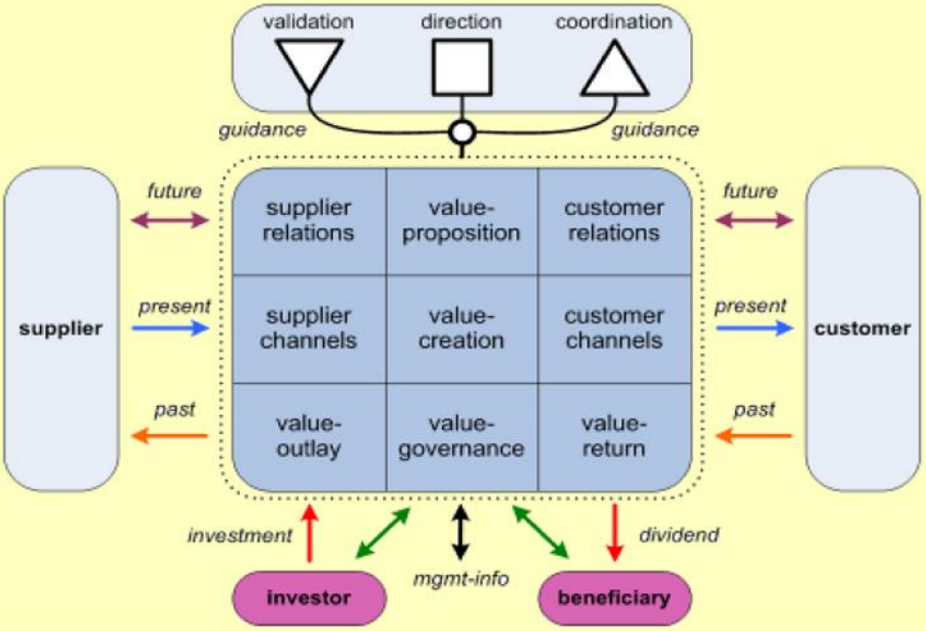
Enterprise Canvas

a framework for modelling the enterprise as services - *the Tetradian weblogs*



Tom Graves

Tetradian Weblogs Series

Enterprise Canvas

The Tetradian weblogs

Tom Graves

This book is for sale at http://leanpub.com/tp-ecanvas

This version was published on 2014-02-02

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Introduction

This ebook is an edited collection of posts from the Tetradian weblog[1] on the **Enterprise Canvas**[2] model-type and related modelling methods. This current edition covers the period from July 2010 to January 2014.

Enterprise Canvas is designed for use in service-oriented modelling for enterprise-architecture. It can be used to describe *any* aspect of the enterprise, providing a consistent, unified view all the way from strategy to execution. The full specification is in my book *Mapping the Enterprise*[3], and there's also a free two-page summary-sheet on the Tetradian Books website[4] in PDF format.

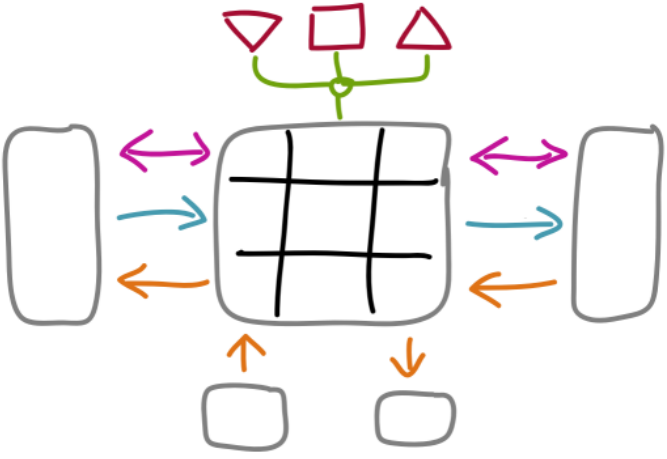
The real purpose of Enterprise Canvas is to provide a common frame across the whole of the enterprise-architecture space. To do this, it incorporates and unifies themes from a wide variety of other model-types in common use in EA, such as Zachman[5], Archimate[6], Business Model Can-

[1] http://weblog.tetradian.com
[2] http://weblog.tetradian.com/tag/enterprise-canvas/
[3] http://tetradianbooks.com/2010/11/ecanvas/
[4] http://tetradianbooks.com/2010/12/ecanvas-summary/
[5] http://en.wikipedia.org/wiki/Zachman_Framework
[6] http://en.wikipedia.org/wiki/ArchiMate

vas[7], VPEC-T[8], Viable System Model[9], Market Model[10] and many others.

Because Enterprise Canvas links all these different model-types together in a unified way, around a single core concept of values and services, it provides a powerful and versatile means to model what's going on in an enterprise, or what we need to happen or to change within that enterprise. Although there's a lot to Enterprise Canvas, and some of it might seem unfamiliar at first, in essence it always comes down to one core idea or entity that we can use or view in different ways, according to what we need at the time.

[7] http://en.wikipedia.org/wiki/Business_Model_Canvas
[8] http://en.wikipedia.org/wiki/VPEC-T
[9] http://en.wikipedia.org/wiki/Viable_System_Model
[10] http://www.slideshare.net/tetradian/what-is-an-enterprise



As in that example above, it's simple enough to be used in freeform 'back-of-the-napkin' sketches, yet it also supports the kind of formal rigour needed for structured diagrams, information-repositories and automatable simulations. And although it's simpler and easier to use than most of the common EA notations, it's also compatible enough with them not only to link to such models, but to use essentially the same notations within existing EA toolsets.

That's what this ebook describes.

The articles themselves are in mostly chronological order, though I've moved a few articles around were it would help the overall sense and flow.

The first set of posts, from July 2010, present the initial version of Enterprise Canvas and the ideas behind it. This set begins with **The Quest for the Enterprise Canvas**,

a somewhat jesting 'taster' that introduces the basic idea of a need for "one model to rule them all" across the enterprise space-modelling space. **Part 1: Context and Value** outlines the core concepts of values and services, whilst **Part 2: Market and Supply-Chain** places this within the overall enterprise context, and **Part 3: Owners and Managers** links it to a more immediate business scope. **Part 4: Layers** and **Part 5: Recursion, Flows and Systems** describe some of the internal structures of the model-type; **Part 6: Models** summarises the modelling-process, whilst **Part 7: Patterns** outlines four stereotypical patterns that can be found in different types of enterprises; and **Part 8: Integration** shows how all of this can cross-link to various other well-known model-types and frameworks. As the article-titles suggest, **Summary and Index** and **The Enterprise Canvas: a Really Simple Summary** provide a couple of different summaries, the former essentially text-based, the latter more visual. A somewhat later post, **An Enterprise Canvas update: 'value-governance'** provides a key name-change for one of the model-segments, from 'value-management' to 'value-governance'. And finally, around a year later, the post **Metaframeworks in practice, Part 5: Enterprise Canvas** gives a more detailed summary of where all the ideas for Enterprise Canvas came from, and how they came together.

On towards practice. As explained in **Context-space mapping with the Enterprise Canvas**, modelling with Enterprise Canvas is, in essence, a variant of a sensemaking technique called 'context-space mapping', as described in

my book '*Everyday Enterprise-Architecture*[11]'. This brief series on 'Context-space mapping with Enterprise Canvas' continues with **Part 2: Business context**, which explores the role of layering in Enterprise Canvas models; **Part 3: Value-proposition**, which shows the links to enterprise vision and values; **Part 4: Rethinking vision bottom-up**, which illustrates the previous ideas with some half-dozen real-world business-examples; and **Part 5: Service content**, which outlines what and how to model the active content within each service. There's also **Architecture: To-be is top-down, as-is is bottom-up**, that uses the row-mappings for Enterprise Canvas to illustrate 'top-down' versus 'bottom-up' aspects of enterprise-architecture.

Progress on Enterprise Canvas summarises the 'state of play' at that time (November 2010), shortly before the Enterprise Canvas book *Mapping the Enterprise* first hit the streets.

Enterprise Canvas uses the VPEC-T framework (Values, Policies, Event Content, Trust) in two different ways: as a means to model inter-service relationships, and, in non-standard form, to describe the sequencing of interactions within those relationships. **Not quite VPEC-T** and **More on 'Not-quite VPEC-T'** describe this second non-standard usage of the framework. It's then explored again, a couple of years later, in the post **Identifying meaning in context: VPECS and VPEC-T**, which finally brings a more explicit separation between the two frameworks and their respec-

[11] http://tetradianbooks.com/2010/05/everydayea/

tive roles.

A constant frustration in enterprise-architecture at present is the inadequacy of current modelling-tools in relation to what we actually need for our work. The next set of posts explore different aspects of this theme: **Enabling enterprise-architecture conversations** and **Models as decision-records (Enterprise Canvas)** outline some of the needs, whilst **Next-generation toolsets for enterprise-architecture?** and **The toolset-ecosystem** explore the broader scope. We'll return to this theme several times more in this collection of posts.

Much of current enterprise-architecture is so IT-centric that it all but ignores the true defining-feature of every enterprise: *people*. The posts **People, assets, relationships and responsibility** and **Modelling people in enterprise-architecture** describe how to include people in Enterprise Canvas models – and how and when and why not to do so – whilst **Power, people and enterprise-architecture** explores the contentious yet very important question of power in business-relations. And from rather later in the weblog-timeline, we have a post on **Positioning Enterprise Canvas in enterprise-architecture work**.

We next have a few posts that don't really fit into any obvious category. **Great conversations on enterprise-architecture** and **Notes on architecture versus design** describe some themes that came up in passing, which in part happen to touch on Enterprise Canvas; and **Rethinking the layers in enterprise-architecture** uses Enterprise

Canvas to help break out of the metaphoric box created by the IT-centric pseudo-layers (Business, Application/Data, Technology) that are still used far too often in enterprise-architecture. There's **CRUD, CRUDE and other action-acronyms** – a kind of rethink or extension of the classic Create, Read, Update, Delete. And finally a couple of rather technical posts, **Enterprise Canvas and the Service Cycle** and **Direction and governance flows in Enterprise Canvas.**

The next two posts – **Where marketing meets enterprise-architecture** and **Who is the customer?** – begin a section that looks more closely at business-architecture, the relationship between Enterprise Canvas and the popular Business Model Canvas model-type. This theme continues with **Using Business Model Canvas for non-profits**. We then have **From business-model to enterprise-architecture** and **Why business-model to enterprise-architecture?** which show how to use Enterprise Canvas to bridge 'downward' between Business Model Canvas and implementation-oriented model-types such as Archimate; whilst **Upwards and sideways from business-model** and the shorter summary-article **Upward and sideways from business-model (short version)** describe how to link Business Model Canvas 'upwards' to link with enterprise-values, and 'sideways' to investors and beneficiaries.

We then have a brief reprise on the toolset theme, with **More on that enterprise-architecture 'help needed'** and **Back to the roots for EA toolset metamodels**.

This links to the question of how build Enterprise Canvas models within the constraints of the existing toolsets. **Simplifying the Enterprise Canvas** and **More on simplified Enterprise Canvas** introduce a simplified notation that's compatible with UML, BPMN, Archimate and like; whilst **Enterprise Canvas as service-viability checklist** shows how this simplified notation can be used to map and verify the interrelationships that each service needs to underpin its own viability, and the viability of the enterprise as a whole.

The next brief set describe various modelling-themes. **Two kinds of Why** draws a useful distinction between 'question' and 'decision'; **Dependency and resilience in enterprise-architecture models** introduces a concept of 'distance' in dependency-modelling; and **Backbone and business-rules** describes a means to guide an appropriate balance between stability and agility in an enterprise-architecture.

If everything is a service, what then is the real role of management? The three posts **Management as 'just another service'** and **Rebalancing top-down management-architectures** and, rather later, **Managers, leaders and hierarchies** explore some of the architectural implications that arise from that potentially-explosive question…

Moving onto safer ground, **This: an exploratory game for service-oriented EA** and **More on the 'This' game for enterprise-architecture** introduce a new technique to elicit ideas and information for enterprise-architecture models; whilst **The 'This' game and EA toolsets** outlines

a set of scenarios on how this could be work in real-world-architecture practice – and how our toolsets could be improved to make this happen. **Using the 'This' game in EA modelling** describes a real modelling-session with the 'This' game, and practical lessons-learned that arose from that process.

Services manage and act on assets to create value for the enterprise. **Modelling mixed-value in Enterprise Canvas** explores specific aspects of asset/value relationships, whilst **Charisma, connection and brand** provides some clarity around the often little-understood aspirational-assets that are essential to enterprise function. **More on EA and asset-types** is a single article on assets in Enterprise Canvas that originally appeared as a four-part series on the blog. **Assets and services** and **On function, capability and service** describe the relationships between assets and the functions and capabilities that act on them within the services of the enterprise.

There's then a small group of posts on services and their relation to other elements, starting with **The meaning of service**, and then **Service, function and capability (again)** and **Service, function and capability (an addendum)**.

We return to the theme of Enterprise Canvas and business-models – particularly in relation to the popular Business Model Canvas, from the book *Business Model Generation*, by Alex Osterwalder and others. We start with a key question that's often misunderstood: **What is a value-**

proposition? Then a reminder that, in reality, **Every organisation is 'for-profit'** - we first need a better understanding of what profit really *is*. Next, a three part-series on **Business Model Canvas beyond startups** - an overview, and then more detail on the customer-facing 'front-end', and the 'back-end' and supply-chain. We continue with **Business-models between for-profit and not-for-profit**, another exploration about business-model workarounds for the arbitrary 'for-profit' / 'not-for-profit' split. The last item in this section uses Enterprise Canvas to help explain the **Costs of acquisition, retention and deacquisition**.

Finally, **Yet more on 'No jobs for generalists' (Part 2)** – part of a longer series on the challenges around how to gain employment in enterprise-architecture – uses the service-content model in Enterprise Canvas to explain why the conventional concept of a 'job-description' is misleading for generalist-oriented work such as that of enterprise-architects, and what types of role-descriptions would better serve that need.

Enough for now, anyway: on to the posts themselves.

Contents for the Sample edition

This ebook is an edited collection of posts from the Tetradian weblog[12] on the **Enterprise Canvas**[13] model-type and related modelling methods. This current full edition of the ebook includes all the weblog-posts described in the Introduction. This sampler-edition contains the following posts:

[12] http://weblog.tetradian.com

[13] http://weblog.tetradian.com/tag/enterprise-canvas/

The Quest for the Enterprise Canvas

[There is, as usual, a serious point to this post. Yet a gentle note to Americans and other non-English speakers: much what follows is riddled with puns and other artefacts of that imaginary entity called 'British humour'. My apologies… You Have Been Warned etc…!]

From long ago, in the lost land of Rubb, there is tell of a lore-full legend: the Quest for the Enterprise Canvas. This strange saga begins with the following poem, herewith transcribed from the original Rubbish:

> Three maps for the bizarchs high upon the office sky
> Seven for the itarchs who hide below alone
> Nine for the sysarchs, doomed to sigh
> One for the entarch on his dark throne
> One map to rule them all, one map to find them
> One map to bring them all and with mad humour bind them
>
> *(with acknowledgements and apologies to JRR Tolkien…)*

It's true that much may have been Lost In Translation[14], yet it seems that at the time the island was populated by an odd tribe of overburdened architects, each with their own specialist views: business-architects, IT-architects, systems-architects, and so on. One small caste – the enterprise architects – were tasked with overall overview: the note about 'his dark throne' appears to reference their continual struggle against being 'in the dark' about what was going on in the enterprise. Hence to them was assigned the 'one map to rule them all' – the fabled Enterprise Canvas.

This miraculous model-type had for many years been scoffed at by scholars, dismissed as mere myth, an imaginary artefact from the dawn of time; yet we can now reveal that it did indeed exist. What is more, by careful analysis of cross-references in ancient manuscripts such as the *Zachman Codex*[15], the *Book of Bmgen*[16] and *The Chronicles of the Af Family*[17] (the mapmaker-brothers Togaf[18], Feaf[19], Peaf[20], Modaf[21], Dodaf[22] and their cousins Itil[23] and Iso[24]),

[14] http://www.amazon.co.uk/Lost-In-Translation-Nigel-Green/dp/0978921844

[15] http://en.wikipedia.org/wiki/Zachman_Framework

[16] http://www.businessmodelgeneration.com/

[17] http://en.wikipedia.org/wiki/Enterprise_architecture

[18] http://en.wikipedia.org/wiki/TOGAF

[19] http://en.wikipedia.org/wiki/Federal_Enterprise_Architecture_Framework

[20] http://www.pragmaticea.com/

[21] http://en.wikipedia.org/wiki/Modaf

[22] http://en.wikipedia.org/wiki/DODAF

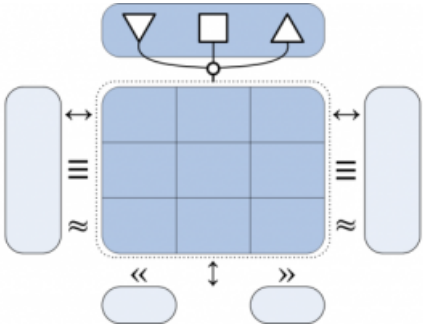
[23] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[24] http://en.wikipedia.org/wiki/ISO_9000

we now know what it once looked like, and what it could do. Some garbled hints of the shape and structure of this definitive diagram have occurred throughout history:

Not the Enterprise Canvas

Yet here, at last, is its true form:



The Enterprise Canvas

We know that this strange symbol was a way to describe services throughout the enterprise, each positioned at a cross-over of value and action. (Those of a religious disposition once interpreted this as a sign of the 'holy cross', but we now know that it was because people became Wholly Cross when things went wrong…).

Yet the full story of what it all means, and how to use it, will have to wait until the next post… Watch This Space?

Source (Tetradian weblog)

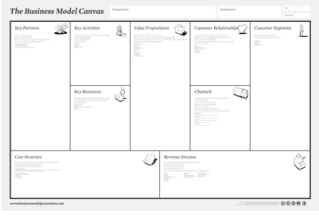
- *Date*: 2010/07/02
- *URL*: quest-for-enterprise-canvas[25]
- *Comments*: (minor)
- *Categories*: Business, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[25] http://weblog.tetradian.com/quest-for-enterprise-canvas

The Enterprise Canvas, Part 1: Context and Value

After that suitably silly start[26], it's time to be sensible and serious. Sort-of, anyway. :grin:

The real story is this: When I first saw Alex Osterwalder's Business Model Canvas[27], back some nine months or so ago, I was immediately struck by its elegant simplicity:



Business Model Canvas

In effect, it shows the supply-chain for a business-model, centred around the core value-proposition, all cross-linked to revenue-sources and cost-structure. In a word, it's brilliant. (If you don't already have a copy, get it! – or at least

[26] http://weblog.tetradian.com/2010/07/02/quest-for-enterprise-canvas/
[27] http://en.wikipedia.org/wiki/Business_Model_Canvas

download the generous 72-page sample from the book's website[28].)

The trade-off for that simplicity is that it only describes the business-model itself – not how the business-model is implemented, or where it sits within the overall enterprise. (In its present version it's also only suitable for for-profit organisations, not government or NGOs or the like, but that limitation is already being addressed[29] by Osterwalder and others.) So whilst it's excellent as a tool for business-architecture, it doesn't really cover enough scope for enterprise-architecture – either the usual IT-centric 'enterprise'-architecture or a true whole-of-enterprise architecture. Which is something we really need in EA. Which is frustrating. Oh well.

Most readers of this blog would know that I've done a lot of work over the past few years on models and methodologies for whole-of-enterprise architectures[30], yet I'd have to admit that none of them as yet have really had the immediacy of the Canvas. Most of those models, I'd also have to admit, have been too abstract to make much sense as yet to most everyday architects – especially as most so-called 'enterprise'-architectures are still centred entirely around IT, with a much narrower scope than we would need here.

