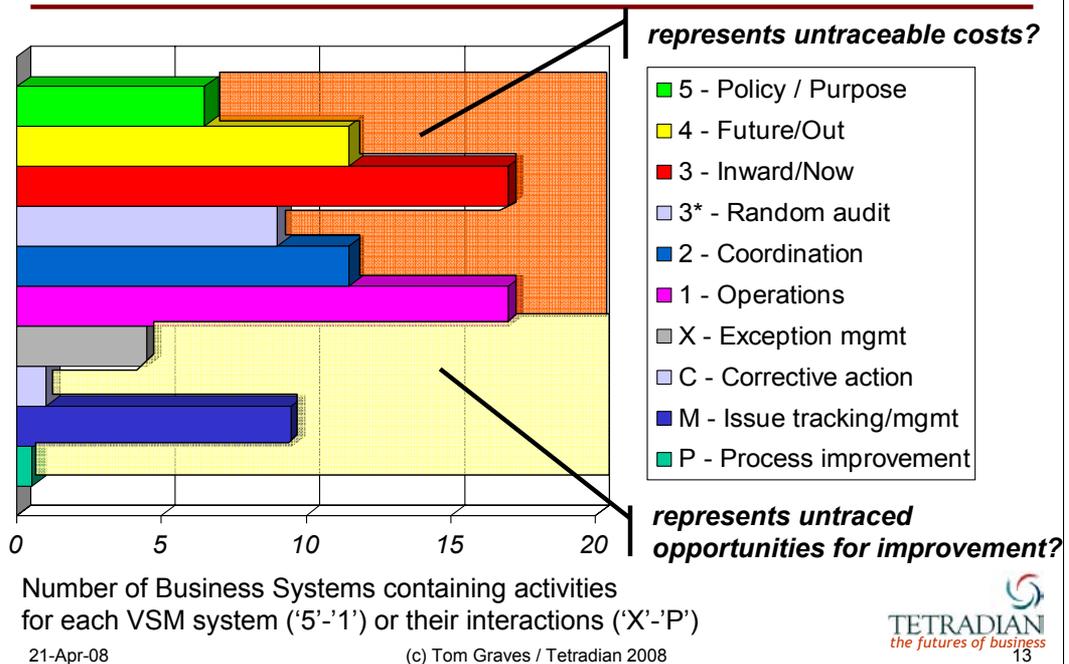
(c) Tom Graves / Tetradian 2008



This is a graphic version of the function model, with an example set of viable-services mappings. Service-support escalates upward into management and strategy layers.

(For confidentiality, I've changed the detail, to look like a plastics factory. The point is that the same principles apply to every enterprise.)

xVSM analysis: logistics [2]



Since every VSM 'system' must exist for viability, untraceable VSM links represent untraceable costs; untraceable xVSM links represent potential problems in quality- and service-management.

This was the result of cost-tracing for the transformation-project, mapping xVSM links for twenty business-systems.

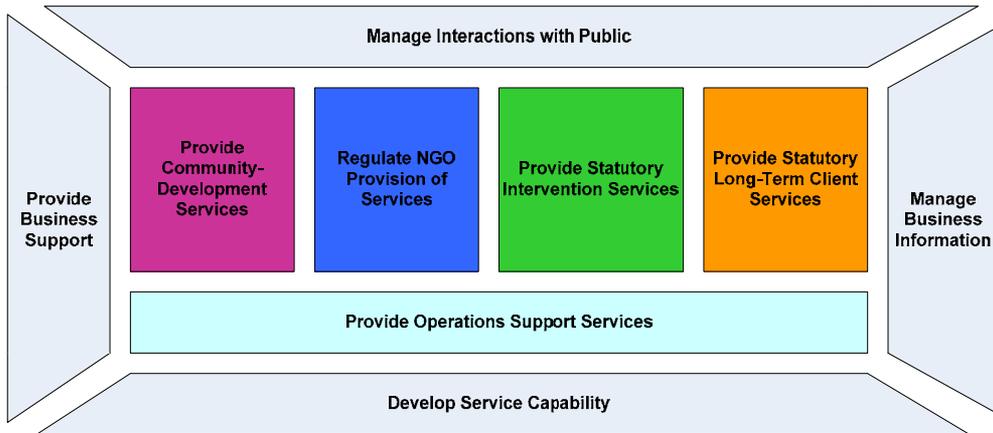
Barely half the costs could be traced to the respective system.

Barely a *sixth* of the needed quality-management links could be identified.

Scary – but also a valuable call to action.