Yet it struck me that there might be a way to link all those

[28] http://www.businessmodelgeneration.com/
[29] http://www.businessmodelsbeyondprofit.com/
[30] http://tetradianbooks.com

models together into a single structure that *could* cover the whole of the scope, from enterprise vision right to way down to the fine-detail of actual operations. The core of this came back to two themes that I've worked on for quite a while:

- the *enterprise* in scope for an enterprise-architecture is larger than the organisation – it needs to include the partners, suppliers, customers, prospects, market and overall 'ecosystem' as well as the organisation itself
- everything in the enterprise can be described in terms of *services* that support the aims of the enterprise – with the organisation itself as a supplier of services to the overall enterprise

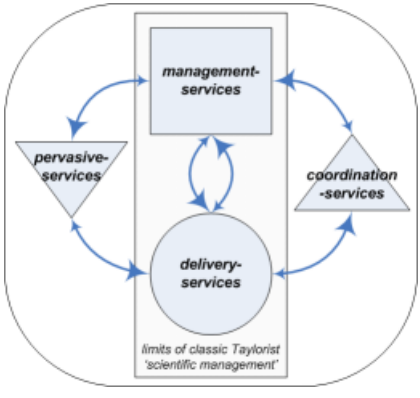
Following on from a more service-oriented view[31] of Stafford Beer's Viable System Model[32], we can identify four distinct categories of services:

- *delivery-services* – the 'profit-centre' services that contribute directly to the organisation's 'value-proposition'
- *management-services* that direct what the delivery-services should – the classic vertically-oriented 'management hierarchy' and suchlike

[31] http://tetradianbooks.com/2008/12/services/
[32] http://en.wikipedia.org/wiki/Viable_System_Model

- *coordination-services* that link horizontally across the silos to allow the supply-chains and the like to flow smoothly through the organisation
- *pervasive-services* that provide various forms of whole-of-system support – finance, HR, infrastructure – and that monitor and maintain quality throughout the organisation

Crucially, classic Taylorism[33] only acknowledges the first two types of services – one of the core reasons why it is so problematic in practice:



Viable Services Model

All services actually have the same structure, because all of them deliver a service: whatever way we might label or categorise them, in the end *every* service is a 'delivery-

[33] http://en.wikipedia.org/wiki/Scientific_management

service'. In effect, all that changes in each case is the value-proposition, and the content that the service delivers: the relationships within and between services remain much the same. This suggests that a service-oriented approach to enterprise-modelling might well give us a conceptual structure that's as simple as the Business Model Canvas, yet *can* cover the whole enterprise scope at any level of abstraction.

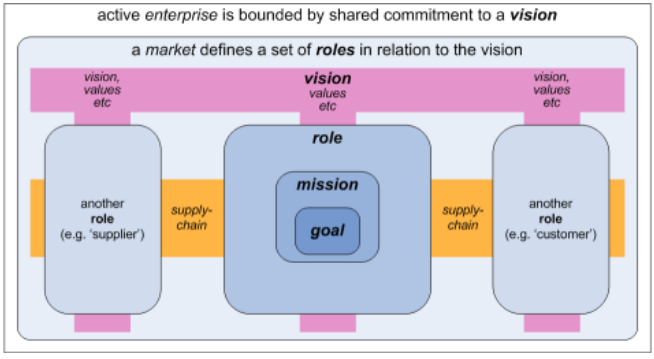
Next is the core distinction between *enterprise* and *organisation*. An organisation is also an enterprise, of course, but the two terms are not synonymous: following the definitions in FEAF[34], an organisation is primarily about legal and similar '*truth*'-type boundaries of roles, rules and responsibilities, whereas an enterprise is bounded more by *values*, by agreements and commitments:

> An enterprise includes interdependent resources – people, organisations and technology – who must coordinate their functions and share information in support of a common mission or set of related missions. … [I]t must be understood the enterprise may transcend established organisational boundaries – e.g. trade, grant management, financial management, logistics.

The actual core of an enterprise is its *vision*. Crucially, a functional vision for an enterprise is *not* a crude piece

[34] http://en.wikipedia.org/wiki/Federal_Enterprise_Architecture

of self-aggrandising puffery set out as a marketing-pitch – which is what is usually purported to be a 'vision' – but is a much simpler phrase or 'mantra' that summarises that enterprise's 'reason to be'. Good examples of valid visions include the Open Group[35]'s vision of '*boundaryless information flow*', or the vision for the TED conferences[36], '*ideas worth spreading*'. The key point here is that the vision is a stable, permanent, shared reference-point for *all* players in the enterprise – including the organisation for whom we're building an enterprise-architecture. We can summarise this in terms of the structure 'vision, role, mission, goal[37]':



Vision, **role**, **mission**, **goal**

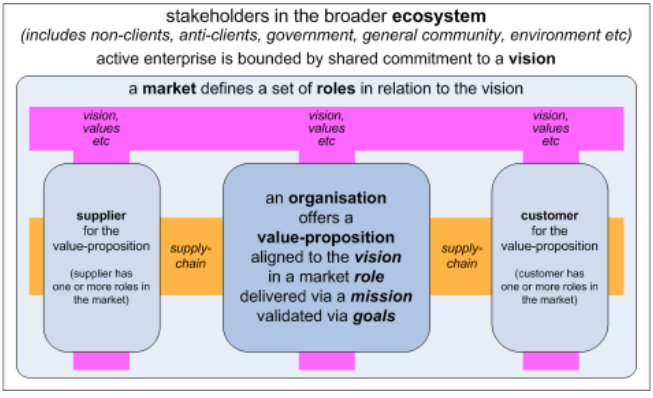
The enterprise vision and values are shared by all players,

[35] http://www.opengroup.org

[36] http://www.ted.com/

[37] http://www.slideshare.net/tetradian/vision-role-mission-goal-a-framework-for-business-motivation

which in turn is the reason *why* they are players in that specific enterprise. And each of those players presents some kind of 'value proposition' that aligns with the vision and that then links with other players in the enterprise, "in support of a common mission or set of related missions".

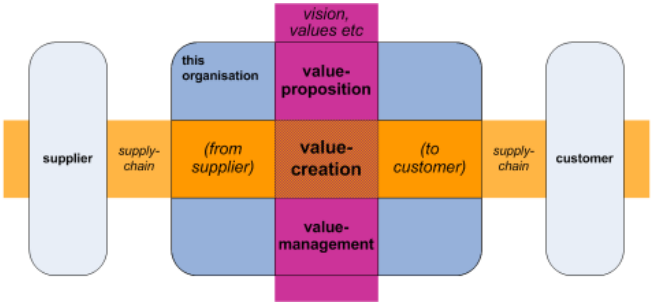


Market roles

We could explore further this question of 'what is an enterprise[38]', but the key point here is that the vision and values in effect traverse vertically through each player – each organisation – whereas the supply-chain or supply-web provides a kind of horizontal link between them. Each player thus exists at an intersection in a 'value-web', a point where organisation brings the vision and values of the enterprise in touch with respective roles of the supply-chain. Every organisation provides a *service* within its

[38] http://www.slideshare.net/tetradian/what-is-an-enterprise

chosen enterprise:



Value-web of services

Each organisation can be subdivided into sub-organisations, and sub-sub-organisations, and so on, almost ad-infinitum; and each of these 'units' provides a *service* that delivers *value* to the overall enterprise. Each service – organisation, sub-organisation, sub-sub-organisation, whatever – has its own emphases on value:

- *value-proposition*: what the organisation commits to be responsible for (what value it will add to the enterprise) [aligns with *quality*, with the *pervasive-services*]
- *value-creation*: how, where and by what means the organisation will add value to the enterprise [aligns with *action*, with the *coordination-services*]
- *value-management*: how the organisation will ensure that it can, has and will continue to add value to the enterprise [aligns with *information*, with the

management-services]

In a service-oriented architecture, *everything* is a service – so we'll see these three value-themes recur in *every* part and aspect of the organisation and enterprise. If so, it's clear that we have the beginning of an Enterprise Canvas there – a model that we can use in a consistent way to model *anything* in the enterprise. Yet that describes only the 'value'-axis of the value-web: we need to explore the supply-chain axis as well. That's what we'll look at in the next part of this series.

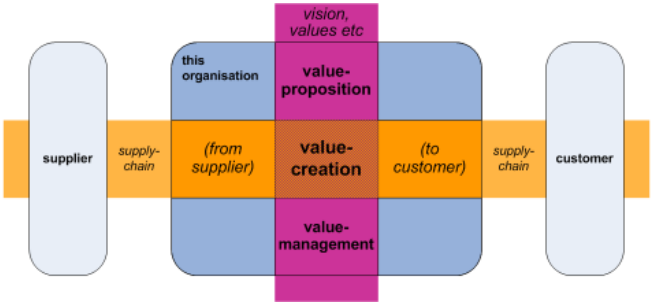
Source (Tetradian weblog)

- *Date*: 2010/07/02
- *URL*: enterprise-canvas-pt1[39]
- *Comments*: 2
- *Categories*: Business, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[39] http://weblog.tetradian.com/enterprise-canvas-pt1

The Enterprise Canvas, Part 2: Market and Supply-Chain

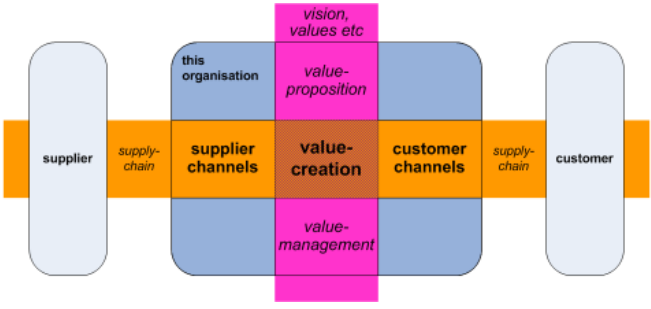
In the previous section[40] we looked at how the broader extended-enterprise is defined by its vision and values, how every part of the enterprise is made up of interdependent services, and how the enterprise vision and values pervade through every service. We ended up with this description-so-far of the structure of a service, which summarises that 'vertical' axis of *value*:



Enterprise Canvas: service value-web

We now need to link that to the 'horizontal' axis of the supply-chain:

[40] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/



Enterprise Canvas: service supply-web

To make sense of this, we first need to look in some depth at what actually happens in a market.

Even at a first glance it should be evident that **markets are transactions** – there's a lot going on, and on the surface at least, most of it is about deals, about supply-and-demand, about 'things', and those 'things' being exchanged. (And paid for, of course, in a commercial market – a point we'll need to come back to later.) This is the most visible part of the *content* of the supply-chain *channels*.

Yet as the *Cluetrain Manifesto*[41] famously observed, **markets are conversations**. Information is being exchanged in many different ways – sometimes as the content of the transaction itself, but some of it with little direct import for the transaction at hand. Sometimes a conversation is just a conversation – yet it's also evident that it's an essential component of the workings of the market.

And as Doc Searls, one of Cluetrain's authors, discovered

[41] http://www.cluetrain.com/

in an airplane conversation[42] with a Nigerian pastor named Sayo, **markets are relationships** too:

> [Sayo] went on to point out that, in his country, and in much of what we call the developing world, relationship is of paramount importance in public markets. Transaction still matters, of course. So does conversation. But the biggest wedge in the social pie of the public marketplace is relationship. Prices less set than found, and the context for finding prices is both conversation and relationship. In many cases, relationship is the primary concern, not price. The bottom line is not everything.
>
> Transaction rules the Industrialized world. Here prices are set by those who control the manufacturing, distribution and retail systems. Customers do have an influence on prices, but only in the form of aggregate demand. The rates at which they buy or don't buy something determines what price the "market" will bear — in a system where "market" means aggregated demand, manifested in prices paid and quantities sold. Here the whole economic system is viewed mostly through the prism of price, which is seen as the outcome of tug between supply and demand.

[42] http://www.linuxjournal.com/content/building-relationship-economy

> Price still matters in the developing world, Sayo said, but relationship matters more. It's a higher context with a higher set of values, many of which are trivialized or made invisible when viewed through the prism of price. Relationship is not reducible to price, even though it may influence price. Families and friends don't put prices on their relationships. (At least not consciously, and only at the risk of cheapening or losing a relationship.) Love, the most giving force in any relationship, is not about exchanging. It is not fungible. You don't expect a payback or a rate of return on the love you give your child, your wife or husband, your friends.
>
> Even in the industrialized world, relationship has an enormous bearing on the way markets work, Sayo said. But it is poorly understood in the developed world, where so much "comes down to the bottom line".

Finally, **markets are purpose** – as we saw in the previous section, the market is actually *defined* by the shared-purpose of the *enterprise vision*, the overall descriptor for its "common set of missions or goals".

So there are four distinct strands here, each in rough correspondence to one of the four distinct classes of assets:

- *physical* – alienable 'things' – if I give it to you, I no

longer have it

- *virtual* – non-alienable items such as information – if I give it to you, I still have it
- *relational* – a person-to-person connection that cannot be directly exchanged with or transferred to anyone else
- *aspirational* ('spiritual') – a personal connection to an idea such as a brand or enterprise-vision – again, cannot be *directly* exchanged with or transferred to anyone else

As Sayo says above, "Transaction rules the industrialized world". By that, we typically mean the transfer of 'exchangeable goods' – through what is most often described as the 'supply-channel' or *main-channel* – and, separately and in return, payment or its equivalent – through what we might describe as the *backchannel*. And in its simplest form, 'profit' is the difference between the *value-return* that comes in from the backchannel from the customer-side, compared to the *value-outlay* that goes out on the backchannel on the supplier-side. Once we also subtract the value-outlay for internal costs of running the service itself, the result is the 'bottom-line' net-profit that is the near-obsessive focus of so many business-folk.

Yet those transactions of 'exchangeable items' are only one part of the overall activities of the market; and all of those transactions – and the resultant profit – occur only at the tail-end of what we might call the *market-sequence*:

reputation > trust > respect > attention > conversation > transaction > exchange > profit

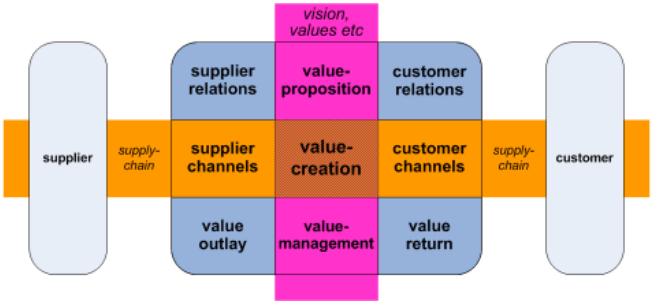
And the whole structure is a complex feedback-loop: for example, reputation – which itself is a kind of third-party precursor to trust – depends in part on whether the transactions are perceived overall as 'fair exchange'. Hence *before* any transactions can take place, the relationships need to exist: and those relationships – and conversations too – are strongly linked to and dependent on the service's value-proposition. Likewise *after* the transaction, there not only needs to be the backchannel exchange that completes the 'fair exchange' of the transaction, but that too also needs to link back to the value-proposition and the relationships, tying everything together in terms of the various definitions of what 'value' actually *is* within the overall enterprise.

So we have three distinct stages:

- what happens *before* transactions of 'exchangeable items' – an emphasis on *relations*, which link most strongly to the service's *value-proposition*
- what happens *during* (and with) transactions of 'exchangeable items' – an emphasis on *actions*, which link most strongly to the service's *value-creation*
- what happens *after* transactions of 'exchangeable items' – an emphasis on *value-transfers* (payments and receipts etc, as 'value-outlay' and 'value-return') and other follow-up and overall integration, which link most strongly to the service's *value-management*

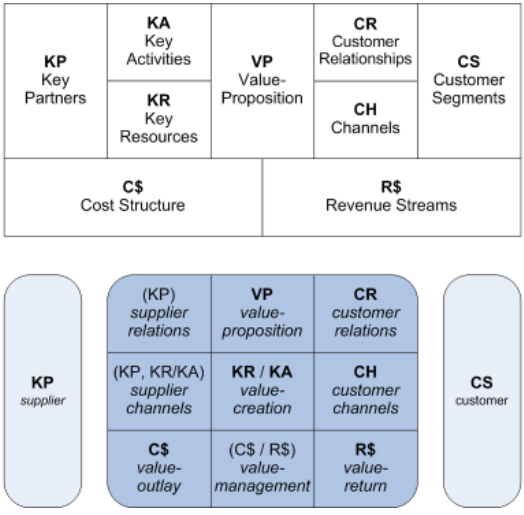
And all this needs to be symmetrical, in the sense that – within the overall market – this service that is our current focus is both a customer of other services, and a provider to other services, hence all of the above will apply in similar ways on both 'supply side' and 'customer side'. We need 'supplier relationship management', for example, just as much as we need 'customer relationship management'. And the value-transfers and follow-up need to be in terms of *all* forms of 'value' defined by the overall enterprise – and *not* measured solely in monetary terms, for example, no matter what the shareholders may demand!

When we put all of this together, what we end up with so far as a summary of the structure of *any* service – and the core of our Enterprise Canvas – is this:



Enterprise Canvas: service-cross

A comparison with the Business Model Canvas would be useful at this point:



Enterprise Canvas: comparison to Business Model Canvas

The Business Model Canvas [BMC] (upper) applies to a single context – the business-model for the overall organisation, at the 'logical' level of abstraction (relationships and attributes, as will be described in a later article in the series). The prototypic Enterprise Canvas [EC] (lower, cross-mapped to the BMC's cell-labels) applies to *any* type of service, at *any* level of abstraction. There are also a few differences in the detail: for example, I've deliberately bundled the equivalents of the BMC's 'Key Activities' and 'Key Resources' together into the EC's 'Value Creation' cell, for reasons that become clear when we do a Zachman-style analysis of what actually goes on in there. The BMC's 'Cost Structure' and 'Revenue Streams' are generalised in the

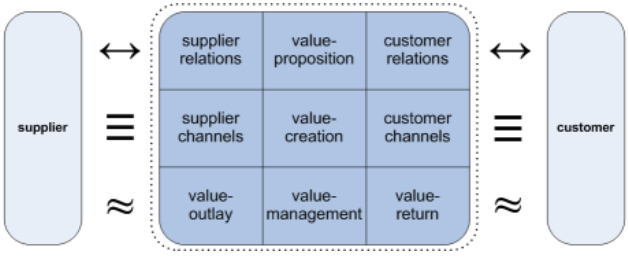
EC to cover *all* forms of value-exchange or value-transfer, not solely monetary exchanges. And the EC's symmetry of supply-side versus customer-side is absent in the BMC, which instead bundles several distinct types of activity into the one 'Key Partners' cell. But beyond that, they *are* quite similar – and need to be so, for the simple reason that both BMC and EC describe much the same entity, the activities of a service in delivering a value-proposition to the enterprise.

What's also *not* covered in the BMC, and that we do need to include in the EC, is the structure and content of the flows between services. It's useful to separate these out for the 'before', 'during' and 'after' phases of transaction, because they're often handled by different groups of people and/or different sub-services within the overall service. And in some cases – as with the service's interactions with its 'non-customer/supplier' stakeholders, such as government, non-clients, the local community or the general public – only one or two of the before/during/after categories of flows will apply. In addition, there are also some important differences in the emphasis of each flow:

- '*before*'-flows link primarily with supplier/customer relations, and must *always* be regarded as bidirectional
- '*during*'-flows link primarily with supplier/customer channels; whilst there are always some bidirectional components, the main flow is along the supply-chain (i.e. left to right in the EC diagram)

- '*after*'-flows link primarily with value-outlay/-return; whilst there are always some bidirectional components, the main flow is *opposite* to the supply-chain (i.e. right to left in the EC diagram) – hence 'backchannel'

We could typically illustrate these flows respectively as two people talking ('before'), a package ('during') and a small pile of coins ('after'). But since I couldn't find the right icons, I've used symbols from the keyboard character-set as follows:



Enterprise Canvas: supply-chain flows

All of these flows may be usefully described in terms of the VPEC-T[43] frame:

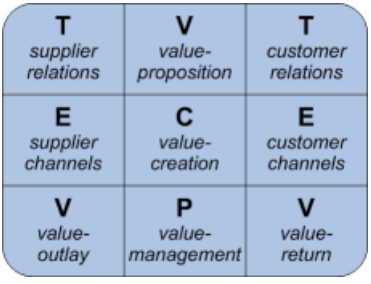
- *value/values*: that which *defines* the flow – the overall values that frame the context of the flow, and in which the quality of the results will be assessed

[43] http://en.wikipedia.org/wiki/VPEC-T

- *policies*: that which *directs* the flow – the decisions, rules and reasonings that will guide activities that would (in principle) add value to the content of the flow
- *events*: that which *triggers* the flow – the triggers that mark the initiation and/or completion of action on the content of the flow (and that themselves often contain information and other content as part of the event-trigger)
- *content*: that which *informs* the flow – the core 'content' of the flow, consisting of a context/role-specific mix of asset-types
- *trust*: that which *dominates* the flow – the feedback on the perceived value of the flow, and belief in future efficacy (effectiveness: efficient, reliable, elegant, appropriate, integrated) of equivalent future flows along this pathway

V (value) initiates the conditions in which the flow may take place; PEC (policy, event, content) represent the actual transaction, and actions taken subsequent to the transaction; T (trust) represents the feedback that adds to or subtracts from the expected V of further transactions of this type – and hence enables or disables future flows on this path.

Although all flows incorporate all of these elements, the respective VPEC-T emphases in each cell within the unit may be summarised as follows:



Enterprise Canvas: VPEC-T emphases

It's important to note that the 'value-creation' cell is the core of it all, and also – as we'll see later in the description of layering – is the only cell without direct external interfaces. More explanation on that in subsequent posts, anyway.

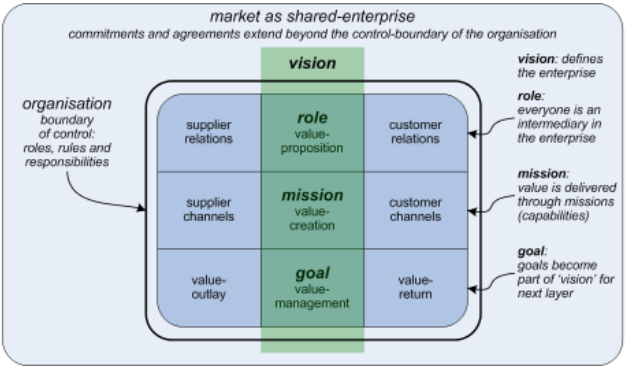
Source (Tetradian weblog)

- *Date*: 2010/07/03
- *URL*: enterprise-canvas-pt2[44]
- *Comments*: 4
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[44] http://weblog.tetradian.com/enterprise-canvas-pt2

The Enterprise Canvas, Part 3: Owners and Managers

In Part 1[45] of this series, we explored the 'vertical' axis of the Canvas: the vision and values that define the overall shared-enterprise, and the value-proposition that defines the reason-to-be for each organisation and service and sub-service:

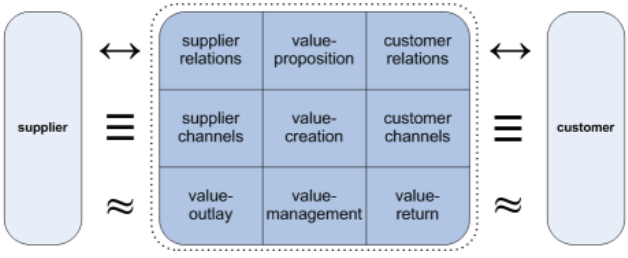


Enterprise Canvas: vision, role, mission, goal

In Part 2[46], we linked this with the 'horizontal' axis, the way

[45] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/
[46] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/

in which the supply-chain of added-value intersects with the value-proposition of each service. This also gave us a means to identify the flows that happen before, during and after the main transactions on the supply-chain, and what the service needs to do to handle each of these flows:

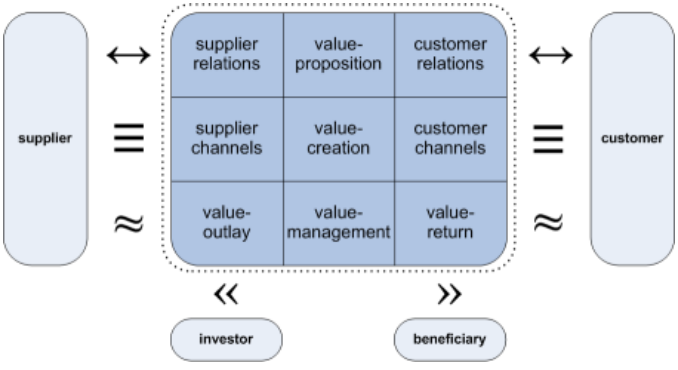


Enterprise Canvas: supply-chain flows

We need to explore how way in which this pattern recurs at various layers of abstraction. Before we do that, though, there are another couple of sets of flows that we will often need to include in our assessment. The first is the links with any external 'owners' – the external investors and beneficiaries:

- *investors* provide value (initial funds, etc) to start up and, if required, make up any shortfall in the value-outlay – in other words, to start up the outgoing backchannel and keep it going
- *beneficiaries* receive 'excess-value' (dividends, etc) from the value-return side of the backchannel – typically the 'net profit' after value-outlay and internal operating 'costs' have been subtracted

Although technically part of Value-Management, it's simplest to show these flows as linked to Value-Outlay and Value-Return respectively. (In the absence of appropriate icons, I've again used symbols from the character-set: left double-guillemet for investment, and right double-guillemet for dividend.)



Enterprise Canvas: investors and beneficiaries

From a simple monetary perspective, especially in a commercial context, the investors and beneficiaries will often be the same – the stockholders or shareholders, for example, or the members of a cooperative. In a not-for-profit or government context, though, they can often be different, at least in a money-only calculation: for example, donors or taxpayers as 'investors', and welfare-recipients as 'beneficiaries'. In that type of case, the imbalance is (mostly) intentional, but all such imbalances can lead to stress on the overall operating-model, especially if the flows are not balanced elsewhere in the system. Once we remember to

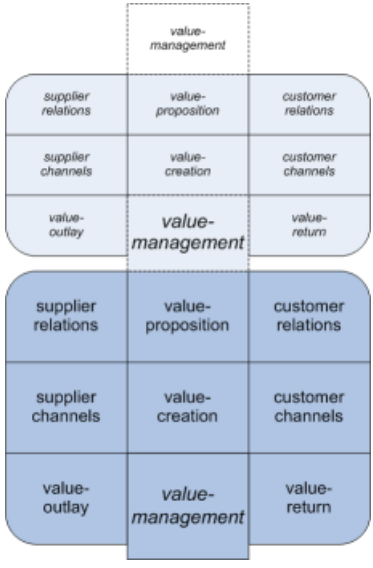
include other types of value beyond money, sometimes it's clear that the overall flow *is* balanced: charity-donors provide money, for example, but receive the satisfaction (i.e. as 'beneficiaries' of a non-monetary measure of value) that their money has gone to achieve the aims of the charity (i.e. aligned with the *vision* of the shared-enterprise).

But sometimes the imbalance can be unintentional, unresolved, and severe in impact: for example, where many people pour effort and ideas in a project, but only a single person or sub-group will gain the benefit – "take the credit" and so on. This type of *parasitic* investor/beneficiary model is invariably destructive to the enterprise, especially in the longer-term – and detailed value-analysis, especially when non-monetary value is taken into account, will show that it is disturbingly common within many organisations and enterprises, especially in the commercial context. The VPEC-T[47] frame of Values, Policies, Events, Content and Trust is again particularly useful for this analysis of the flows between the organisation and its 'owners'. These are major architectural concerns that the Enterprise Canvas can help to surface.

The other key type of flow here is the conventional information-flow that passes up and down through the management hierarchy: instructions and performance-criteria going 'down' the hierarchy-tree, and performance-information returning back 'up'. As indicated in the first diagram above, the Value-management cell provides part of the guidance

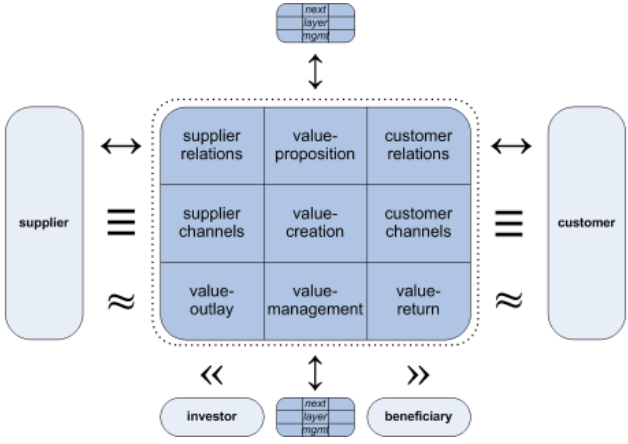
[47] http://en.wikipedia.org/wiki/VPEC-T

('direction') and, in effect, the 'vision' for the next layer down, and also provides bridge between the layers. Although the layering – as we'll see in the next article in this series – is actually about layers of abstraction, from overall architecture to detailed-implementation, that layering is reflected in part by the management-hierarchy, in which each layer of management moves steadily closer to the actual point of contact where value meets the supply-chain. We can indicate this on the Canvas by kind of extending Value-management outward as a connector to the next layer:



Enterprise Canvas: layer-hierarchy

In practice, though, this needs to be indicated by its own distinct flow, again typically assessed via the VPEC-T frame. As usual, there isn't an easily-available icon that seems appropriate for this, so I've used another key-character symbol, a vertical double-headed arrow:



Enterprise Canvas: layer-hierarchy with vertical-flows

Everything in the enterprise is like this: a service with its own value-proposition and means of creating value for the enterprise, its own customers and suppliers, its own 'owners' and 'managers', and all of the flows of things and information and relationships and trust that go with each of those. So it's clear that we are getting closer now to that single 'map to rule them all', a model that can be used as common base-map for all other types of models in the enterprise.

Yet there's one key theme we still haven't addressed: the *layering* of this structure as it applies in different ways throughout the enterprise. That's what we'll look at in the next article.

Source (Tetradian weblog)

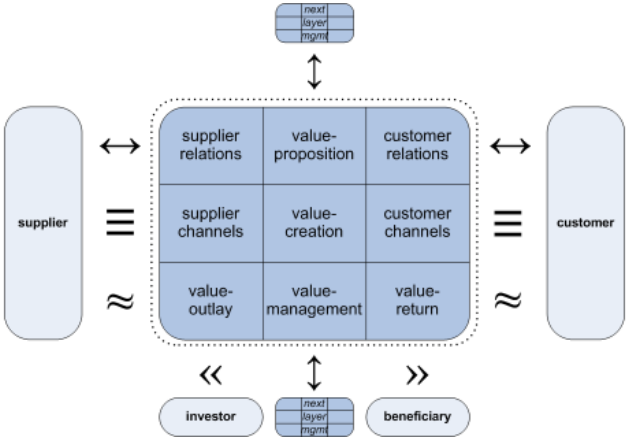
- *Date*: 2010/07/04
- *URL*: enterprise-canvas-pt3[48]
- *Comments*: 2
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[48] http://weblog.tetradian.com/enterprise-canvas-pt3

The Enterprise Canvas, Part 4: Layers

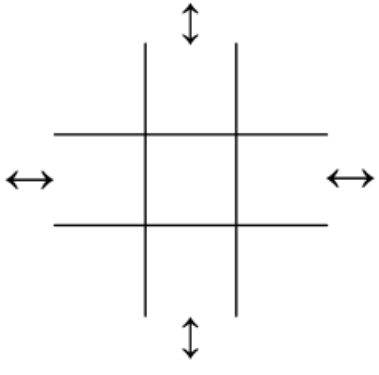
In the first part of this series of articles we explored context and values[49]; in the second part we linked this to the market and supply-chain[50], and the related flows that pass in the 'horizontal' channels between services; and in the third part we added the flows that link the service to its investors, beneficiaries and managers[51]. We ended up with a frame that looks like this:

[49] http://weblog.tomgraves.org/index.php/2010/07/02/enterprise-canvas-pt1/
[50] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[51] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/



Enterprise Canvas: layer-hierarchy with vertical-flows

...or, much simplified, but the same overall idea, the kind of diagram we would scribble on the back of a napkin:

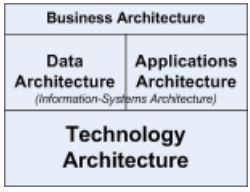


Enterprise Canvas: back-of-the-napkin

In the last article we started to explore the idea of *layering*,

as seen, for example, in the hierarchical trees of management reporting-relationships that so evidently link to the Value-management cell of each service. The problem there is that this isn't consistent: every organisation – and often every part of an organisation – will structure its management reporting-hierarchies differently from everyone else. And those management-structures keep changing almost on a day-to-day basis: we'll soon get ourselves into a right old tangle if we try to use them as a basis for an architecture of the whole enterprise.

So what else could we use? Well, there's the 'BDAT' layering – Business, Data, Applications, Technology – that's used in TOGAF and so many other 'enterprise'-architecture models:

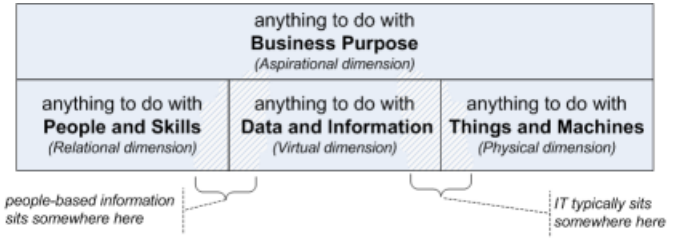


BDAT layers

Common though this layering may be, it's extremely misleading. In reality, it's only one tiny subset of what we need: it's a sort-of map of the enterprise in which the 'layers' are actually 'distance from self', where 'self' in this case is detail-level IT. This becomes clear when we look at the dependencies:

- assume that our nominal task is to specify the architecture for the *IT-infrastructure*
- to identify the IT-infrastructure that we'll need in the organisation, we first need to know the software-*applications* that will run on that infrastructure
- to identify the right applications, we need to understand the *data* that those applications will manage
- to identify the right data, we need to understand the business *information* that that data will underpin
- to identify the right information, we need to know the *business meaning* of that information
- to identify the business meaning, we need to know the *strategic drivers* of the business
- to identify the appropriate business-drivers, we need to know the nature of the enterprise – its *vision and values*

What happens in practice is that everything 'not-IT' gets bundled into a random, jumbled-up grab-bag labelled "business-architecture, which doesn't matter because it's not IT". Which, to say the least, is *not* helpful… because, as a few EA frameworks such as FEAF do sort-of acknowledge, what we *actually* have in the enterprise is something much more like this:



Actual layering

Every business-process exists for a purpose, and may consist of and/or be implemented by any combination of people, information/IT, things and machines. People-based 'manual' processes may be needed at any time to substitute for or take over from IT-based automation or machines: and to make that happen, they cannot be in different 'layers', but *they are at the exact same level.*

Once we understand that point, it becomes immediately clear that the standard BDAT stack is actually useless – or worse than useless – for anything than detail-level IT. And since IT is only a very small part of the overall enterprise, even in technology-intensive organisations – typically some 3-5% of costs and/or FTE – it's clear that we'll need something else with which to describe the required layering.

One IT-oriented 'enterprise'-architecture framework that *does* get this part right is Zachman[52]: what we need for this purpose are layers of *abstraction*, moving from abstract architecture to real-world implementation. Zachman

[52] http://www.zifa.com/framework.html

describes the overall organisation in terms of five distinct layers: scope, business-model, system-model, technology-model, and detailed-implementation. For our purposes, though, we need one more layer at either end of the scale, to place the organisation in context of the broader enterprise, and to distinguish what was supposed to happen ('detailed implementation') from what *actually* happened in real-world practice. These layers are distinct from each other because they add something more at each transition 'downward' in the stack. For compatibility with Zachman, it's best to start the layer-numbering at 0 (zero) rather than 1.

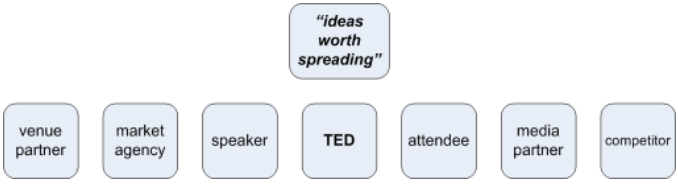
***Row-0*: Enterprise**. This topmost level consists of just one item – the enterprise as a whole, defined by its vision and values. In the ISO-9000[53] model for quality-systems, this likewise described as the *vision*, the root-anchor for the quality-system:

Enterprise Canvas: row-0 example

***Row-1*: Scope**. At this level every entity is just a member of a list, without relationships or attributes as such. Given that the 'parent' is the enterprise, this would be a list of relevant players (roles) in the enterprise, with the organisation

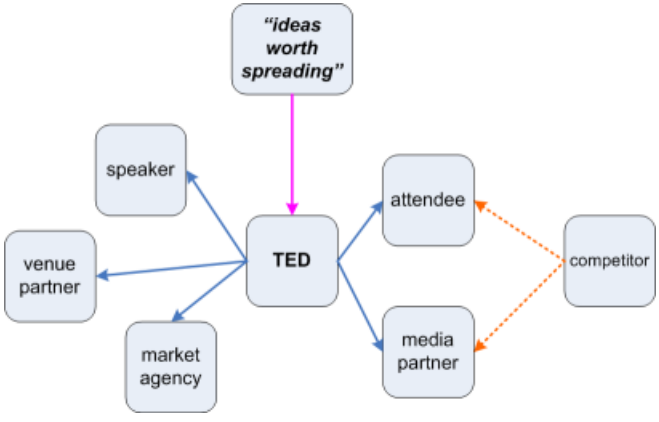
[53] http://en.wikipedia.org/wiki/ISO_9000

typically placed in the centre, and with all the other roles – customer, supplier, partner, competitor, regulator and so on – as other arbitrarily-labelled entities in the model:



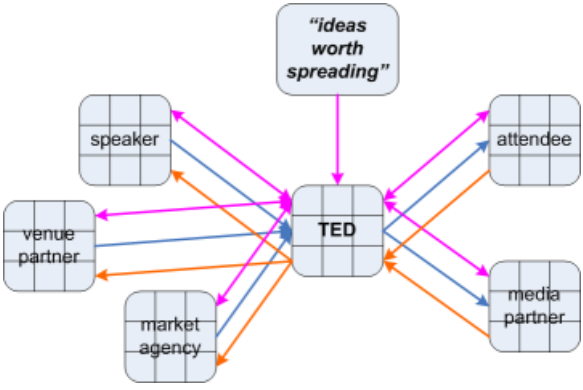
Enterprise Canvas: row-1 example

***Row-2*: Business model**. At this layer we add relationships between the roles – in other words, we start to add the horizontal 'supply-chain' dimension to the vertical dimension of shared-value. (In the ISO-9000 model, this is the *policy* layer.) Each of these relationships is determined by the respective *value-proposition* we can offer to and/or be offered by each of these other players:



Enterprise Canvas: row-2 example

***Row-3*: System model** (also known as *Logical model*). In this layer we start to include more details of attributes alongside the relationships, describing the internals of each high-level service and the flows between them. These are always abstract and generic: they define one or classes of possible implementations, but do *not* specify any particular technology or method. Most of what we would think of as 'enterprise architecture' or 'business architecture' will work at this level. (The Business Model Canvas sits part-way between row-2 and row-3: it summarises the relationships and value-propositions, but will rarely go into full formal detail of attributes and flows. In the ISO-9000 quality-system model, this is the *procedure* layer.) This is this first layer in which we begin to see the internal structure of each service, as described by the Enterprise Canvas:



Enterprise Canvas: row-3 example

***Row-4*: Design model** (Zachman 'Technology-model', also known as *Physical model*). In this layer we provide detailed specifications for technologies, processes, interfaces, flows, protocols, skillsets and the like, translating the abstract system-models into requirements and project-plans for detailed implementation. (In the ISO-9000 quality-system model, this is the generic component of the *work-instruction* layer: specifications for every instance of a particular implementation, rather than for one specific implementation.) On the surface, the model looks much the same as for the row-3 System-model: the difference is that it describes an *actual* intended solution rather than a whole class of possible implementations.

***Row-5*: Operations model** (Zachman 'Detailed implementation', also known as *Action plan*). This layer specifies the exact details of the full configuration of the service, as

scheduled to be used and/or enacted on a specific day at a specific location for a specific purpose with specific staff, and so on. Again, we could describe it with the Enterprise Canvas in the same way as for row-3 and row-4, but there would be a *lot* more detail embedded in the model. For example, this is the CMDB (Configuration Management Database), the staff-rosters, the production-schedule and such-like; where the row-4 design-model might specify a database-server, for example, here we need to know the exact identifiers, types, configurations, physical locations and virtual addresses of the production-servers, fallback-servers, development-servers, test-servers, and all related support-equipment and switchgear. (In ISO-9000 terms, this is the *specific* rather than generic version of the *work-instruction*.) For architecture, the main use of this level of detail is to provide an understanding of qualitative concerns such as availability, adaptability, variance in loads, and so on.

***Row-6*: Action record**. Unlike all the others, this layer is not a model of an intended future, but a record of the past. It carries the same level of detail as the row-5 action-plan, and in principle it should be exactly the same as row-5. In practice, though, it's rare that it *is* the same: rostered staff are absent for any number of reasons, a server breaks down, people get switched around onto different machines, half our customers are at home watching the football match, a traffic-hold-up forces a change of the delivery-schedule – all manner of deviations from the expectations set down in the plan. Although there's no

chance of changing anything – for the simple reason that it's all in the past – these records, and comparisons to the row-5 plans, are very important to process-designers, to organisational-learning specialists and many others who do extensive system-design (row-3 and row-4) in order to enhance operational effectiveness at the planning level (row-5). Enterprise architects can learn a great deal about real-world ebbs and flows here, about agility, resilience and emergent properties; and a solid understanding of the 'bottom-up' constraints evident here is an essential prerequisite for tackling the subtle yet serious pain-points and wicked-problems[54] that are likely to be embedded deep within the structures of the organisation.

So that's the key layering to use with the Enterprise Canvas: a structured sequence of transitions from most-abstract to most-concrete, with clear distinctions between each of these layers.

In the next article we'll explore another type of layering provided by a link to Stafford Beer's Viable System Model[55], which also acts as a precursor to the all-important topic of recursion. More later, anyway.