Some of the problems were already known, but this made the reasons *visible* – and what to do about them.

xVSM analysis: human services [1]



Government human-services department: initial high-level function-map, based on existing organisation structure

21-Apr-08

(c) Tom Graves / Tetradian 2008


TETRADIAN
the futures of business
14

Another Australian example, a large government department in the human-services sector. This was the first cut for a high-level, whole-of-enterprise view of its business functions and services.

Although IT is very important here, it's not an IT-centric organisation – it's not a bank, or insurance. The real complexity is between IT-based and people-based parts of business-processes. For this it's essential to break out of the 'IT-only' mindset of most 'enterprise-architecture'.

xVSM analysis: human services [2]

	Provide Community-Development Services	Regulate NGO Provision of Services	Provide Statutory Intervention Services	Provide Statutory Long-Term Client Services	Manage Government and Public Relations	
Contact						
Deliver						
Fund to Deliver						
Educate and Train						
Coordinate						
Monitor and Improve						
Research, Prepare, Plan						
Support Frontline Services						
	Manage Resources	Manage Corporate Administration	Provide Operations Support	Manage Workforce	Improve Practice	Manage Information
Deliver						
Fund to Deliver						
Educate and Train						
Coordinate						
Monitor and Improve						
Research, Prepare, Plan						

Same department: revised high-level function-map as matrix against on two-tier xVSM categories

21-Apr-08

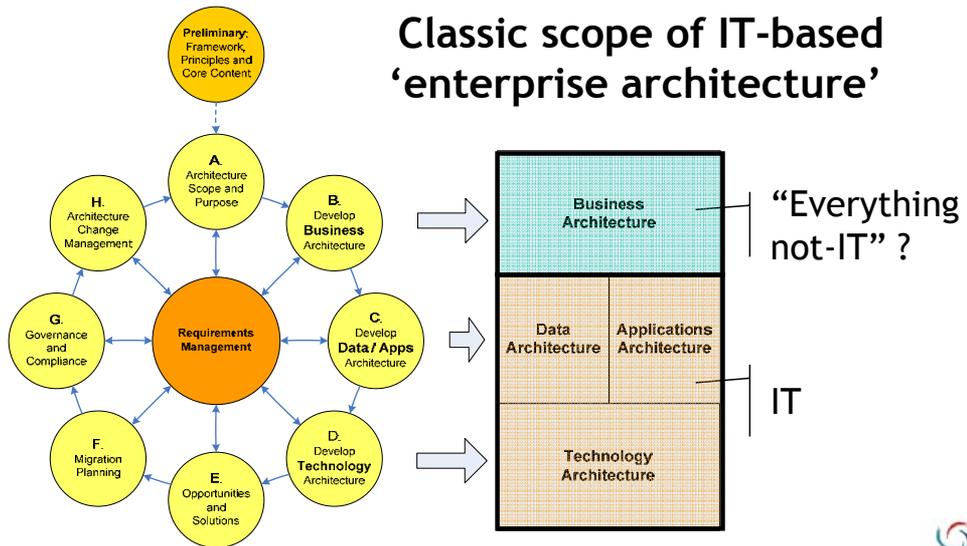
(c) Tom Graves / Tetradian 2008



This is a later version of the same high-level business-function model, restructured as a two-layer matrix between high-level functions and the extended VSM.

This grid-layout makes it much easier to see the probable dependencies – and information-needs – as we drill downward into the lower-level detail.

TOGAF and SOE scope



21-Apr-08

(c) Tom Graves / Tetradian 2008


TETRADIAN
the futures of business
16

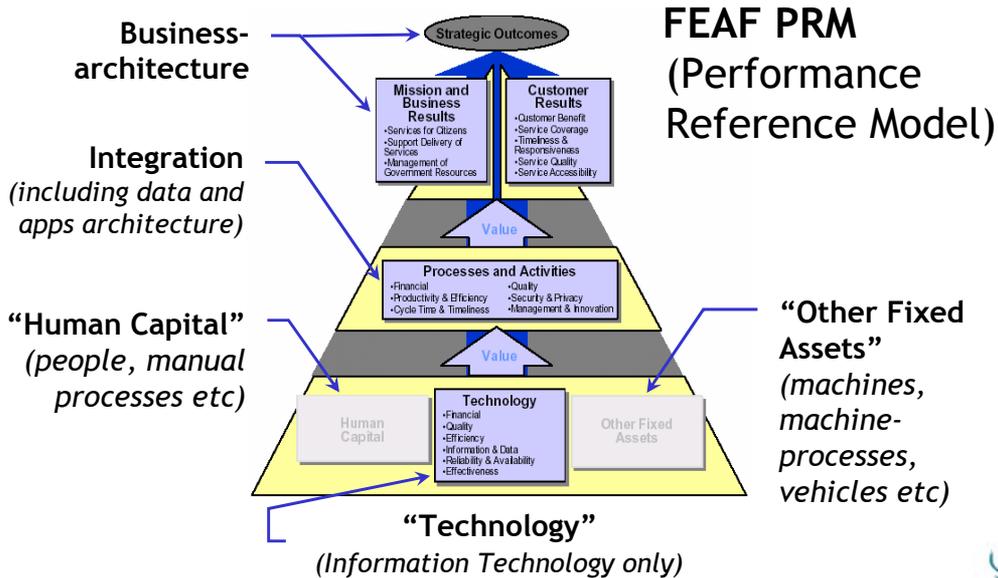
So where does TOGAF come into this?