Source (Tetradian weblog)

[54] http://en.wikipedia.org/wiki/Wicked_problem
[55] http://en.wikipedia.org/wiki/Viable_System_Model

- *Date*: 2010/07/05
- *URL*: enterprise-canvas-pt4[56]
- *Comments*: 2
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[56] http://weblog.tetradian.com/enterprise-canvas-pt4

The Enterprise Canvas, Part 5: Recursion, Flows and Systems

So far in this exploration of the Enterprise Canvas[57] we've looked at context and values[58], market and supply-chain[59], owners and managers[60], and layering[61]. The next stage needs to take us for a brief wander through the wilds of systems-theory: specifically, on recursion, flow and overall integration.

The point here is that *everything* is a system, is part of a larger system, contains other systems; systems ad-infinitum. In dealing with anything on the scale of an enterprise, we *need* to be to able to think in terms of systems and systems-of-systems if we're to have any chance of making sense of what's going on. Yet whilst formal 'hard-systems theory' has gained a well-deserved reputation for near-incomprehensibility, most of what we need here can be summarised in five simple and straightforward principles: *rotation*, *reciprocation*, *resonance*, *recursion* and *reflexion*.

[57] http://weblog.tetradian.com/2010/07/02/quest-for-enterprise-canvas/
[58] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/
[59] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[60] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/
[61] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

Rotation is the simplest of the lot. Any real-world system is too large and too complex to be seen in its entirety from one view alone: to make sense of it all, we need to be able to *rotate* through a variety of different views and viewpoints, and merge those different views together in our minds. There's nothing difficult about this: every enterprise-architecture framework does it; every checklist does it. The whole point of the Enterprise Canvas is that it provides a checklist to give us many different views on the service in scope. We just have to remember to *use* it, *as* a checklist, *as* a systematic rotation through multiple views, that's all.

Reciprocation and **resonance** are a matched pair of principles that apply mainly to flows. In a simple world, flows need to balance somehow: "for every action there is an equal and opposite reaction", and suchlike. In that sense, flows need to be *reciprocal*; at the market level, there needs to be 'fair exchange', 'quid pro quo' – or at least a verifiable perception of that – and so on. Reciprocation is a large part of what we're exploring when we do a VPEC-T[62] analysis of a flow between services. The catch is that although things do have to balance up somehow, the flows may not be a simple 'tit for tat': often there will be delays in a feedback-loop, or translations from one form of energy to another – for example, donors to a not-for-profit don't get their money back, but do (we hope!) gain a different kind of satisfaction. In terms of point-to-point physics,

[62] http://en.wikipedia.org/wiki/VPEC-T

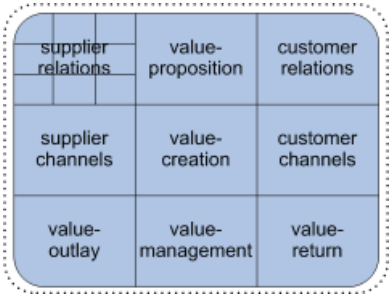
each transaction can be seen as a simple win/lose; but when we take the *resonance* of those feedback-loops into account, win/lose turns out to be a special-case within a full spectrum between win/win and lose/lose. That last point applies especially in social flows and social transactions – and frequently leads to 'unexpected' wicked-problems[63] if we're not aware of what's actually going on. To help us make sense of that, we can use the Enterprise Canvas to describe resonance and reciprocation in the interactions and flows across the whole system, especially over time.

Recursion and **reflexion** are another matched-pair. *Recursion* occurs when the same pattern repeats – or is 'self-similar' – at different levels of the same system. A conventional reporting-hierarchy is recursive: the same pattern repeats at different levels of management. A service-oriented architecture is recursive: the same concept of services, the same *pattern* of services, applies in much the same ways at every level of the enterprise. *Reflexion* is the inverse of this: if recursion means that every part in some ways reflects the whole, then we can also infer the whole from within every part. (Or *some* aspects or themes of that whole, at any rate: the fact that most things are not identical but only 'self-similar', always somewhat unique, means that the view we get from within any one part will always be somewhat blurry, somewhat incomplete. It's like a holograph: every discrete part of the image also contains *all* of the image, but in less detail – and that loss of detail can sometimes be

[63] http://en.wikipedia.org/wiki/Wicked_problem

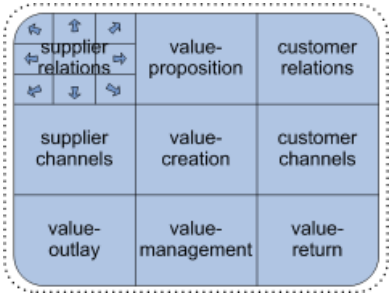
misleading if we extrapolate anything out too literally.)

Within the Enterprise Canvas, one of the most immediate examples of recursion is that every cell within the Canvas is also itself a service. So the Supplier-Relations cell, for example, not only provides 'supplier-relations' services on behalf of the service that we're depicting on the Canvas, but to do that it also has its own Value-Proposition in presenting those services, it has its own Supplier-Channels and Customer-Channels (the latter rather than the former being how, in this case of Supplier-Relations, it actually delivers its services), it has its own Value-Creation in which it creates the value of those supplier-relations, and so on.



Enterprise Canvas: recursion of child-services

And each of those sub-cells too have their own flows, some of them linking externally on behalf of the main service, others linking to other cells and sub-cells *within* the service:



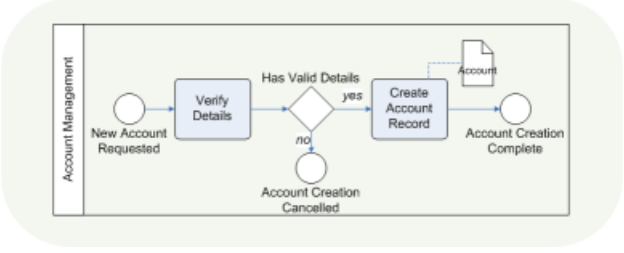
Enterprise Canvas: recursion of service-flows

The inverse is also true: whenever we come across a business entity or function that delivers what we might interpret as 'supplier-relations services', we need to ask who or what service it delivers these services *for*, and we need to look for and identify the matching Supplier-Channels, Value-Creation, Customer-Channels, Value-Outlay and the like that provide its 'sibling' services in terms of the Enterprise Canvas.

Typically, whenever we go 'down' a level into the recursion, we move closer towards real-world implementation; and whenever we go 'up' a level in the recursion, we're usually also moving up a layer of abstraction, as described in the previous article on layering[64]. In that sense, abstraction and recursion are closely linked within the Canvas – which is also another reason why the most useful framework for layering is that of abstraction rather than, say, management-hierarchy.

[64] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

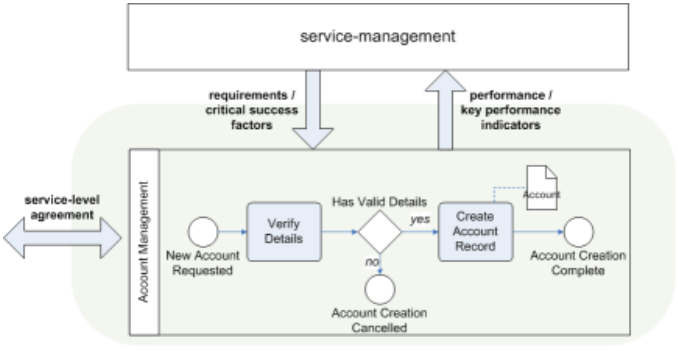
But the management-hierarchy does have a key part to play in another form of recursion that links well with the Enterprise Canvas. To create some background, consider a simple business-process that creates a new account – a service that we might well use in Customer-Relations or Supplier-Relations, as a precursor to any transactions happening in Supplier-Channels or Customer-Channels. At this point it's still a fairly abstract description – for example, we don't care whether it's implemented by a manual system or by IT – so from an Enterprise Canvas perspective, all of this is at the *row-3* layer or *System-model*, the most common area of emphasis for architectural assessment. Let's illustrate this with a (much-simplified) process-diagram in BPMN process-model notation:



Account-service: BPMN process

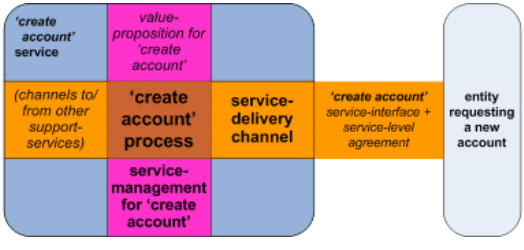
These diagrams are common enough in solution-architectures, but they actually tell us very little that's of real use in a *service*-oriented architecture. For example, at the very minimum, we need to know the interface flows between this process and its external 'provider' and 'client' processes; and we also need to know how it would be managed, what

performance-criteria it should report, and where those reports should be sent:



Account-service: service-context

Which, in terms of the simplified version of the Enterprise Canvas, looks like this:



Account-service: Enterprise Canvas relationships

(…and yes, each drawing faces the opposite way round: my apologies, it's just a notation-difference in the ways that BPMN and the Enterprise Canvas show the flow of the supply-chain – BPMN emphasises the sequence of events,

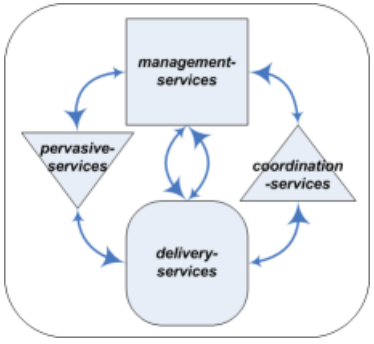
Enterprise Canvas the overall through-flow.)

Yet there's still a whole lot that's missing from this. For example, who or what manages the services-managers? Who or what coordinates this services with other services? Who or what ensures that quality is established and maintained? None of that is in here as such: and there's a real need for some forms of *guidance* to make sure that these things *do* happen.

It's at this point that we need to reprise part of where we started, back in Part 1, namely the idea that although in principle every service is a 'delivery service', there are three other key categories of service: management-services, coordination-services, and the 'pervasive services' that help to create and maintain quality throughout the enterprise. Adapting Stafford Beer's Viable System Model[65] [VSM] to a more service-oriented view of the enterprise[66], we can describe their relationships like this:

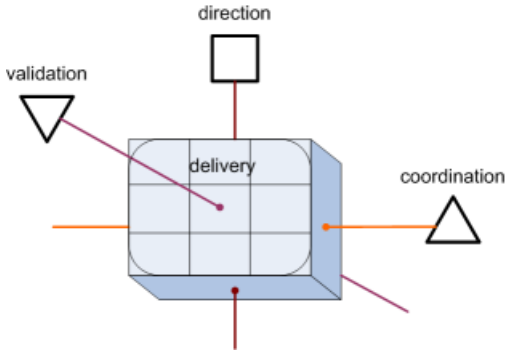
[65] http://en.wikipedia.org/wiki/Viable_System_Model

[66] http://tetradianbooks.com/2008/12/services/



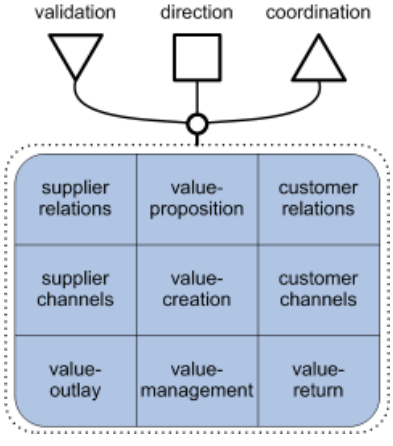
VSM: four service-categories

An alternate name for 'management-services' is *direction*, because their role is actually quite a bit broader than simple management of a business-unit or service. And another, perhaps more business-friendly name for 'pervasive-services' would be *validation*, because that's what they do – they ensure that everything holds to the values of the chosen shared-enterprise. The actual relationships of these other categories of services, though, are almost orthogonal to each other:



Four service-dimensions

Perhaps a simpler way to show this would be to link them to the frame of the overall 'delivery-service' – shown here as a dotted-line around the standard nine-cell Canvas:



Enterprise Canvas: four service-dimensions

Note that each of these attaches *at the respective layer or 'row'*, as described in Part 4: the same layering and recursion applies to these services too, from most-abstract (row-0) to most-concrete (row-6).

With the exception of some of the coordination-services, few of these services have much if any impact on the day-to-day running of most of the delivery-services. Their real role is to assist in the *dynamics* of those services – the ways in which they can, may and must change over time to adapt to changing context and to align more strongly with the enterprise vision.

Direction represents the management-services providing oversight of the direction and operation of the unit. In turn, these services are split into three distinct categories:

- *policy, purpose and identity*: long-term view to 'develop the business' for the unit [VSM 'system-5']
- *strategy and context* ('outside/future'): near-future view to 'change the business' for the unit [VSM 'system-4']
- *direction and tactics* ('inside/now'): immediate focus to 'run the business' for the unit [VSM 'system-3']

Within the nine-cell Enterprise Canvas frame, these connect most strongly with the Value-Management cell and, from there, to Value-Outlay and Value-Return. There's usually only one of set of these services attached to each 'delivery-service' unit [VSM 'system-1'], though one of

these will often be shared across and guide many sibling-units (as in a classic organisational-hierarchy).

Coordination represents the coordination-services [VSM 'system-2'] that link units together to create webs of cross-functional processes as required. As with the 'direction' services, these can be split into three distinct categories:

- *develop the business*: coordinate portfolios of longer-term change across units – also provides cross-functional bridge between direction's 'policy purpose and identity' and 'strategy and context'
- *change the business*: cross-functional coordination of change-projects – also provides cross-functional bridge between direction's 'strategy and context' and 'direction and tactics'
- *run the business*: cross-functional coordination of run-time processes – also provides cross-functional bridge between direction's 'direction and tactics' and the unit's own processes and interfaces

Within the nine-cell Canvas frame, these connect most strongly with Value-Creation and the 'supply-chain' interfaces – Supplier-Channels and Customer-Channels. There's often only one organisation-wide 'develop the business' strategy-coordination service, though a variable number of 'change the business' services, dependent on the organisation's portfolio/project-management mix. There will be a large number of 'run the business' links, often forming a

complete 'shadow network' that is almost invisible to the standard hierarchy.

Validation represents the broad range of 'support-services' that guide the organisation towards ever-stronger alignment with the enterprise vision and values. [In VSM this is 'system-3*', although in the original it only addresses financial and/or operational audit.] These services need to touch *every* part of the organisation, without exception, ultimately as part of the background ethos, culture and collective habits – hence the alternate label of 'pervasive-services'. Once again, these too can be split into three distinct categories:

- *develop awareness*: advertise and evangelise to create awareness of the importance of the respective values and their practical implications (principles etc)
- *develop capability*: educate in practices and metrics to implement and monitor compliance to the enterprise-values via their derived principles etc
- *verify and audit*: review records and lessons-learned to assure compliance to the values

Within the nine-cell Canvas frame, these connect most strongly with the Value-Proposition cell and, from there, the 'relations' cells – Supplier-Relations and Customer-Relations. In principle, there should be one matched set of these services, organisation-wide, for *each* key value of the enterprise (e.g. safety, security, quality, innovation,

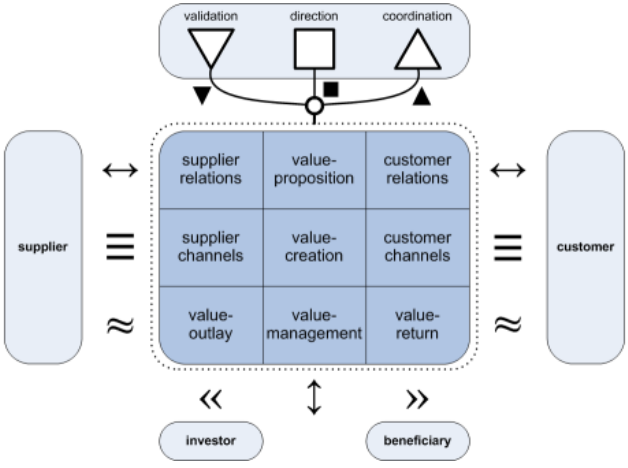
knowledge-sharing etc) – although the whole point of these services is that they reaffirm that responsibility for the respective value is ultimately *everyone*'s responsibility. The full set of services needed will be different for every enterprise, aligning with the different needs and different values of the respective enterprise. Note that for some values such as financial probity or occupational health and safety, various laws or functional constraints may mandate that the 'verify and audit' services should or must be kept separate from the respective 'develop awareness' and 'develop capability' services.

(For more detail on all of these guidance-services, and the inter-relationships between them, see my book *The Service-Oriented Enterprise: enterprise-architecture and viable services*[67].)

Each of these services has their own flows that pass between them and the service in focus in the Enterprise Canvas – which, once again, is easiest to show of the diagrams with a key-character symbol that matches the respective VSM shape (square for 'direction', downward- or upward-pointing triangle for 'validation' or 'coordination' respectively).

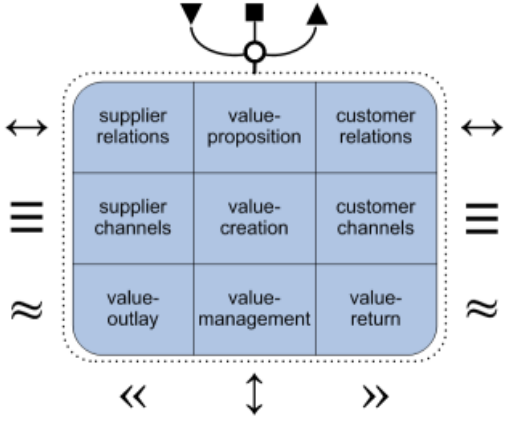
So we can now put all of this together, into a single diagram – the full Enterprise Canvas, complete with all of its flows and external relationships:

[67] http://tetradianbooks.com/2008/12/services/



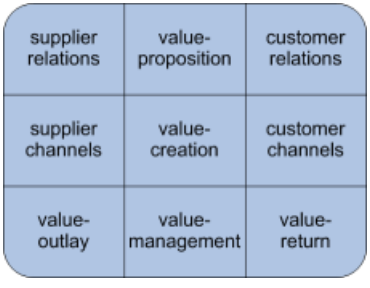
Enterprise Canvas: complete

Which, as one slightly unkind colleague pointed out, does bear an unfortunate resemblance to a *Lost In Space*-style Robbie the Robot… oh well… If in doubt – or in fear of mockery – you can always trim it back to just the flows:



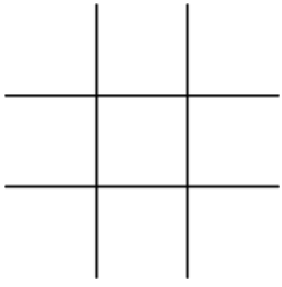
Enterprise Canvas: flows

Which I suppose could be said to look like some kind of beetle… also 'oh well'… Never mind, you could just use the standard nine-cell frame, without the flows:



Enterprise Canvas: core

Or, in extremis, the minimalist back-of-the-napkin version – which you could also use to play noughts-and-crosses (tic-tac-toe) if you wish:



Enterprise Canvas: minimal

So, that's it: the Enterprise Canvas – so-called because you can use it to model *any* type of services, at *any* level, *anywhere* in the enterprise.

And I did promise, way back in that light-hearted introduction[68], that this would be:

> One map to rule them all, one map to find them
> One map to bring them all and with mad humour bind them

Not quite sure now about the 'with mad humour' bit, but the remainder certainly *is* still true – and we'll start to look at how that works in the next part of this series.

Source (Tetradian weblog)

[68] http://weblog.tetradian.com/2010/07/02/quest-for-enterprise-canvas/

- *Date*: 2010/07/06
- *URL*: enterprise-canvas-pt5[69]
- *Comments*: (none)
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

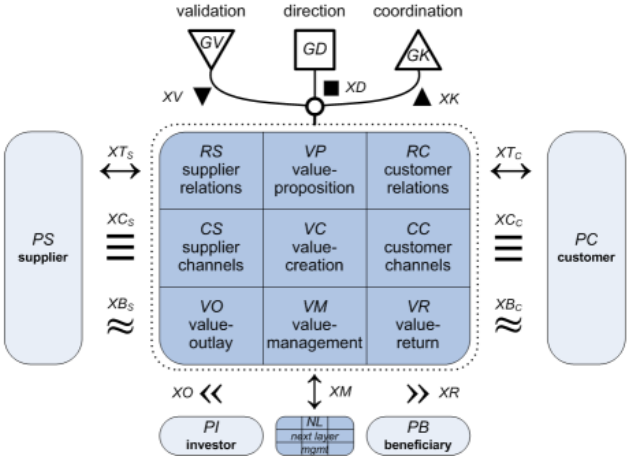
[69] http://weblog.tetradian.com/enterprise-canvas-pt5

The Enterprise Canvas, Part 6: Models

After that rather lengthy wander through context and value[70], market and supply-chain[71], owners and managers[72], layers[73] and recursion[74], we now have our complete Enterprise Canvas. Time to put this model-type to practical use.

There'll be a lot of cross-references in what follows, so for brevity's sake here's a version of the complete Canvas with simple two-letter codes assigned to each cell, flow and external party:

[70] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/
[71] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[72] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/
[73] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/
[74] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/



Enterprise Canvas: complete (coded)

The Canvas model assumes that the enterprise can be described in terms of – and hence consists of – any number of interrelated services, each of which can be described at any appropriate level of abstraction and granularity – see Part 4: Layers[75]. Each instance of a Canvas describes a single service at a single layer of abstraction, in context of the flows and interchanges it shares with each of its stakeholders.

We really do first need to know the enterprise vision and values, in order to know what 'value' *is* and means within any part of the enterprise. If we don't already have it, we need to start at row-0 to set that in place. (We can sometimes infer the content of the Enterprise layer from

[75] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

the various value-propositions on offer by the organisation, but architecturally it's a risky tactic, particularly when – as is all too often the case – the values and value-propositions are inconsistent with each other and will be based primarily on what happened in the past. Far better to do the extra work upfront to establish a stable foundation that will apply to past, present and future.) Once we *do* have that Enterprise layer in place, though, we can start anywhere, with any service, at any level, anywhere in the organisation and extended-enterprise.

The modelling process is exactly the same in every case, whether we want to do an as-is, an as-was or a to-be model for the respective service that is the focus of this Canvas.

The Canvas is also holographic: every service ultimately connects with everything else, hence any assessment we do in any part of the enterprise implicitly contributes to the detail of everywhere else. So we don't need to try to gather every possible piece of information about everything in 'excruciating detail', as John Zachman once put it; instead, we start anywhere that seems appropriate, model only what we need for that specific task, link the models together wherever practicable and appropriate, and allow the overall detail to emerge over time through the layering and interconnections of the Canvas structure.

So: start somewhere. Anywhere will do. It doesn't need to be a 'service' as such – just an area of interest, an aspect of the organisation or enterprise where something happens.

What **layer** is it, in terms of the Enterprise Canvas? To

put it another way round, how abstract is it? or how concrete? Use the descriptions in the layers[76] article to identify which layer you want to be working on. (The important point here: *don't mix layers*. Each layer has a different architectural function, and you create architectural risks if you blur them together. If you need to see how something changes across layers – going from row-3 abstract to row-4 implementation-design, for example, or from an actual row-5 implementation back up to row-3 to rethink alternate options – then use a separate Canvas for each layer, and link them together via GV/GD/GK oversights [see the labels in the diagram above] and/or via XV/XD/XK or XM service-interfaces and flows. *Don't mix them up*!)

Next, we use the ***tetradian*** dimensions – physical, conceptual, relational, aspirational – to summarise the nature of the service (or whatever-it-is):

- What does it **do**? ('*physical*') What services or products or other 'deliverables' does this create through its work? (The answers define the role and emphasis of the Canvas' Value-Creation cell.)
- For whom or for what does it do this? With whom or with what does it **relate**? ('*relational*') Via what transactions and flows does it deliver? (The answers define the role and content of each of the flows, with an emphasis on each of the 'supplier-side' and 'customer-side' cells and the XC interfaces, and the direct transactional stakeholders.)

[76] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

- What **information** does it need, to guide and plan and schedule and improve what it does? ('*conceptual*') With whom or what does it need to share this information? (The answers define the role of the Value-Management cell, with some emphasis on the Value-Outlay and Value-Return cells and the XK, XD, XM and other interfaces.)
- **Why** does it do what it does? ('*aspirational*') What value does this add for each of its stakeholders, and for the overall enterprise? (The answers define and emphasise the role of the Value-Proposition cell, and key aspects of the Customer-Relations, Supplier-Relations, Value-Outlay and Value-Return cells, the XV, XT, XB, XO and XR interfaces, and the non-transactional stakeholders.)

Repeat the process for each of the cells (Vx, Rx, Cx), each of the flows (Xx), and each of the stakeholder-relationships (Px): What does it do? For and/or with whom or what does it do this? What information does it need? Why does it do this?

For example, often the service will need some form of investment to get it started, or keep it going (XO interface, linking between VO cell and PI or other stakeholder). What *is* this investment? What value and/or values does it represent? (Remember to think of 'value' in more than solely monetary terms.) What does this investment do? From whom does this investment come? What information is required to determine what investment will be needed?

Why is this investment needed? and so on.

Explore the service's needs for the *guidance* services (GV, GD, GK) that connect this to other parts of the enterprise and its values:

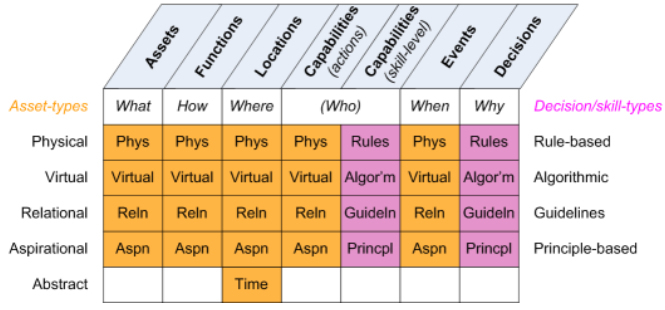
- Who or what provides run-time *coordination* (GK) with other services? Identifies and coordinates medium-term change of this service (such as via change-projects, -programmes or -portfolios), in parallel with other services? Develops overall potential for value-creation, in conjunction with other services?
- Who or what provides operational or run-time *direction* (GD) for this service and its sibling-services? (What *are* this service's sibling-services? What do *they* do? Why?) Who or what provides business-intelligence and other guidance for medium-term transition from strategy to tactics for this service? Who or what links this service to the overall enterprise to identify appropriate longer-term strategy?
- Who or what ensures *validation* (GV) and alignment to enterprise values, principles, rules and regulations? (What values and and the like will apply to this service? Why? What are their relative priorities, and why? What values and principles do *not* apply? Why not?) Who or what develops awareness within this service of the need, importance and application of each value, principle and so on? Who or what assists this service in developing its capability to enact and align with that value? Who or what will monitor,

audit and assure compliance with each value within this service?
- What interchanges, flows and interfaces are needed to support, guide, coordinate and validate all of this?
- What value does all of this provide to the service, in delivering its value-proposition to the enterprise? What impacts and constraints would apply to this service in the short-, medium- and longer-term if this coordination, direction and/or validation does *not* occur?

With all of that in place, we now have some solid understanding of what this service does, and why, and for and with whom. We now need to look more closely at the 'how' and 'with what' of the service – for example, that which the Business Model Canvas describes as Key Activities and Key Resources. As described in various of the earlier articles, one of the best frames with which to assess the interfaces and flows is **VPEC-T**[77] – Values, Policies, Events, Content, Trust – though we do need to remember to pay especial attention to the 'front end' of the market-sequence (reputation > trust > respect > attention > conversation), all of which will precede transaction, exchange and any potential and/or future profit. For the service itself, and for each of its cells – especially Value-Creation – another valuable frame is the **extended-Zachman**[78], adapting the standard IT-oriented Zachman frame for use in the much

[77] http://en.wikipedia.org/wiki/VPEC-T
[78] http://tetradianbooks.com/2008/12/silos-frame-ref/

broader scope of whole-of-enterprise architecture. The full version of the extended-Zachman frame requires three dimensions – rows, columns and asset/decision-category segments – but in this case we already know the applicable row, so we can use a simpler grid-type layout:



Enterprise Canvas: service-content

Note the somewhat different meanings of the columns here compared to standard-Zachman[79] – particularly the 'Capabilities' or 'Who' columns. These differences are extremely important, because without them the standard-Zachman frame is too IT-oriented for use in a whole-of-enterprise scope. One side-effect is that, by intention, there is no actual person (no 'who') in this version of the frame, because individual *people are not assets*: the capabilities that we need from people, for example, can only be accessed via something that actually *is* an asset of the enterprise, namely the *relationship* that it has with that

[79] http://www.zifa.com/framework.html

person.

Functions act on Assets to change them (i.e. create value); a Function needs to be linked to appropriate Capabilities in order to do so (which creates a *service*). Overall, we can summarise as follows:

with *Asset* do *Function* using *Capability* at *Location* on *Event* because *Decision*

Assets, functions, locations, capabilities and events all are, act on or are triggered by the various asset-types: physical 'things', virtual information, relations between people, or 'aspirational' links between people and abstract ideas or beliefs or values – see the explanation of asset-types in the 'Market and Supply-Chain[80]' article for more detail on this. For example, an event may be triggered by a actual incident (*physical* event), a specific value in a parameter (*virtual* event), the arrival of a person in a shop (a *relational* event) or a merger between two companies (a change of 'belief and belonging', hence an *aspirational* event). In this context, many if not most real-world items are actually composites or aggregations of different asset-types: for example, a book is a 'thing' (*physical*) that contains and carries information (*virtual*).

(Two other quick notes. Time is best understood as a *location* – events occur *in* time, but time itself is not an event. And whilst some people might class money as a special case of abstract, it's actually better understood as a simple

[80] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/

composite of *virtual* and *aspirational*, as information about a belief in future 'rights' of access to assets. All a bit technical, perhaps, but actually quite important once we start to explore in depth how value actually works within enterprise-architectures.)

Decision-levels and skill-levels are somewhat different but are closely-related. The **Cynefin**[81] categorisation can also be useful here: the skill-levels of simple rule-based, algorithmic, guidelines/heuristics and principle-based are likely to be needed in contexts that respectively match the Cynefin categories of Simple, Complicated, Complex and Chaotic. One important point is that there's a cross-alignment with the asset-types here: physical machines in general can only follow simple rules; IT-systems can also manage complicated algorithms for decision-making; real people with real skills are needed to handle true complexity; and very high skill-levels are needed to work within 'chaotic' contexts where nothing ever actually repeats. One of the most serious problems to affect organisations in recent times has been the repeated attempts to use IT for complex or 'chaotic' contexts where, by definition, it is simply not suitable: for example, this was a key cause of failure in many attempts at IT-driven 'business process re-engineering'.

So: apply this extended-Zachman frame to the service in focus on the Enterprise Canvas, and to each of its cells:

- What assets, functions and capabilities are required

[81] http://en.wikipedia.org/wiki/Cynefin

here? What asset-type categories are involved in each case?

- In which locations do they occur? Of what asset-type categories are these locations: a physical place, a virtual-address, a reporting-relationship or other social-network location, a point in time?
- What events and event-types trigger these activities? (Cross-reference to the previous VPEC-T assessments on external flows, and perhaps apply similar analysis to any internal flows between cells.)
- What skill-levels and decision-types apply in each case? – rule-based, algorithmic, pattern-based guideline or heuristic, and/or principle-based?

Next, apply a **RACI**[82] frame to identify responsibilities for each item, activity, event and decision within the service and its cells, within and for each flow or interchange, and the equivalent responsibilities in and of each stakeholder:

- Who is *responsible* for this, and any changes to this? – "those responsible for the performance of the task"
- Who *assists* in this, and any changes to this? – "those who assist in the completion of the task"
- Who should be *consulted* about this, and any changes to this? – "those whose opinions are sought, and with whom there is two-way communication"

[82] http://en.wikipedia.org/wiki/Responsibility_assignment_matrix

- Who should be *informed* about this, and any changes to this? – "those who are kept up-to-date on progress, and with whom there is (usually) one-way communication"
- Optionally, who does *not* need to be (or should not be) responsible, accountable, consulted and/or informed about this, or any changes to this? (e.g. an executive does not normally need detailed reports about every minor operational change)

There should always be *one* person who is uniquely responsible ('accountable') for any item; and, equally, every item needs one person who is uniquely responsible for it at all times. Look for overlaps in responsibilities (often occurring in transitions between layers – for example, one person has operational responsibility for an item, another has tactical responsibility, another the strategic responsibility). Perhaps even more important, look for gaps in responsibilities, where *no-one* has apparent responsibility or where the responsibility has been nominally assigned but not actually taken up or enacted. (Note that simply assigning someone responsibility does not necessarily mean that that responsibility is taken up! – functional responsibility is an active *choice*, not an arbitrary label.)

Also note that real responsibility can only be held by a real person: a machine or an IT-box is *not* capable of taking ultimate responsibility for anything – hence, especially wherever IT or machines are involved, it may be necessary to follow lengthy trails of non-responsibility (or evasions of

responsibility…) in order to identify the actual responsible person. And, finally, a person can *only* be responsible for something if they also have the *authority* and *competence* to make the required decisions: if they don't have that authority or competence, they *cannot* and must not be considered responsible or accountable for the respective item, and hence a further search for someone who *can* have both responsibility and authority will need to be made. (Note that such mismatches of responsibility, authority and competence are extremely common, especially in dysfunctional organisations.)

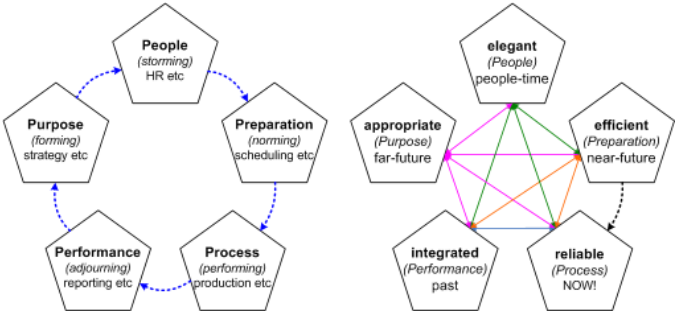
Next, do a **SWOT**[83] assessment on the overall service, and on each of its cells, its flows and its links with each of its stakeholders, and on the links up and down the rows, from abstraction (redesign) to implementation. (For this purpose I would actually recommend my own extended variant of SWOT, called SCORE[84] – Strengths, Challenges, Opportunities/risks, Rewards, Effectiveness – but I would have to admit that SWOT is better-known!)

- What are the *Strengths* in this context? Which of these strengths – if any – is underutilised, or could be applied in other, even more effective ways?
- What are the *Weaknesses* in this context? What impacts could or do these have on operations, on tactics, on strategy? In what ways could these challenges be redressed or mitigated?

[83] http://en.wikipedia.org/wiki/SWOT_analysis
[84] http://tetradianbooks.com/2008/07/semper/

- What are the external *Opportunities* in this context? In what ways could these opportunities be exploited? What concomitant risks do each of these opportunities imply?
- What are the external *Threats* in this context, from what and/or from whom? In what ways could these risks be redressed or mitigated? What concomitant opportunities do these risks imply?

Another useful frame here is one that I documented in my book *Real Enterprise Architecture*[85] – a 5×5 matrix that links **lifecycles** and **effectiveness**:



Five Elements: lifecycles and effectiveness

The *lifecycle* part of the matrix – shown on the left-hand side of the diagram above – is derived in part from Bruce Tuckman's well-known Group Dynamics[86] project-lifecycle: forming, storming, norming, performing, adjourn-

[85] http://tetradianbooks.com/2008/04/real-ea/

[86] http://en.wikipedia.org/wiki/Tuckman's_stages_of_group_development

ing. For the service overall, for each of its cells, and for each of its flows, we need to ask:

- Who or what is responsible for the *Purpose* of this item? (This should link strongly with the Value-Proposition cell, for example, and also the 'direction' guidance-services.)
- Who or what is responsible for the *People*-issues for this item? (This is where the 'validation' guidance-services are likely to play a key part in the service and, for external links, the Supplier/Customer-Relations cells.)
- Who or what is responsible for the *Preparation* and scheduling for this item? (This should link strongly with the Value-Management cell, and also the 'coordination' guidance-services.)
- Who or what is responsible for the *Process* and action of this item? (This should link strongly with the Value-Creation cell, probably the Supplier/Customer-Channel cells, and probably also the 'coordination' guidance-services.)
- Who or what is responsible for the overall *Performance* of this item, including completions and lessons-learned? (This should again link strongly with the Value-Management cell, the Value-Outlay/Return cells, and also the 'validation' guidance-services.)

Serious architectural problems will occur if these elements are out of balance – which they often are, in real-world

practice. In commercial organisations, for example, there's often a strong tendency to attempt to short-cut the cycle, jumping from the early stages of the *Performance* phase – the point at which monetisation occurs – back to the start of the *Preparation* phase – so as to return to *Process* and purported profit as quickly as possible; lessons-learned, long-strategy and people-issues are all minimised and glossed-over as 'unnecessary overhead'. The result does deliver higher gains in the short-term – because the costs of the skipped-over phases *are* avoided – but will *inevitably* lead to longer-term failure. Identifying and resolving such imbalances is a key task for whole-of-enterprise architecture.

The *effectiveness* part of the matrix – shown on the right-hand side of the diagram – also crosslinks to the lifecycle part, as can be seen in the secondary captions for each of the five effectiveness-domains. For the service overall, for each of its cells, and for each of its flows, we need to ask, everywhere, and in every way:

- Is it *Efficient?* – maximises use of resources, minimises wastage of resources
- Is it *Reliable?* – predictable, consistent, self-correcting, supports 'single source of truth'
- Is it *Elegant?* – clarity, simplicity, consistency, self-adjusting for human factors
- Is it *Appropriate?* – supports and maximises support for business purpose
- Is it *Integrated?* – creates, supports and maximises synergy across all systems

The aim here is that everything should fit in with and support everything else: these questions help to identify where they do, where they don't and, if they don't, what to do about it!

Before we move on to real examples, I'll recommend one final frame for additional crosschecks: **SEMPER**[87]. I'll admit this is another of my own tools, but one I've found immensely value in assessing *ability to do work* within a context. The point here is whilst the physics definition of 'power' approximates to 'the ability to do work', many social definitions of 'power' are more like 'the ability to *avoid* work' – and therein lie some *huge* problems for organisations and for enterprise-architectures. It's actually based on the *tetradian* model, as described earlier above, but the SEMPER-5 variant crosslinks with the lifecycle/-effectiveness matrix and uses a simple 1-5 scale to identify potential problems across a context, and suggest options for action to resolve them. All I can say is try it, see what you think – but my experience is that it *does* work well for the kind of contexts we cover with the Enterprise Canvas.

So, that's one suggested set of models that you could use with the Enterprise Canvas – where the Canvas becomes a base-map for deep exploration, a "one map to rule them all" and bind them together to create new insights about the overall architecture. Yes, it's a lot of work: but then anyone who thought that it *wouldn't* be a lot of work probably hasn't done much real-world enterprise-architecture…

[87] http://sempermetrics.com

Enough for now, anyway: in the next article we'll look at some examples of what this all looks like in real-world practice.

Source (Tetradian weblog)

- *Date*: 2010/07/07
- *URL*: enterprise-canvas-pt6[88]
- *Comments*: (none)
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

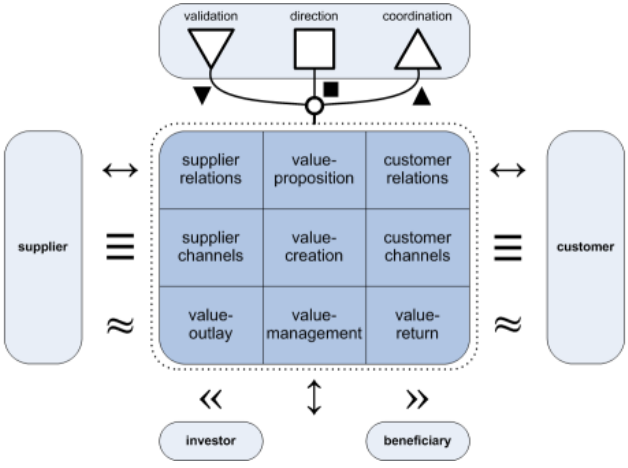
[88] http://weblog.tetradian.com/enterprise-canvas-pt6

The Enterprise Canvas, Part 7: Patterns

We wandered through the background and theory of the Enterprise Canvas with context and value[89], market and supply-chain[90], owners and managers[91], layers[92] and recursion[93]; we've made a start on putting it to use, by showing how different models[94] can be layered on top of each other to provide a richer view of a context. Time now to add to that by exploring some example patterns.

Start with a recap on the Enterprise Canvas:

[89] http://weblog.tetradian.com/index.php/2010/07/02/enterprise-canvas-pt1/
[90] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[91] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/
[92] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/
[93] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/
[94] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/



Enterprise Canvas: complete

The Canvas itself is the centre section, nine cells that describe a single service at the intersection of value and supply-chain.

The labels 'supplier' and 'customer' are arbitrary labels for other players in the shared extended-enterprise. The only difference between them (and their relations with this service, too) is the respective emphasis on the direction of flow in specific interfaces; the interfaces themselves (horizontal double-head arrow for 'relations', triple-line for 'main channel', double-squiggle for 'backchannel') are always much the same in each case. Likewise the only real difference between 'supplier-side' and 'customer-side' on the Canvas itself is the direction that the respective cells and sub-services will face: the respective interfaces are

actually the same (or 'symmetrically equivalent') on either side.