The TOGAF ADM should be the best methodology and discipline for developing architecture for the service-oriented enterprise.

But at present the ADM is too IT-centric. So IT-centric that, in TOGAF 8.1, 'business architecture' is just a grab-bag for 'everything not-IT'.

To make TOGAF usable for SOE, we must adapt the ADM for a much broader scope.

FEAF and SOE scope



21-Apr-08

(c) Tom Graves / Tetradian 2008

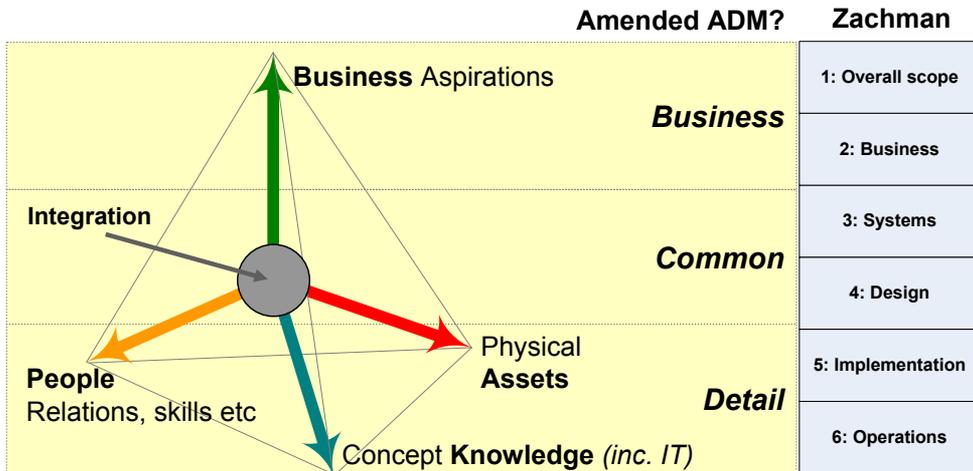

TETRADIAN
the futures of business
17

Compare the Federal Enterprise Architecture Framework’s ‘Performance Reference Model’. At first glance it’s much the same as TOGAF.

But notice the two greyed-out boxes on either side of Information-Technology – placeholders for future work, labelled ‘Human Capital’ and ‘Other Fixed Assets’. They’re distinct domains, separate from business-architecture proper.

Rather than bundling everything non-IT into ‘business architecture’, the ADM does need to treat these as separate domains.

Dimensions of SOE scope



Tetradian - dimensions of architecture

21-Apr-08

(c) Tom Graves / Tetradian 2008



So we need to deal with not just two dimensions, but at least four, plus their integration.

It's useful to visualise this as a 'tetradian', a four-axis tetrahedron.

We can also see how these dimensions map loosely to Zachman, and to a broader view for the ADM.

Visualisation of SOE scope

Business Architecture		
People Integration-Architecture	Information Integration-Architecture	Machine / Asset Integration-Architecture
<i>IT domain (typical)</i>		
Manual-Process Detail-Architecture	Information-Process Detail-Architecture	Machine-Process Detail-Architecture

Whole-of-enterprise scope

- three layers: Business, Integration (Common), Detail
- three columns: People, Information, Physical Assets

IT is only a subset (not even all of Information)

21-Apr-08

(c) Tom Graves / Tetradian 2008

From there, this gives us a simple flattened view of the scope for the service-oriented enterprise.

This shows us why IT-centric architecture can be such a problem: it only covers a small part of what's really needed. IT doesn't even cover the whole of information, because there's also all the people-based 'tacit' knowledge, central to knowledge-management.