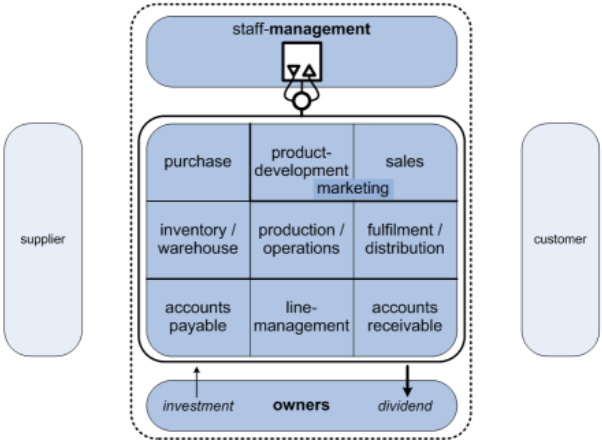
The 'investor' and 'beneficiary' could just as well be considered specific types of 'customer' and 'supplier' respectively: the main difference is that (through the interfaces shown as left or right double-guillemets) they provide or receive forms of value that would more typically go through the backchannel (e.g. money) rather than through the main channel (e.g. goods and services).

The vertical double-arrow represents a sort-of interface that connects services at different levels of abstraction. In real-world practice, this is reflected in the information-flows that pass between different levels of management (particularly middle-management): going 'down', generalised strategy is expanded out to more explicit tactics, and thence to detailed operational instructions; and the resultant activity-reports and performance-reports are aggregated and summarised 'upward'.

The 'guidance' interfaces and functions – validation, direction, coordination – link this specific service with other services within the organisation *and beyond* to the broader shared-enterprise.

Example: for-profit business

I'll admit this is very much a stereotype, but is actually a typical outcome of Taylorist 'scientific management' and current commercial law, especially in the US.



Enterprise Canvas: stereotype Taylorist business

In this structure, 'the enterprise' is deemed to be synonymous with 'the organisation as a whole' (the boundary shown by the dotted line). By custom and even by law, it's purported that the only reason that the organisation exists is to 'make money' for the stockholders ('owners'); monetary return is defined as the only meaningful form of value. There is no real link – or even awareness – of a shared-enterprise beyond the organisation; relations with suppliers and customers are characterised by a combative 'us against them'. All of the guidance-functions are subsumed into 'direction', possessed exclusively by the executive management on behalf of the stockholders; 'validation' and 'coordination' are replaced by 'command' and 'control' respectively, and often explicitly disconnected from anything beyond the organisation (e.g. insistence on proprietary

standards and the 'not invented here' syndrome).

Within the service-functions represented in the main Canvas, the natural boundaries between the cells tend to be exaggerated and reinforced as distinct silos. There is often a rigid separation between development and production, and even more so between line-management (which claims the entire backchannel as its own domain, asserting rigid control over budget and all monetary matters) against all other parts of the service: the classic Taylorist separation between 'management' and 'workers'.

The anomalous position of 'marketing' – stuck halfway between the Value-Proposition ('product-development') and Customer-Relations ('sales') cells – is a direct outcome of the disconnect from any shared extended-enterprise. Since no connection exists to link the organisation to a broader enterprise-vision, there is no possibility for a 'pull'-based marketing model[95] via a shared Value-Proposition; the only available alternative is product-development, in essence derived from internal assumptions about 'something we can sell'. Without any direct engagement in the value-proposition, the organisation's potential customers are unlikely to feel much inherent engagement in the product; hence it becomes the unhappy role of marketing to attempt, via 'push', to manufacture a 'relationship' with customers where no real reason for any such relationship actually exists.

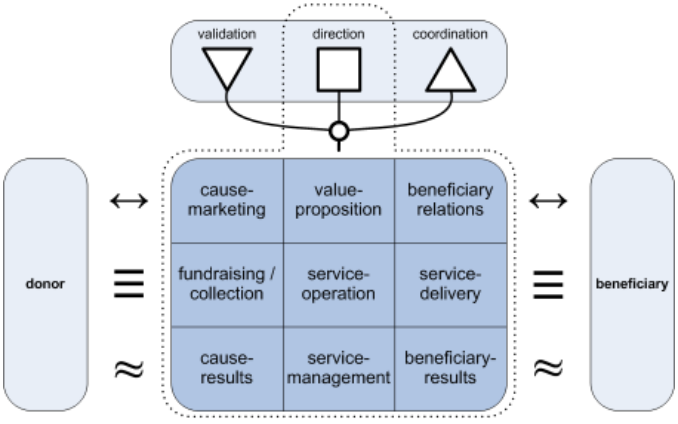
Overall, not a happy picture; certainly not a very effective

[95] http://blogs.hbr.org/bigshift/2010/04/a-brief-history-of-the-power-o.html

one. It is also, unfortunately, very common. Oh well… Yet a comparison of this dysfunctional structure with the generic Enterprise Canvas further above indicates what *can* be done to improve matters – of which the most urgent and important is to break the delusion that the organisation 'is' the enterprise, and instead connect the organisation with the shared-vision and shared values of its broader market and extended-enterprise.

Example: not-for-profit charity

This is somewhat less of a stereotype than the previous example, but still very much generic. (Much the same principles as here *should* apply to the government context, since that too is nominally 'not-for-profit'; but in practice the 'guidance' functions there often tend to be subsumed into the executive, much as in the dysfunctional for-profit model, and with much the same dysfunctional results.)



Enterprise Canvas: stereotype charity

In this example, the charity aims to deliver a service to its beneficiaries, yet to do so must engage donors to provide the resources needed in order to deliver those services, and hence must also act as a direct and explicit intermediary between donors and beneficiaries. (The same principles apply to almost any charity, but charities in practice often have very complex relations between many different types and roles of providers; for this example we'll keep it simple, and imagine a charity that provides clothing to victims of natural disasters.)

Here the boundaries of the organisation (shown by the dotted-line) may include the 'direction' guidance-functions – as in a standard for-profit business – but the 'validation' and 'coordination' functions *must* extend beyond the organisation itself. On the 'coordination' side, this

charity *must* coordinate its service-delivery with other charities and services that operate the same disaster-zone – medical, shelter, food, rescue, security and suchlike; and on the 'validation' side, the charity's services and activities will only make sense to donors and beneficiaries alike if they connect to a shared-vision such as "keeping people safe, warm and comforted after an emergency". (Many charities start off with something like a 'product-development' model, designing support-services of some kind as a metaphoric equivalent of 'something we can sell', and then delivering those services: but as a 'push'-type activity it may be difficult for it to make sense even to the nominal beneficiaries, and unless the connection is made to a 'higher cause' that will engage others, the charity will soon run out of resources, and fail. All the problems and potential mistakes faced by commercial 'startups' apply just as much to charities and other not-for-profits: the only difference is that resources and 'profit' may be measured in a different way.)

The value-proposition here is straightforward: given the shared-vision, there is a clear need for clothing to replace that lost in natural disasters – a need which will be all too evident to the intended beneficiaries, and will also make immediate sense to potential donors. This in turn implies the need for a service to deliver such clothing in a disaster-recovery context (the Value-Creation cell, here described as 'service-operation') and for some form of management, in donor locations, on the spot in the disaster-affected region, and in logistics in between (the Value-Management cell,

here labelled 'service-management'). The linkage to value and enterprise-values is therefore reasonably clear, though we will have to drill down from this very simple row-2 model all the way to a row-5 detailed-implementation model to get a proper picture of what all of this would entail in practice.

On the donor side, we need to explain what the need is, and why it's important (Supplier-Relations, here labelled 'cause-marketing') and encourage *two-way* conversations that will engage people directly in the enterprise-values and thence in the charity's value-proposition. Assuming this engagement does happen, we'll also have created the space for transactions to occur, much like a supplier-relationship in a commercial context – in this case supplying either clothing, or money to assist in delivery of that clothing, or both (Supplier-Channels, here labelled 'fundraising/collection').

On the beneficiary side, we need to identify appropriate 'customers' for our clothing service; and given that this is a disaster-recovery context, gaining their *trust* will also be crucial here – hence, again, the importance of *two-way* communication to create engagement in the shared-vision (Customer-Relations, here shown as 'beneficiary-relations'). We then need to match their needs with the available resources, both on-the-spot and via requests further back along the charity's logistics chain (Customer-Channels, here labelled 'service-delivery').

In both cases the backchannel (Value-Return and Value-

Outlay, here labelled 'beneficiary-results' and 'cause-results' respectively) is *significantly* different from that in a commercial context. For a business, 'success' is a profitable sale; income and costs alike can, it seems, all be reduced to monetary metrics, which makes the measurement of 'success' a mere matter of quick calculation – all very simple and straightforward. But as soon as we realise that there are other forms of value in play than money alone – as is immediately evident here, if perhaps less so in a commercial context – then we need to look for a much broader definition of 'success', in fact drawn directly from the shared-values of the extended-enterprise. In this case the charity would probably not expect any payment from the beneficiaries for the clothing: what will matter much more instead will be some form of proof that the service-delivery was in accordance with the shared-values of the enterprise – hence, for example, all those photos of happy, smiling, *well-clothed* children that we would expect to see passing through the backchannel, to appear as 'cause-results' on the charity's website and in reports to donors. The backchannel closes the loop of value; and its balance overall, across all relevant forms of value, provides the proof that the organisation has indeed delivered *added value* in terms of the extended-enterprise's vision and values.

Note that it's not true that money doesn't matter here: it does. In fact it matters a *lot*, not least because potential donors are very quick to withdraw support from any charity that is perceived to carry too much overhead in any

form, monetary or otherwise. The key point here is that money is not the *only* form of value in play in the overall transactions of the charity – a point which, architecturally, also applies in every 'for-profit' organisation as well.

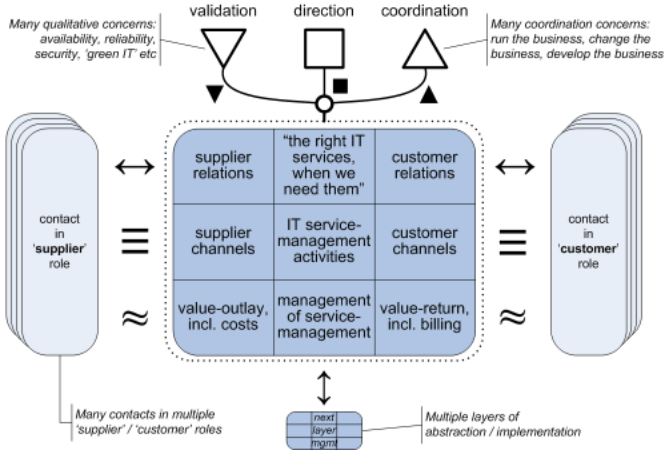
Example: ITIL IT service-management

For this next example, let's pick on something more in the mid-range: IT service-management within a large organisation. The typical reference for this is ITIL Version 3[96] [PDF, 58pp], the IT Infrastructure Library[97]:

[96] http://www.best-management-practice.com/gempdf/itSMF_An_Introductory_Overview_of_ITIL_V3.pdf

[97] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

This describes service-management overall in terms of five distinct strands: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Process Improvement. So: map those core ideas onto the Enterprise Canvas, and see what we get in terms of an overall pattern for service-management. The result shows that it does all fit quite well:



Enterprise Canvas: stereotype IT service-management

One immediate point is that the definitions of 'customer' and 'supplier' are fairly fluid here: many actual stakeholders will take both types of roles at various times, so it definitely does help if – as the Canvas allows us to do – we take a symmetric view of the respective relations, main-channel and backchannel interfaces.

Another key point is that it does depend strongly on a clear value-proposition, here summarised as "the right IT services, when we need them". All of the ITIL processes revolve around that core idea, and the values that lie behind it. The ITIL specification helps in this, too: for example, in describing Service Strategy, the Overview document states:

> The service strategy of any service provider must be grounded upon a fundamental ac-

knowledgement that its customers do not buy products, they buy the satisfaction of particular needs. Therefore, to be successful, the services provided must be perceived by the customer to deliver sufficient value in the form of outcomes that the customer wants to achieve.

Achieving a deep understanding of customer needs, in terms of what these needs are, and when and why they occur, also requires a clear understanding of exactly who is an existing or potential customer of that service provider. This, in turn, requires the service provider to understand the wider context of the current and potential market places that the service provider operates in, or may wish to operate in.

A service strategy can not be created or exist in isolation of the over-arching strategy and culture of the organization that the service provider belongs to. The service provider may exist within an organization solely to deliver service to one specific business unit, to service multiple business units, or may operate as an external service provider serving multiple external businesses. The strategy adopted must provide sufficient value to the customers and all of the service provider's stakeholders – it must fulfill the service provider's strategic

purpose.

All of this requires very strong linkages beyond the service itself – almost the exact antithesis of the Taylorist-style model for the 'for-profit business' example above. In the Enterprise Canvas, these linkages are provided by (or rather, modelled as) the guidance-services: the structured layers of 'validation', 'direction' and 'coordination', as described in the 'Systems' section of Part 5[98] of this series of articles.

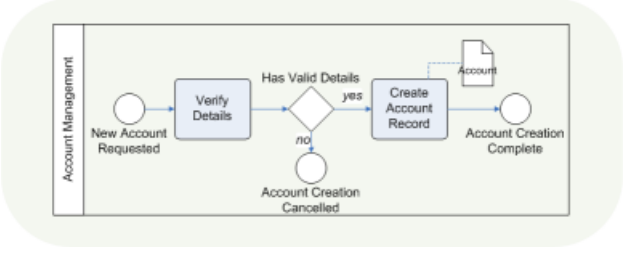
To model ITIL in its entirety, we would probably create a separate Enterprise Canvas at row-2 for each of the five major service-streams, shown in parallel on a single diagram, and linked upward to the organisation (row-1) and the extended-enterprise (row-0). We would then expand downward, into a more detailed set of row-3 models (some of which would be shared across two or more of the ITIL service-groups), and then downward again to row-4 and beyond. But it's clear that it does all fit well: a very useful exercise for any group of enterprise-architects with suitable time to spare, perhaps?

Example: BPMN process-model

This one is perhaps a bit unfair, but it *is* a pattern, and one that will help to demonstrate why there should be serious concerns about the limitations and incompleteness

[98] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/

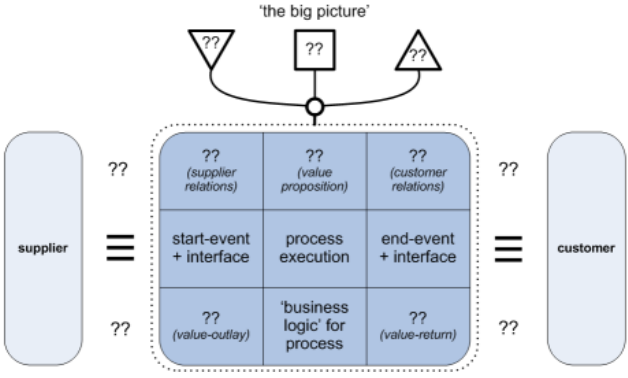
of so much current so-called 'enterprise-architecture'. To illustrate this, let's return to the very(!) simple BPMN diagram from back in Part 5[99]:



Account-service: BPMN process

The process-model purports to describe the *service* of creating an account – in other words, something that we would expect to model on the Enterprise Canvas. So let's map this onto the Canvas, and see what we get:

[99] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/



Enterprise Canvas: stereotype BPMN process

For which the short answer is "Not much"… only a tiny subset of what we actually need in order to make sense of it *as* a service:

- There's no concept of any precursors to transactions – the content and interfaces for the usual Supplier/Customer-Relations cells.
- There's no concept of any backchannel – the content and interfaces for the Value-Outlay/Return cells.
- There's only the most minimal of content for Value-Management, namely the small part that deals with the decision-making and business-logic with the execution of the process.
- There's no 'why' *anywhere* in the process-model – no Value-Proposition, no known reason why this particular process and service should even exist.

- There's no linkage anywhere to 'the big picture', the guidance-services that link this process to the various qualitative concerns, or even to the inter-service choreography need to coordinate the execution of the complete end-to-end business-process.

(To be fair, the equivalents of 'Supplier' and 'Customer' and their interfaces would be represented by headers in parallel swimlanes – but that still doesn't tell us anything about the coordination of the full *end-to-end* business-process.)

The Canvas itself doesn't show the assets (the account-record, for example) or the triggering events, or the location either (something else that's absent from the BPMN model); but all of those are items that we would expect to pick up straight away by applying an extended-Zachman view[100] to the respective cells (especially the Value-Creation cell, here shown as 'process execution'). Likewise a VPEC-T[101] assessment of the interfaces would tell us much more about the respective flows than we have here in the BPMN model.

Again, I know this example has been a bit unfair, somewhat of a straw-man (rather than a robbie-the-robot? :grin:). But what worries me is that I've seen all too many examples where BPMN diagrams and the like have been presented *as* 'architecture' – even as 'enterprise-architecture'… which it isn't. At all.

What this pattern *does* show us, though, is how use a BPMN

[100] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/
[101] http://en.wikipedia.org/wiki/VPEC-T

diagram to populate this part of the base-content for an Enterprise Canvas – and then go on from there to fill in all of the blanks, to build the complete of the service that we need for an architecture.

So: that's it – a few Enterprise Canvas patterns to play with. But how does all of this fit in with all the other models that abound in enterprise-architectures? Something to explore in the next and final article in this series!

Source (Tetradian weblog)

- *Date*: 2010/07/08
- *URL*: enterprise-canvas-pt7[102]
- *Comments*: 2
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[102] http://weblog.tetradian.com/enterprise-canvas-pt7

The Enterprise Canvas, Part 8: Integration

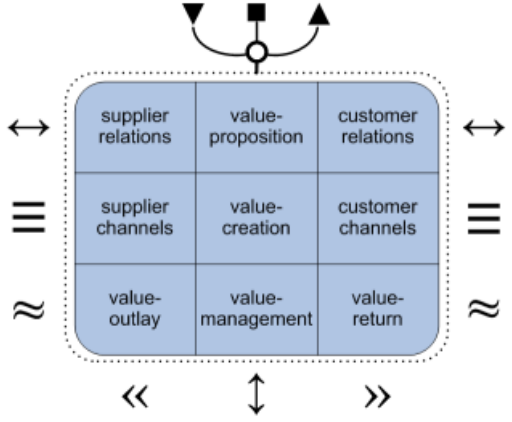
Over previous articles in this series, we established some of the background and theory of the Enterprise Canvas with context and value[103], market and supply-chain[104], owners and managers[105], layers[106], and recursion[107]. We then started to explore how to put this to practical use, with a detailed process that links to a variety of other architecture-models[108], and some example-patterns[109] for specific architectural contexts.

The other promise, way back in the introduction[110], was that the Canvas can be used as "one model to rule them all" within enterprise-architectures – not by 'ruling from above', or replacing anything else at all, but by providing a unifying frame through which all the different model-types at all the different levels can link together and support each other. So in the this final section I'll skim through a wider

[103] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/
[104] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[105] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/
[106] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/
[107] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/
[108] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/
[109] http://weblog.tetradian.com/2010/07/08/enterprise-canvas-pt7/
[110] http://weblog.tetradian.com/2010/07/02/quest-for-enterprise-canvas/

range of model-types that are in common use in or around enterprise-architectures, to show how the Canvas could be used with each of them to support an overall integration across the architecture.

As in the previous article, let's start with a recap on the Enterprise Canvas, using the 'beetle' version of the model:



Enterprise Canvas: complete, with flows

Each Canvas represents and describes a single unit or entity that provides a service of some kind. It sits at an intersection of value and action: the 'vertical' axis defines what 'value' *is* for the respective enterprise, the 'horizontal' axis describes how the unit links in with a supply-chain or 'value-web' of of services that each add value to the overall shared-enterprise.

Three types of horizontal flows (the horizontal double-

arrow, triple-line and double-squiggle) represent interconnections with other parties that take place before ('relations'), during ('main-channel') and after ('backchannel') the main or more-tangible transactions with those parties.

Three types of vertical flows 'below' the Canvas link to investors (left double-guillemet), more-detailed implementations (vertical double-arrow) and beneficiaries (right double-guillemet).

Three further types of flows – shown above the Canvas, but in essence each orthogonal to the others – link the overall service to a more enterprise-wide support, with information and emphasis on values and value-implementation ('validation', downward-pointing triangle), on purpose, strategy and tactics ('direction', small square), and on cross-functional coordination and change-management ('coordination', upward-pointing triangle).

These flows indicate the items that need to be exchanged across them; with whom are what these should be exchanged – 'suppliers', 'customers' and so on – is another question entirely. Sometimes it's useful to start from a list of stakeholders and identify the respective flows from there; sometimes it's better to start with the flows, and identify the respective stakeholders. It's important to do both, though, because that's the best way to identify gaps that could otherwise render the service non-viable.

Architectural frameworks and model-types

VPEC-T[111] – Values, Policies, Content, Events, Trust – is, as we've seen in other[112] articles[113] in this series, probably *the* most useful frame with which to assess the various flows between this service and its stakeholders. You'll find more details about the VPEC-T book *Lost In Translation* on Amazon[114].

Archimate[115] is an upcoming standard notation for IT-oriented components of enterprise-architectures, originally developed in the Netherlands and now adopted as an international standard[116] by The Open Group[117]. (There is an intent that in the medium-term it should extend to cover a true whole-of-enterprise scope, but at present it's still mostly constrained to IT-oriented architectures only.) It defines architectures in terms of the usual three IT-oriented 'layers' (Business, [IT] Applications and [computer-based] Technology) and three interlinked 'vertical' emphases: Static Structure (data etc), Behaviour (services and functions) and Active Structure (roles and interfaces). From a Canvas perspective:

[111] http://en.wikipedia.org/wiki/VPEC-T
[112] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[113] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/
[114] http://www.amazon.co.uk/Lost-In-Translation-Nigel-Green/dp/0978921844
[115] http://en.wikipedia.org/wiki/ArchiMate
[116] http://www.archimate.org/
[117] http://www.opengroup.org/archimate/

- Archimate 'layers': 'Business' primarily sits in the row-2 to row-3 Canvas layers[118], whereas 'Application' and 'Technology' sit *in parallel domains* in the Canvas row-3, row-4 and sometimes row-5 layers.
- 'Static Structure': data-items and other artefacts would translate to content that is held and/or manipulated within the Canvas cells or exchanged via the Canvas flows (it would be considered primarily as 'Content' in VPEC-T terms, or as 'assets' in modified-Zachman[119] terms).
- 'Behaviour': each service in each Archimate layer would typically be modelled on its own Canvas, whilst processes and functions would typically sit in the Value-Creation cell of the respective Canvas.
- 'Active Structure': roles would typically translate as external parties, interfaces as Canvas flows, and components ('device', 'application component', 'network' etc) as part of the respective Supplier/Customer-Channel cell.

TOGAF[120] (The Open Group Architecture Framework) is another very well known framework in IT-oriented 'enterprise'-architectures, in essence consisting of a well-defined process (Architecture Development Method [ADM]) and an underlying metamodel. The metamodel is less well-structured than Archimate's, but the same general princi-

[118] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/
[119] http://tetradianbooks.com/2008/12/silos-frame-ref/
[120] http://www.opengroup.org/togaf/

ples apply as for Archimate: in fact there is an ongoing effort within the Open Group to align the two models, so any differences should fade away in the relatively near future. The ADM works well with the Enterprise Canvas, but in its standard form is still *very* IT-oriented and almost unusable beyond IT; over the past few years, though, I've done a lot of work to adapt it for use at a whole-of-enterprise scope, as documented in some of my books such as *Bridging the Silos*[121] and *Doing Enterprise-Architecture*[122] and in TOGAF-conference presentations available on Slideshare[123], and also summarised in a brief two-page reference-sheet on the Tetradian Books website[124].

The **Zachman framework**[125] is probably the 'granddaddy' of all enterprise-architecture frameworks, and an essential counterpart to the Enterprise Canvas. (The standard version[126] is unfortunately still somewhat IT-centric – for example, the only asset-type ('What') listed is Data, the only decision-type ('Why') is [IT] Business-Rule – and needs some modification[127] to be usable for whole-of-enterprise architectures.) As described in Part 4[128] of this series, the Enterprise Canvas layers are an extension of the Zachman layering, and directly compatible with it; and as

[121] http://tetradianbooks.com/2008/04/silos/
[122] http://tetradianbooks.com/2009/03/doing-ea/
[123] http://slideshare.net/tetradian
[124] http://tetradianbooks.com/2008/10/silos-method-ref/
[125] http://en.wikipedia.org/wiki/Zachman_Framework
[126] http://www.zifa.com/framework.html
[127] http://tetradianbooks.com/2008/12/silos-frame-ref/
[128] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

described in Part 6[129], a single-row version of the modified-Zachman should be used to assess the content and structure of a Canvas and each of its cells.

VSM[130], Stafford Beer's Viable System Model, was one of the core inspirations for the Enterprise Canvas. Although the original model primarily focussed on information-flows, the same concepts can be applied to flows of any asset-types, both simple and composite. On the Canvas:

- VSM system-1 (operations or service-delivery): represented by the nine-cell core of the Canvas
- VSM system-2 (coordination): represented by the 'coordination' guidance-service (upward-pointing triangle)
- VSM system-3* (audit): represented by the 'validation' guidance-service (downward-pointing triangle)
- VSM system-3 (planning and control), -4 (near-future) and -5 (far-future): represented by the 'direction' guidance-service (square)

Beer's concept of an algedonic high-priority feedback-path that can bypass the normal channels is extremely important, and should be considered when assessing any or all of the Canvas flows. For more information on the standard

[129] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/
[130] http://en.wikipedia.org/wiki/Viable_System_Model

VSM and its application in business, see Patrick Hoverstadt's excellent *The Fractal Organization*[131]; for more on its adaptation for service-oriented whole-enterprise architectures, see my book *The Service-Oriented Enterprise*[132].

Context-space mapping[133] is the actual framework and process on which the Enterprise Canvas is based, and is described in more detail in my book *Everyday Enterprise-Architecture*[134]. The underlying process draws strongly on John Boyd's OODA[135] (observe, orient, decide, act) model for agile problem-solving, crosslinked to a concept of sensemaking-domains[136] derived from my book *Inventing Reality*[137] (originally published in 1986). In context-space mapping, we choose a base-frame as a core-model, and then cross-map to other compatible model-types whilst we 'go for a walk' – metaphorically speaking – through all of the different perspectives offered by the base-model and the cross-maps. The Enterprise Canvas was designed for use as a base-map in context-space mapping; as described in Part 6[138] of this series, other useful base-maps or cross-maps include the modified-Zachman framework, the Cynefin-categorisation, RACI and SWOT, and the Five-Element

[131] http://www.amazon.co.uk/Fractal-Organization-Creating-Sustainable-Organizations/dp/0470060565

[132] http://tetradianbooks.com/2008/12/services/

[133] http://weblog.tetradian.com/2010/03/04/context-space-mapping/

[134] http://tetradianbooks.com/2010/05/everydayea/

[135] http://en.wikipedia.org/wiki/OODA_loop

[136] http://www.tomgraves.org/3science

[137] http://www.tomgraves.org/inventin

[138] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/

lifecycle and effectiveness frameworks.

Strategic frameworks and model-types

The **Business Model Canvas**[139] was in some ways the starting-point for this work, and the Enterprise Canvas was intentionally designed to be compatible with it. As shown in Part 2[140] of this series, business-models developed with the Business Model Canvas can be mapped directly onto an Enterprise Canvas for architectural expansion 'downward' towards implementation, and 'upward' for verification of alignment with the vision and values of the shared-enterprise. The book *Business Model Generation*[141] by Alex Osterwalder et al. should be regarded as an essential companion for any work with the Enterprise Canvas, especially at a row-2 or row-3 level for business-architectures.

The OMG/BRG **Business Motivation Model**[142] [PDF] provides a standard framework to assess business drivers and business goals, and would particularly apply to the XD flow between the core Canvas and the 'direction' guidance-services. Be warned, though, that its method for definition of 'vision' – which should be the key to the entire enterprise – conforms almost exactly to how *not* to frame an

[139] http://en.wikipedia.org/wiki/Business_Model_Canvas
[140] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[141] http://www.businessmodelgeneration.com/
[142] http://www.businessrulesgroup.org/second_paper/BRG-BMM.pdf

enterprise-vision, and in practice is all but guaranteed to cause failure, especially over the longer-term. To avoid major architectural problems, use the more functional framing for shared-enterprise vision, as described in *Everyday Enterprise Architecture*, or in my presentation 'Vision, Role, Mission, Goal[143]' on Slideshare[144].

Other common strategy-frameworks that would work well with the Enterprise Canvas include Porter's **Five Forces**[145], **Blue Ocean Strategy**[146], **Porter Value-Chain**[147] and the ubiquitous **SWOT**[148]. In each case we would typically use the framework to assess the service or organisational-unit in the Canvas, comparing it against others (such as for competitiveness analysis) or assessing the flows between the service and other players in the shared-enterprise (supply-chain partners, competitors etc). SWOT would be used to assess strengths, challenges, opportunities and the like in the usual way, on the service as a whole, on individual cells, on flows, and on stakeholder-relationships, as summarised in Part 6[149] of this series.

[143] http://www.slideshare.net/tetradian/vision-role-mission-goal-a-framework-for-business-motivation

[144] http://www.slideshare.net/tetradian

[145] http://en.wikipedia.org/wiki/Porter_five_forces_analysis

[146] http://en.wikipedia.org/wiki/Blue_Ocean_Strategy

[147] http://en.wikipedia.org/wiki/Value_chain

[148] http://en.wikipedia.org/wiki/SWOT_analysis

[149] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/

Structural and operational frameworks and model-types

Industry-specific structural frameworks such as **SCOR**[150] (supply-chain), **eTOM**[151] (telecoms) and **ITIL**[152] (IT service-management) fit well with the Enterprise Canvas: each model describes its context in terms of services, each of which – with its relationships to other services – would be a candidate for modelling on a Canvas. The main value of modelling with Enterprise Canvas is that it adds distinctions between types of flows that occur before, during and after the main service-transactions, and also documents the necessary 'guidance-service' linkages to overall direction, cross-functional coordination and whole-of-enterprise values.

SCOR is a layered, hierarchical model whose structures map to the Enterprise Canvas as follows:

- SCOR's levels 1, 2 and 3 correspond approximately to the Enterprise Canvas rows 2-4
- the SCOR level-1 structure of Source / Make / Deliver corresponds almost exactly Enterprise Canvas linkage to the supply-chain: SCOR 'Source' corresponds to EC 'Supplier-Channels' flow and cell; SCOR 'Make' corresponds to EC 'Value-Creation'

[150] http://en.wikipedia.org/wiki/Supply-chain_operations_reference

[151] http://en.wikipedia.org/wiki/Enhanced_Telecom_Operations_Map

[152] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

cell; SCOR 'Deliver' corresponds to EC 'Customer-Channels' cell and flow

eTOM is also a layered model: eTOM's levels 0 to 3 correspond approximately to the Enterprise Canvas rows 1-4. As a matrix structure, services are defined within intersecting cells of the matrix, as well as within the row/column overlays themselves, but in each case the respective service and its relationships can be mapped on its own Canvas. At the highest level, the eTOM 'Strategy Infrastructure and Product' grouping has a strong linkage to the 'direction' and 'coordination' guidance-services modelled in the Canvas, whilst the eTOM 'Enterprise Management' grouping is somewhat shared across the 'direction' and 'validation' guidance-services; the eTOM 'Operations' grouping summarises what happens within the nine-cell core of the Canvas. Note too the strong emphasis in eTOM on 'Customer': this would especially be linked to the Canvas 'Customer-Relations' and flows on the Canvas.

I summarised a mapping for ITIL onto the Enterprise Canvas in Part 7[153] of this series. At the topmost level (row-1) there might be a single Canvas for 'service-management', but at row-2, where relationships come into the picture, we would need to model each of the five main service-groups as a separate entity – in other words a separate Canvas for each. We would then devolve downwards as required, detailing each of the services first in generic (row-3) and then more context-specific (row-4) form.

[153] http://weblog.tetradian.com/2010/07/08/enterprise-canvas-pt7/

The **ISO-9000**[154] quality-system international standard aligns almost exactly with the Enterprise Canvas layers:

- ISO-9000 'vision' layer: Enterprise Canvas row-0, row-1
- ISO-9000 'policy' layer: Enterprise Canvas row-2, generic part of row-3
- ISO-9000 'procedure' layer: Enterprise Canvas detail of row-3, generic of row-4
- ISO-9000 'work-instruction' layer: Enterprise Canvas detail of row-4, row-5

Balanced Scorecard[155] is a useful frame to apply across the Canvas, but especially in the Value-Management cell, which is where performance data and post-transaction records should naturally accumulate. It may be useful to augment or review Balanced Scorecard's four standard perspectives – Financial, Internal Business Processes, Learning & Growth, and Customer – with other themes from the enterprise vision and values, as appropriate. The single most crucial point, especially in a commercial for-profit business, is to loosen the stranglehold of finance-centric metrics, and to instead cover the full range of meanings of 'value' that apply in the shared-enterprise.

Standard review-processes and process-improvement models such as Deming/Shewhart **PDCA**[156] (Plan/Do/Check-

[154] http://en.wikipedia.org/wiki/ISO_9000
[155] http://en.wikipedia.org/wiki/Balanced_scorecard
[156] http://en.wikipedia.org/wiki/PDCA

/Act), **Root Cause Analysis**[157], **Value-Stream Mapping**[158] and, where appropriate, **Six Sigma**[159], can all be usefully applied to the models developed on a Canvas, and in some cases modelled in relation to the activities shown on the Canvas. Conceptually and operationally, they belong as part of what the Canvas models as its guidance-services – in practice usually a collaboration between the 'validation' and 'coordination' services. Note that in general Six Sigma should *not* be used unless the service on the Canvas handles literally millions of nominally-identical events.

Summary

There are many, many other types of models used in enterprise-architecture, business-architecture, process-architecture, application-architecture, data-architecture, security-architecture, IT-architecture, values-architecture and so on. In many if not most cases they can either be used via direct cross-mapping to the Enterprise Canvas – as per context-space mapping[160] – or as a secondary analysis of what shows up in cross-maps on the Canvas. The simplest suggestion is: try it and see. :grin:

So, that's it: the Enterprise Canvas. Over to you: perhaps have a play with it, let me know what you think of it, any

[157] http://en.wikipedia.org/wiki/Root_cause_analysis
[158] http://en.wikipedia.org/wiki/Value_stream_mapping
[159] http://en.wikipedia.org/wiki/Six_Sigma
[160] http://tetradianbooks.com/2010/05/everydayea/

suggestions for improvement, and so on. And many thanks for staying with me this far!

Source (Tetradian weblog)

- *Date*: 2010/07/10
- *URL*: enterprise-canvas-pt8[161]
- *Comments*: (none)
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[161] http://weblog.tetradian.com/enterprise-canvas-pt8

The Enterprise Canvas: Summary and Index

There's a lot of material in all of those articles on the Enterprise Canvas, so here's an index:

- Introduction[162] – a somewhat quirky lead-in to the background and reasoning behind the Enterprise Canvas model
- Part 1: Context and Value[163] – introducing the 'vertical' dimension of the Canvas
- Part 2: Market and Supply-Chain[164] – introducing the 'horizontal' dimension for the Canvas, and crosslinking it to the nature of the market
- Part 3: Owners and Managers[165] – incorporating crosslinks to other stakeholders in the 'vertical' dimension
- Part 4: Layers[166] – describing the structural changes that occur when modelling the crossmaps between layers of abstraction and implementation

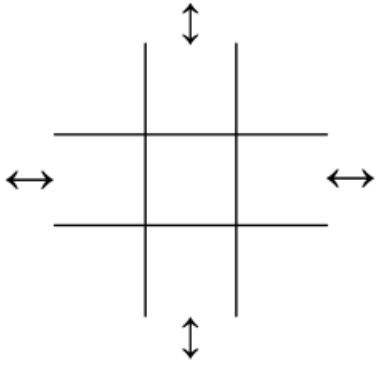
[162] http://weblog.tetradian.com/2010/07/02/quest-for-enterprise-canvas/
[163] http://weblog.tetradian.com/2010/07/02/enterprise-canvas-pt1/
[164] http://weblog.tetradian.com/2010/07/03/enterprise-canvas-pt2/
[165] http://weblog.tetradian.com/2010/07/04/enterprise-canvas-pt3/
[166] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

- Part 5: Recursion, Flows and Systems[167] – refining the utility of the Canvas with systems-theory and systems-models
- Part 6: Models[168] – initial description of how to use the Canvas to model services and structures within architecture-practice
- Part 7: Patterns[169] – four examples of typical patterns described by the Enterprise Canvas, in different business-contexts and at different architectural layers
- Part 8: Integration[170] – describing crosslinks and usages with a range of other model-types commonly used in enterprise-architectures
- Summary and index[171] – this post

For the model itself, there are at least four different versions, depending on how you're working and what you need to model.

The minimalist 'back of the napkin' version:

[167] http://weblog.tetradian.com/2010/07/06/enterprise-canvas-pt5/
[168] http://weblog.tetradian.com/2010/07/07/enterprise-canvas-pt6/
[169] http://weblog.tetradian.com/2010/07/08/enterprise-canvas-pt7/
[170] http://weblog.tetradian.com/2010/07/10/enterprise-canvas-pt8/
[171] http://weblog.tetradian.com/2010/07/10/enterprise-canvas-summary/



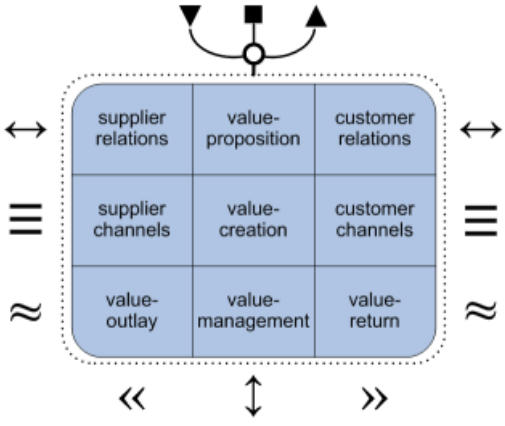
Enterprise Canvas: minmalist

The 'brick', or grid version – the core nine-cell layout at the intersection of enterprise and value-web:

supplier relations	value-proposition	customer relations
supplier channels	value-creation	customer channels
value-outlay	value-management	value-return

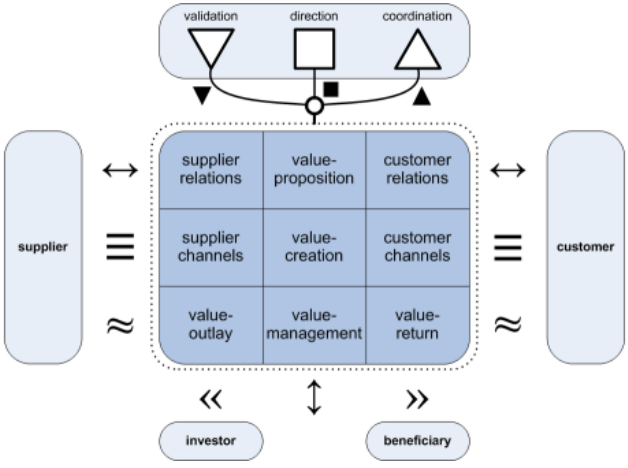
Enterprise Canvas: complete frame

The 'beetle' version, including symbols for each of the related flows:



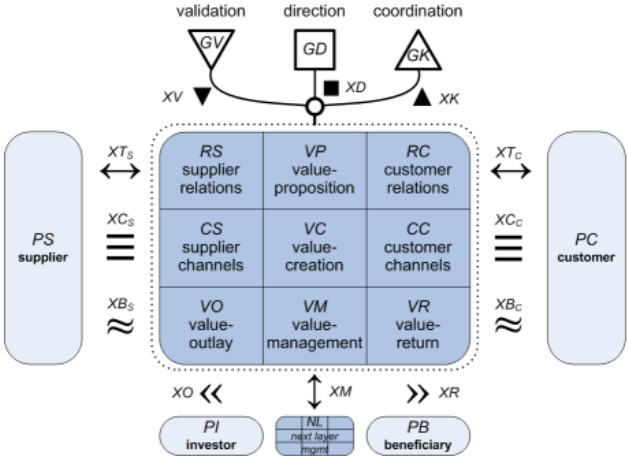
Enterprise Canvas: complete, with flows

The 'robot' version, also summarising all of the stakeholders:



Enterprise Canvas: complete, with stakeholders

And another variant of the 'robot' version ('robot chicken'?), with two-letter codes for each cell, interface and stakeholder:



Enterprise Canvas: complete (coded)

Share and Enjoy, perhaps? And let me know how you get on with it, if you would?

Source (Tetradian weblog)

- *Date*: 2010/07/10
- *URL*: enterprise-canvas-summary[172]
- *Comments*: 7
- *Categories*: Business, Complexity / Structure, Enterprise architecture

[172] http://weblog.tetradian.com/enterprise-canvas-summary

- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

The Enterprise Canvas: a Really Simple Summary

The full Enterprise Canvas[173] model is a complex beast, with many ideas, many layers, many ramifications and side-themes: it can perhaps seem daunting at first. Yet when we strip it right down to its bare essentials, it's actually very simple indeed – and its real power comes from that underlying simplicity. So here's a Really Simple Summary of the ideas behind the Enterprise Canvas:

Everything in the enterprise is a service.