TOGAF ADM for SOE: overview

- **SOE needs iterative build of architecture**
 - broader scope is too large for 'big bang' style
- **Preliminary Phase defines basic shell**
- **Phase A identifies scope for iteration**
 - drawn from 'business question' for iteration
- **Phases B-D identify architecture in scope**
- **Phases E-H monitor solution design etc**
 - as per existing ADM

But it's not hard to adapt the ADM to suit.

The 8.1 ADM assumes a 'big-bang' approach: do all the architecture at once. But for whole-of-enterprise, there's no way we can do that reliably, or in a realistic time-scale. We must revise the ADM to an iterative style.

So we change the Preliminary Phase slightly, to define the main skeleton of the framework we'll populate in later iterations.

The rest of the ADM cycle needs only minor changes to handle iteration. Each iteration has its own scope, which or may not be centred on IT.

TOGAF ADM for SOE: Phase A

- **Architecture iteration is driven by ‘business question’**
 - Horizontal: Optimisation etc
 - Top-down: Strategy etc
 - Bottom-up: Disaster-recovery etc
 - Spiral-out: ‘Pain-point’ resolution
- **Map the scope of business-question onto Zachman-style frame**

Each architecture cycle begins with a business-driver, a business-question that needs to be resolved. This is the focus for Phase A in an iterative ADM.

To understand the scope and interactions within SOE, it's useful to map these drivers onto a Zachman-style frame.

TOGAF for SOE: horizontal drivers

	What	How	Where	Who	When	Why
1: Overall scope						
2: Business						
3: Systems						
4: Design						
5: Implementation						
6: Operations						

Example: optimise systems, reduce redundancy

21-Apr-08

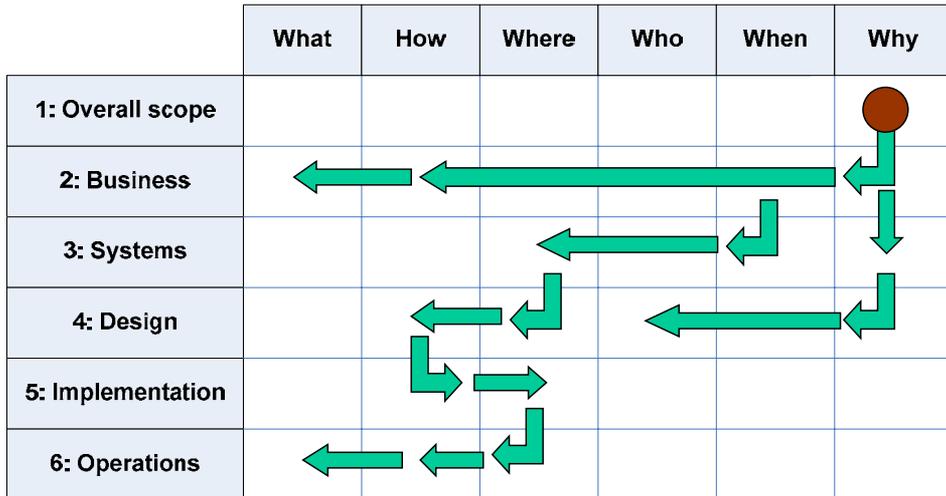
(c) Tom Graves / Tetradian 2008


TETRADIAN
the futures of business
22

Much of early-maturity IT-architecture is focussed on optimisation, reducing system-redundancy, or improving standardisation.

These are horizontal concerns. For example, in classic data-architecture, we optimise at the logical level, then move downward to physical designs.

TOGAF for SOE: top-down drivers



Example: impact of strategy, change of regulation

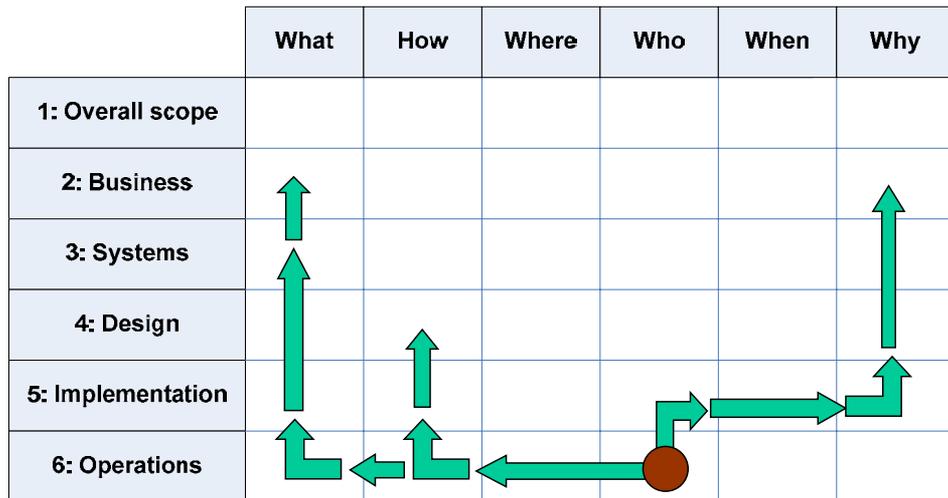
21-Apr-08

(c) Tom Graves / Tetradian 2008

With better optimisation, we're more able to tackle impacts of 'business-level' concerns such as changes to strategy, market or regulation.

These are top-down drivers. Their impacts start at the top of Zachman, and may well ripple downward all the way to the real-time Operations layer.

TOGAF for SOE: bottom-up drivers



Example: disaster-recovery, risk assessment

21-Apr-08

(c) Tom Graves / Tetradian 2008

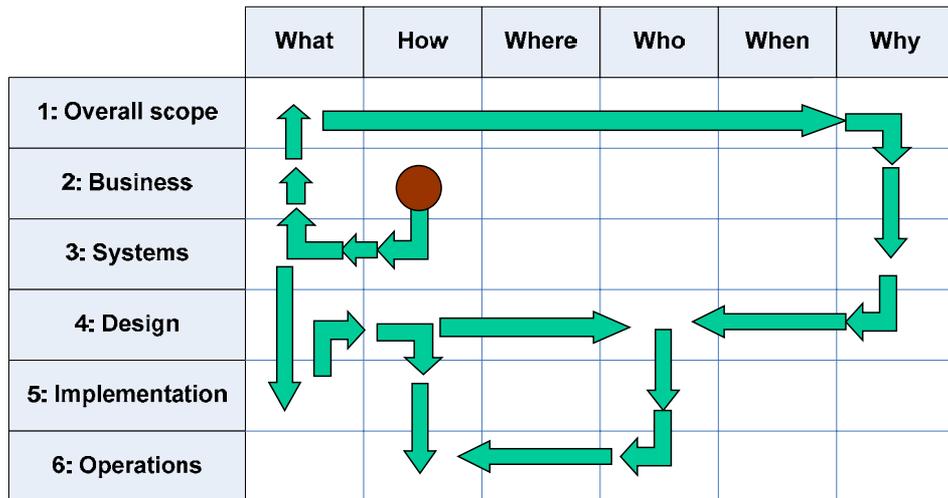
TETRADIAN
the futures of business
24

Bottom-up drivers start at the base of the Zachman frame, with impacts that can ripple upward all the way to the highest business-level metrics.

This is the province of disaster-recovery planning and failure analysis. The service-oriented enterprise makes this easier by identifying beforehand the interdependencies between systems, services and levels.

(It's easier again if there are links between architecture models and real-time tracking – a difficult technical challenge, but some of the EA toolsets can already do this.)

TOGAF for SOE: spiral-out drivers



Example: data quality, service-management planning



21-Apr-08

(c) Tom Graves / Tetradian 2008

What business *really* wants is architecture help with the difficult ‘pain-points’. One we dealt with was a key business-metric they couldn’t trust because of unreliable sources or transforms.

Two other IT examples are ‘single sign-on’ and ‘single source of truth’ - both simple in theory, anything but simple in practice.

In Zachman, the impact spirals-out in almost any direction – which we won’t know at the start. An iterative approach is essential for this kind of work.

TOGAF for SOE: Zachman review

- **Zachman on framework cells**
 - primitive *within* cell, composite *across* cells
- **“Primitives guide architecture; composites guide solutions”**
 - need primitives which cover full SOE scope
- **‘Completeness’ of composites**
 - ‘complete’ composite crosses all columns
 - must be complete at ‘Operations’ layer (row-6)
 - usable to the extent it is complete; re-usable to the extent it is not complete

21-Apr-08

(c) Tom Graves / Tetradian 2008



We also need to rethink Zachman itself for whole-of-enterprise architecture.

Zachman talks about ‘primitives’ versus ‘composites’.

- primitives sit within a single framework cell, as the root for architecture redesigns.
- composites link across cells as re-usable ‘building blocks’ for *solution* design – such as patterns at the logical layer, or clusters of components at implementation.

A composite can be used in a real solution to the extent it’s architecturally ‘complete’, linking across *all* columns.

It’s *re-usable* to the extent it’s *not* ‘complete’ – such as a business-service to re-use for different events or in different locations.

TOGAF for SOE: Zachman scope

- **In its standard form, Zachman is too narrow in scope for SOE (too IT-centric)**
 - limited even for IT: where are servers, UIs etc?
 - is a small subset masquerading as the whole
- **Layers are mostly usable as-is for SOE**
 - needs an additional 'Universals' layer
- **but Columns need significant updates**
 - muddled mixture of primitives and composites
 - need extra dimension for sub-categories

21-Apr-08

(c) Tom Graves / Tetradian 2008



But standard Zachman doesn't work well for SOE. It's too narrow, *far* too IT-centric; and though his layers are almost right, his columns are not.

His columns are a muddled mixture of misplaced primitives, and composites pretending to be primitives.

Some toolset-vendors make things worse by trying to cram all models into single cells – missing Zachman's own crucial distinction between primitives and composites.

To make the ADM work well for SOE, we *must* resolve these.

TOGAF for SOE: Zachman layers

Needs extra layer for
ISO-9000:2000

	Changes over time?	Needs relationships?	Needs attributes?	Implementation-specific?	Actual implementation?	Derived from real-time?
0: Universals (core constants)						
1: Overall scope (context model)	☑					
2: Business view (conceptual model)	☑	☑				
3: Systems view (logical model)	☑	☑	☑			
4: Design view (physical model)	☑	☑	☑	☑		
5: Implementation view (detailed representation)	☑	☑	☑	☑	☑	
6: Real-time view (live operations)	☑	☑	☑	☑	☑	☑

21-Apr-08

(c) Tom Graves / Tetradian 2008

Zachman's structure needs an extra 'Universals' layer at the top.

We need this to match with quality-frameworks like ISO9000:2000.

The new layer straddles *all* columns. Ultimately *everything* in the enterprise links back to entities in this layer – hence 'universals'.

TOGAF for SOE: Zachman columns

Columns need restructure to support SOE

<i>(original)</i>	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>
-------------------	-------------	------------	--------------	------------	-------------	------------

<i>(revised)</i>	Asset	Function	Location	Capability	Event	Reason
<i>(example segment)</i>	Object Information Relationship Value	Mechanical IT-based Manual Abstract	Physical Virtual Relational Temporal	Rules Analysis Heuristic Principle	Physical Virtual Relational Temporal	Rules Analysis Heuristic Principle

At Operations level, we should be able to describe every service as:

<i>with</i>	<i>do</i>	<i>at</i>	<i>using</i>	<i>on</i>	<i>because</i>
<asset>	<function>	<location>	<capability>	<event>	<reason>

21-Apr-08

(c) Tom Graves / Tetradian 2008



We do need to rethink the columns, to resolve the ‘pseudo-primitives’ in Zachman’s original, and to give flexibility for a broader scope.

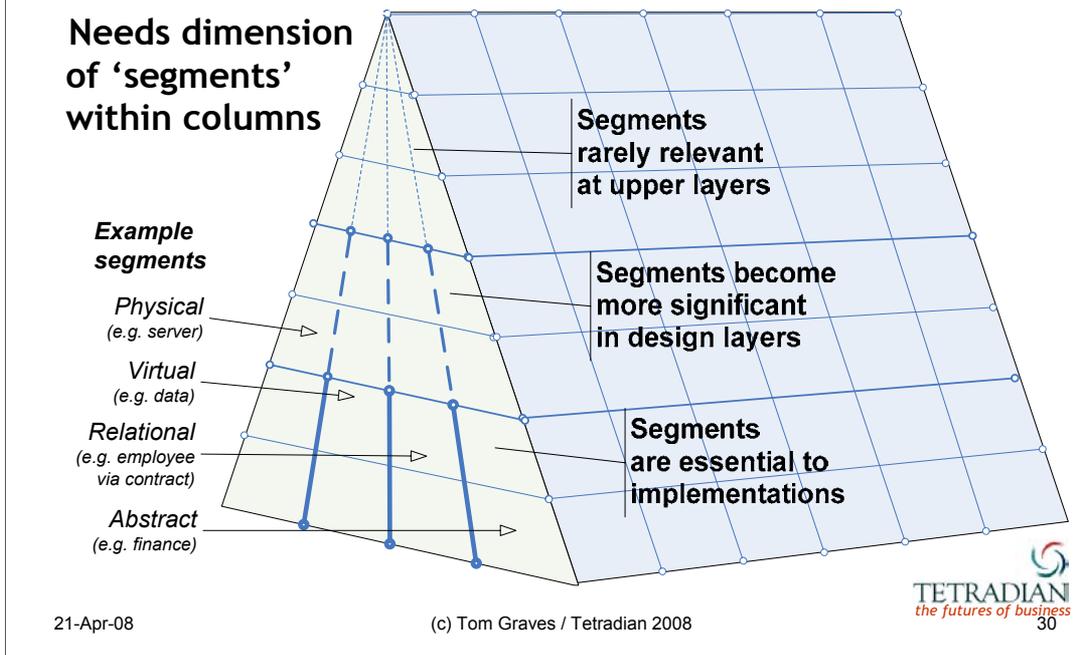
The architecture must be able to handle a human-services context or a chemical-plant as much as for IT.

This is a summary of the columns I use in my own work.

(There’s more detail on this in my *Bridging the Silos* book.)

At the Operations layer, we must be able to describe every ‘complete’ service in terms of that phrase in the slide: “with <asset> do <function>...” and so on.

TOGAF for SOE: Zachman segments



Standard Zachman is missing an entire *dimension*. Especially at the lower levels, we need to split the columns into distinct segments.

There are several variations for segments. The simplest is to split into segments on the same boundaries as in FEAF: physical ('Fixed Assets'), information, people-based ('Human Capital') and abstract ('Business').

(Again, there's more detail on this in the *Silos* book.)

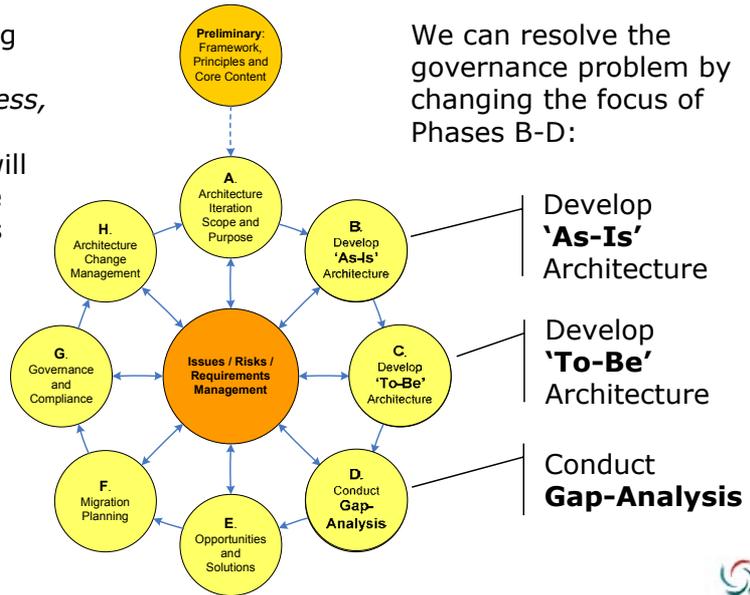
We can often ignore segments at the higher framework layers, we'll need them at design-levels.

At the base, the split is essential – in the real world we *need* to know if a service is provided by a machine, a person or an IT-box.

TOGAF ADM for SOE: Phase B-D [1]

If we retain existing Phase B-D focus – extended as *Business, Common/Shared, Detail-level* – we will have a governance problem of endless review-meetings

We can resolve the governance problem by changing the focus of Phases B-D:



21-Apr-08

(c) Tom Graves / Tetradian 2008

TETRADIAN
the futures of business
31

We define Phase A scope in terms of layers, columns, and segments. We also need to rethink ADM Phases B to D, because at present they assume a fixed IT-centric scope.

We could do it as in the existing ADM, with phase-boundaries on Zachman-type layers: business or strategy, logical shared-interfaces, and implementation-detail.

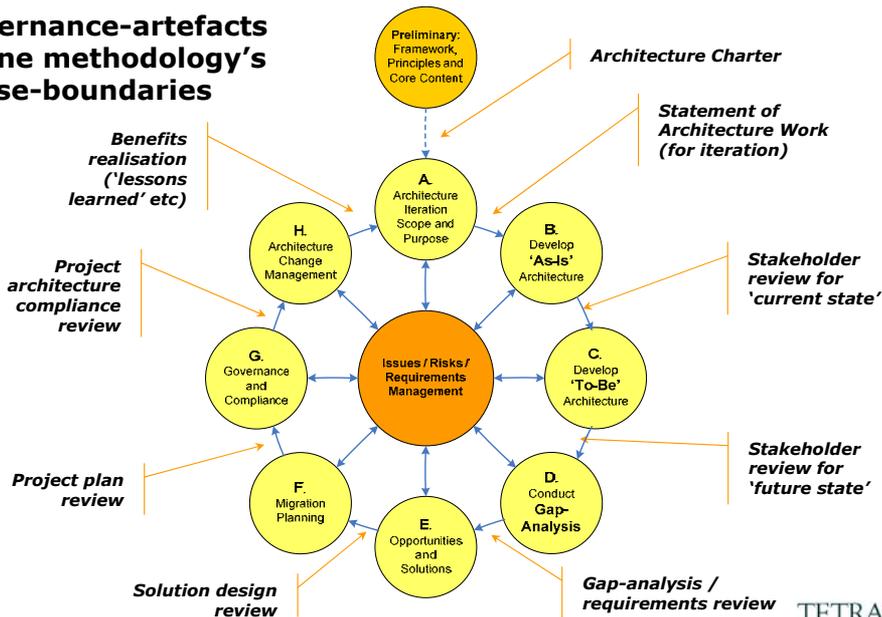
This needs as-is, to-be and gap-analysis in each layer.

But like the ADM, that means endless stakeholder-reviews. Which means trouble in a real business context.

A better split is to set Phase B as the *whole* of the 'as-is'; Phase C as the 'to-be'; and Phase D as the gap-analysis for later solution-design.

TOGAF ADM for SOE: Phase B-D [2]

Governance-artefacts define methodology's phase-boundaries



21-Apr-08

(c) Tom Graves / Tetradian 2008

TETRADIAN
the futures of business
32

In this split, the stakeholder-reviews occur only at phase-boundaries – in fact they are the phase-boundaries.

It's simpler than the existing ADM; it fits well with an iterative approach; and also fits well with the 'product'-based governance in PRINCE2 and ITIL.

TOGAF ADM for SOE: Phase E-H

Will be largely unchanged from existing
8.1 'Enterprise' standard

- **Phase E** will often need broader scope
- **Phase F** needs cross-enterprise governance
- **Phase G** needs stronger emphasis on whole-of-enterprise synergy
- **Phase H** needs emphasis on 'lessons learned', and to drive new iterations

The solution-design phases in the ADM - Phases E to H - need only minor adjustments to work well for iterative SOE.

(Again, I can only do a summary here, but the detail is in the *Silos* book.)

Implications of SOE

- **Enterprise-architecture is literally the architecture of the enterprise**
 - IT-architecture is only one subset of real EA
- **EA should not be under IT control**
 - it belongs under e.g. enterprise-wide PMO
 - IT-architecture can be controlled by IT under EA
- **Cannot do SOE successfully whilst EA is under IT control**

21-Apr-08

(c) Tom Graves / Tetradian 2008



A final problem is that we need to rethink the role and governance of architecture itself.

Enterprise-architecture *means* enterprise-architecture – not just IT-architecture.

It *needs* to operate with a true *enterprise*-wide scope.

So the challenge is that though EA started out in IT, and right now is usually under IT control, it does not belong there.

Unfortunately, our experience in practice is that getting IT to let go of EA can be the hardest task of all.

But this has to change for EA to move up to the next level of maturity, in support of the service-oriented enterprise.

SOE - a summary

- In SOE, *everything* is a service
 - even products are ‘services’, or enable them
- SOE covers whole of enterprise
 - an IT-centric or IT-only focus will cripple SOE
- SOE as web of service-interdependencies
 - can be mapped with VSM / xVSM
- TOGAF ADM is well suited for SOE
 - needs only minor amendments, as described

So, to recap, the basic principle of the service-oriented enterprise is that *everything* is a service.

This means that service-oriented architecture covers every aspect of the enterprise, not just its IT.

Systems-theory gives a useful tool to map service interdependencies, in the Viable System Model and the extended Viable Services Model.

And the TOGAF ADM, with only minor changes, is well suited to the iterative architecture that the service-oriented enterprise will need.

Further information

- **Viable System Model** (Wikipedia article and links)
 - http://en.wikipedia.org/wiki/Viable_System_Model
- ***Brain of the Firm***, 2nd Ed. (Viable System Model)
 - *author*: Stafford Beer; *publisher*: Wiley, 1994; ISBN 978-0-471-94839-1
- “**The Viable Services Model: service quality, service interdependence and service completeness**”
in ***ITSMF IT Service Management Global Best Practices 1***
 - *chapter*: Tom Graves; *publisher*: Van Haren, 2008; ISBN 978-90-8753-100-3
- ***Real Enterprise-Architecture: beyond IT to the whole enterprise***
 - *author*: Tom Graves; *publisher*: Tetradian, 2008; ISBN 978-1-906681-00-5
- ***Bridging the Silos: enterprise-architecture for IT architects***
 - *author*: Tom Graves; *publisher*: Tetradian, 2008; ISBN 978-906681-02-9

See tetradianbooks.com for more details.