The Enterprise Canvas is a generic map to describe any service, anywhere in the enterprise, together with its interdependencies and flows.

The Enterprise Canvas therefore provides a consistent means to model anything, anywhere in the enterprise.

To give a bit more detail, to make that Really Simple Summary more usable in practice:

Everything exists within one infinite ecosystem, which we might label 'the universe'.

[173] http://weblog.tetradian.com/2010/07/10/enterprise-canvas-summary/

For practical reasons – and for sanity's sake – we usually restrict our view to a much smaller subset of that 'the everything'. (We do always need to remember that it actually *is* 'the everything', though.)

One useful option, especially for organisations, is to select the subset that describes that part of the ecosystem within which the organisation operates. This 'extended-enterprise' (or 'enterprise', for short) is always larger than the organisation itself, and coalesces around a single idea or descriptor, usually referred to as the 'vision' for the enterprise.

Within that enterprise, we assert that *every entity represents a service.*

Every entity delivers services, provides services, consumes other services. The ecosystem is made viable by this constant interchange of services.

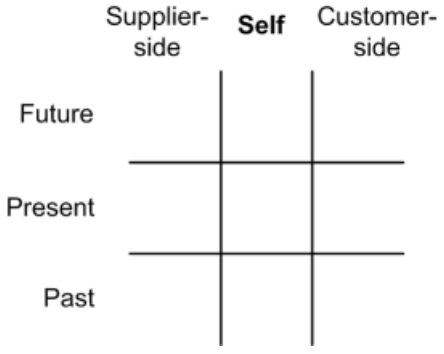
This interchange occurs at every level. Everything is a service, from a whole organisation to the faucet in the bathroom or a single line of program-code.

Everything is a service.

For each entity, it can be useful to divide the view into three partitions, in terms of *role* and *relationship*: the role, function and services of the entity itself; its relationships with the entities that provide the services that this entity consumes ('supplier-side'); and its relationships with the entities that consume the services that this entity provides ('customer-side').

For each entity, it can also be useful to divide the view into three other partitions, in terms of time: what is intended to happen ('future'); what is actually happening ('present'); and what has happened ('past').

These two sets of views are orthogonal to each other. We can therefore map this as a three-by-three matrix:



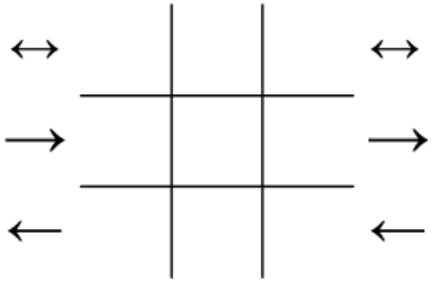
Enterprise Canvas: 3x3 matrix

The relationships with other entities are symmetrical in the sense that *every* entity shares the same pattern: the only difference between 'supplier-side' and 'customer-side' is the main direction of service-flow relative to the entity that is our current focus of attention.

The 'future'-oriented relationships are essentially peer-to-peer, and bidirectional.

The 'present'-oriented relationships are mainly about 'supply-chain' transfer of goods and services from supplier to self, or self-to customer (i.e. left-to-right on the Canvas).

The 'past'-oriented relationships are mainly about balancing the supply-chain transfer via a 'backchannel' from customer to self, and self to supplier (i.e. right-to-left on the Canvas).

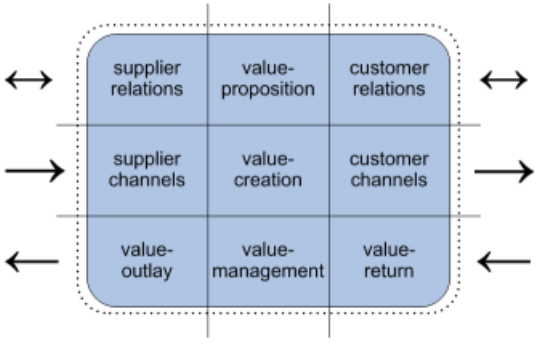


Enterprise Canvas: 3x3 matrix plus flows

We can thus describe the overall entity in terms of nine subsidiary 'cells' or sets of related activities:

- *supplier-side/future*: build and maintain relationships with potential and/or actual 'supplier' service-provider entities, about things that need to happen in the future – ***supplier-relations***
- *supplier-side/present*: receive goods and/or services from 'supplier' entities – ***supplier-channels***
- *supplier-side/past*: provide balance or compensation to 'supplier' entities (e.g. pay for goods) – ***value-outlay***
- *self/future*: identify what this entity will do and deliver, aligned to the overall enterprise purpose and values – ***value-proposition***

- *self/present*: take all actions necessary to create and deliver the goods and/or services specified in the value-proposition – ***value-creation***
- *self/past*: ensure the appropriate functioning of the overall entity, balancing past, present and future – ***value-management***
- *customer-side/future*: build and maintain relationships with potential and/or actual 'customer' service-consumer entities, mainly about what should happen in the future – ***customer-relations***
- *customer-side/present*: deliver goods and/or services to 'customer' entities – ***customer-channels***
- *customer-side/past*: receive balance or compensation from 'customer' entities (e.g. payment for goods) – ***value-return***



Enterprise Canvas: summary

Each of these 'cells' delivers its own services to the entity, and could thus, recursively, be represented by and

described on its own Enterprise Canvas.

Each entity may be described in terms of various layers[174] on a spectrum between most-abstract (the enterprise as a whole) to most-concrete (the detailed-past).

Note that ultimately all service-boundaries are arbitrary, and in most cases exist for descriptive and/or administrative convenience only. Within the overall ecosystem, any or all of its services may be recombined and reconfigured in an infinity of alternate ways. The key criterion for success is not 'correctness', but *effectiveness*:

- *efficient*: optimises use of resources, minimises wastage of resources
- *reliable*: predictable, consistent, self-correcting, supports 'single source of truth' etc
- *elegant*: clarity, simplicity, consistency, self-adapting for human factors
- *appropriate*: supports and optimises support for business purpose
- *integrated*: creates, supports and optimises synergy across all systems

Effectiveness occurs when everything supports everything else, all the way up to the enterprise vision or purpose.

The Enterprise Canvas does not attempt to describe every aspect of every service. Its role is to provide a consistent

[174] http://weblog.tetradian.com/2010/07/05/enterprise-canvas-pt4/

base-frame to link descriptions together into a unified whole. It would generally be used in conjunction with many other model-types, for example:

- use VPEC-T[175] to model each of the flows to and from the entity in focus
- use modified-Zachman[176] to model the assets, functions, locations, capabilities, events and decisions in each flow, in each cell and in the entity as a whole
- use SWOT[177] to assess strengths, challenges, opportunities and risks in each flow, cell and entity

The Enterprise Canvas will also work well with other techniques for SOA[178] (service-oriented architecture) and any other cross-enterprise concerns such as quality, security, safety and environment.

There's a lot more[179] to it than just the above, of course, but I hope this 'really simple summary' will give you enough to get started?

Source (Tetradian weblog)

[175] http://en.wikipedia.org/wiki/VPEC-T
[176] http://tetradianbooks.com/2008/12/silos-frame-ref/
[177] http://en.wikipedia.org/wiki/SWOT_analysis
[178] http://en.wikipedia.org/wiki/Service-oriented_architecture
[179] http://weblog.tetradian.com/2010/07/10/enterprise-canvas-summary/

- *Date*: 2010/07/19
- *URL*: enterprise-canvas-really-simple-summary[180]
- *Comments*: (none)
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: business architecture, Enterprise architecture, enterprise canvas, metamodel, methodology

[180] http://weblog.tetradian.com/enterprise-canvas-really-simple-summary

An Enterprise Canvas update: 'value-governance'

An important email for me this morning, from management consultant Ray McKenzie, that's triggered off a significant re-think on the role and label for one of the nine main cells in the Enterprise Canvas model:

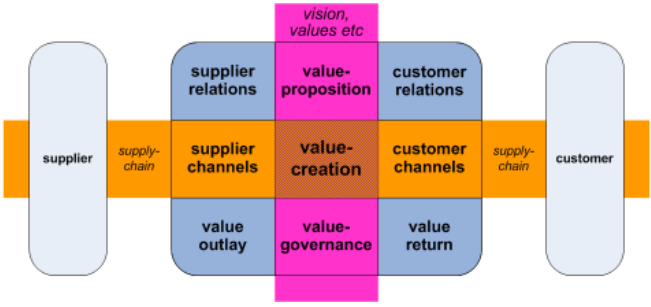
> While you labelled the bottom row of 'Enterprise Canvas' as Value, somehow as I read through the material I kept thinking 'Governance', not sure if this was your intent or just my imagination.

And yes, he's right: of *course* it's 'Value-Governance' – of *course*! Why on earth didn't I see that before? I knew that 'Management' wasn't right when I wrote it, but I couldn't find the right alternative. Yet there it is, staring me right in the face: of *course* it's 'value-governance'! – in fact, given its role, at the intersection of 'our value' and 'the past', it really couldn't be anything else. Using the term 'value-management' for that single cell gives it completely the wrong flavour – in fact that's the main function of the

'value-direction' cell in the somewhat-external 'guidance' group, so it's already covered elsewhere. Management is something that happens *throughout* the service, not just in one place – but it *is* fair enough to say that overall governance-activities for a service tend to be concentrated in one subdomain of that service, enacted via its own domain-specific roles.

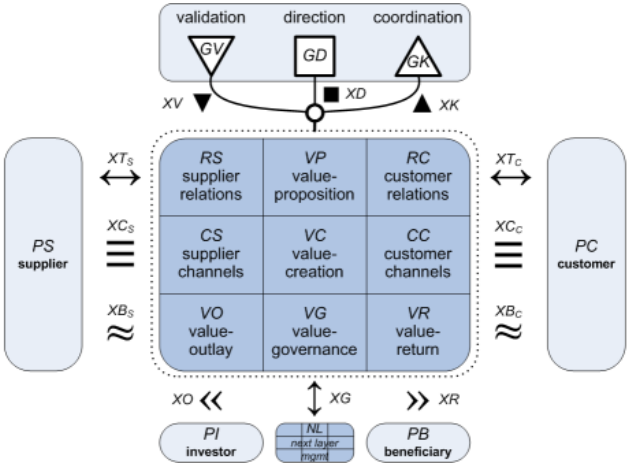
Value-governance makes sense here in both directions on the Canvas' grid. On the vertical 'our value' axis, 'value-proposition' deals mainly with what we *will* do (future); 'value-creation' is concerned with what we *are* doing (present); whilst 'value-governance' looks at what we will do, but perhaps even more at what we *have* done (past), to ensure that they match up correctly. And in the horizontal value-web axis, 'value-governance' sits on the backchannel – completions and the past – to hold the balance between what comes in as 'value-return' from the customer-side of transactions, and what goes out as 'value-outlay' on the supplier-side.

Hence duly-amended versions of the key diagrams – first, the 'service-cross' version of the 'brick':



Enterprise Canvas: service-cross (update)

...and the 'robot-chicken':



Enterprise Canvas: complete, coded (update)

Not many people use the shorthand two-letter codes for cells and flows, but these should change from VM to VG (for the cell-label), and XM to XG (for the flow-label). The

XG flow now focusses primarily on matters relating to governance between the layers (rather than getting mixed up with overall management and direction, which should probably be associated more with the XD guidance-flow).

In all, this cleans up an inconsistency that had been bugging me for ages in the structure of the Canvas, but I hadn't been able to see what was wrong or why. Hence, once again, many thanks to Ray McKenzie for this.

Source (Tetradian weblog)

- *Date*: 2010/08/30
- *URL*: enterprise-canvas-update-value-governance[181]
- *Comments*: (none)
- *Categories*: Business, Complexity / Structure, Enterprise architecture
- *Tags*: Business, business architecture, effectiveness, enterprise, Enterprise architecture, enterprise canvas, governance, metamodel, methodology, values

[181] http://weblog.tetradian.com/enterprise-canvas-update-value-governance

Metaframeworks in practice, Part 5: Enterprise Canvas

What frameworks do we need, to link across all domains and layers of the enterprise-architecture space? How can we create such frameworks?

This is the fifth and final item in this series of worked-examples of metaframeworks in practice[182] - on how to hack and 'smoosh-together' existing frameworks to create a tool that will help people make sense of a specific business-context:

- *Part 1*: Zachman plus tetradian -> *extended-Zachman*[183]
- *Part 2*: TOGAF plus PDCA plus US-Army AAR -> *iterative-TOGAF*[184]
- *Part 3*: Chinese *wu xing* plus Group Dynamics plus VPEC-T -> core *Five Element*[185]

[182] http://weblog.tetradian.com/metaframeworks-in-practice-intro/
[183] http://weblog.tetradian.com/metaframeworks-pt1-ext-zachman/
[184] http://weblog.tetradian.com/metaframeworks-pt2-iterative-togaf/
[185] http://weblog.tetradian.com/metaframeworks-pt3-five-elements/

- *Part 4*: Jung plus swamp-metaphor plus systems-practice plus Kurtz/Cynefin plus many-others -> *context-space mapping* -> *SCAN*[186]
- ***Part 5*** [this post]: Business Model Canvas plus Viable System Model plus extended-Zachman plus many-others -> ***Enterprise Canvas***; plus Five Element plus Market Cycle -> ***Enterprise-Canvas dynamics***

In accordance with the process outlined in the 'Metaframeworks in practice[187]' overview-post, each of these worked-examples starts from a selected base-framework – in this case, Alex Osterwalder's *Business Model Canvas* – and builds outward from there, to support the specified business-need.

Enterprise Canvas

Business-need

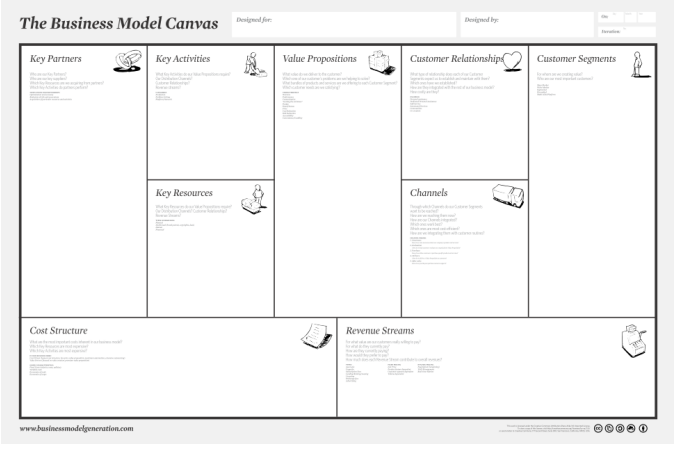
Again no specific business-client on this one: the real driver was just my own repeated frustration at trying to find workable ways to link things together across the whole enterprise-architecture space, from IT to process to people to 'things', and from motivation to execution and all the way back again.

[186] http://weblog.tetradian.com/metaframeworks-pt4-csm-and-scan/
[187] http://weblog.tetradian.com/metaframeworks-in-practice-intro/

There are lots and lots of point-solutions and suchlike, many of which do work well on their own; but the whole space overall is so fragmented that it's almost always a mess trying to make sense of anything that crosses almost *any* kind of boundary. Hence we *really need* something that works here… it doesn't need to be wonderful, it just needs to be able to do it at all!

Metaframework process

Start from Alex Osterwalder's Business Model Canvas[188]:



At first glance, it seems it could be usable as a link-frame across any aspect of enterprise-architecture.

[188] http://en.wikipedia.org/wiki/Business_Model_Canvas

Apply systems-principles such as *rotation*, *reciprocation* (balance) and *recursion*:

- the frame is obviously usable as a *rotation* – a set of 'domains' that provide a consistent and systematic suite of views into a context
- the frame is *asymmetric* ('unbalanced') in that the channels and interfaces on the 'customer-side' are not replicated on the 'partner-side'
- the frame is not inherently designed for *recursion* – it aims to apply at the business-model level only (e.g. Zachman rows 2-3)

In short, it's a very useful tool for business-architecture, but does not *in itself* appear to support a broader use. Yet for me there's still a strong feeling that it's a good place to start.

Move sideways somewhat: think about the 'customer-side' of Business Model Canvas, which seems more complete.

Think about the usual emphasis on transactions.

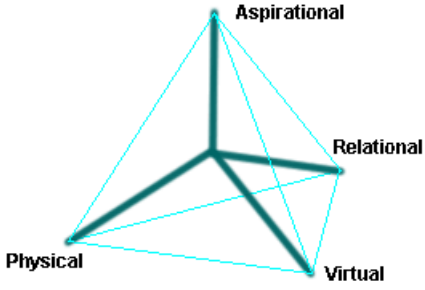
Connect that to the Cluetrain[189] catchphrase *"Markets are conversations"*.

Link to that a later add-on comment by one of Cluetrain's originators, Doc Searls, that markets are also built on relationships[190].

[189] http://www.cluetrain.com

[190] http://weblog.tetradian.com/relationship-economy/

Link from there to the *tetradian* model (see Part 1 of this series[191]), which would in turn suggest that markets are also held together by aspirations or shared-purpose:

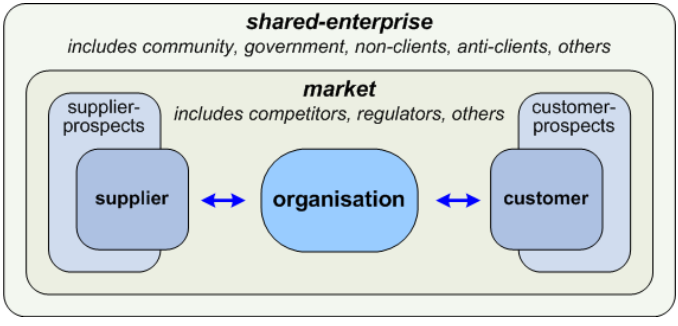


Apply this as a model of the overall enterprise-space:

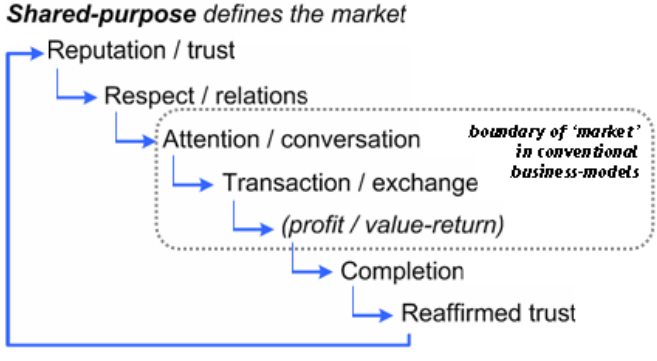
- *physical* and/or *virtual* (information): the ***transactions*** of the enterprise
- *virtual* (conversation) and/or *relational*: the ***market*** of the enterprise
- *relational* and/or *aspirational*: the ***promise*** of the enterprise

Or, in visual form:

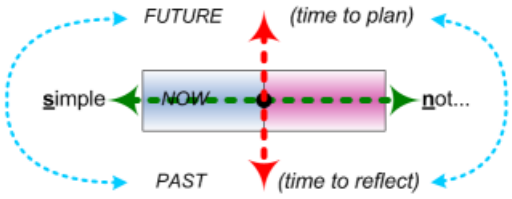
[191] http://weblog.tetradian.com/metaframeworks-pt1-ext-zachman/



Or, as a service-cycle, where ***trust*** is a measure of match to shared-aspirations:



Note that this splits into three segments: *before*, *during* and *after* the main core of what most organisations think of as 'their market'. In SCAN terms, it's what happens *at* the point of action (now: 'transaction'), and either side of that point of action (future: '*before*' and past: '*after*'):



Note that this also aligns quite well with the right-hand (customer-facing) side of Business Model Canvas:

- *before*: Customer Relationships
- *during*: Channels
- *after*: Revenue Streams

Stop there for the moment: let this brew for a while…

Think about products versus services. Business Model Canvas seems to emphasise products, but in essence products are just proto-services, a means through which we can deliver self-service.

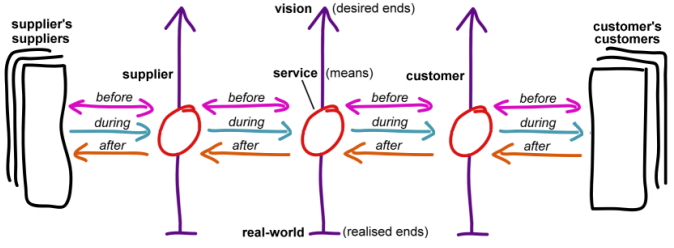
Consider the idea that we could describe *everything* in the enterprise in terms of services.

By *serendipity*, come across an article that describes effects of global climate-change in terms of impacts on 'ecosystem-services' provided by rainforests and suchlike. It emphasises interdependencies across the whole ecosystem, describing all inter-relationships in terms of mutual services that balance overall across the ecosystem. Every service is a provider ('supplier') and consumer ('cus-

tomer') of other services: inherently-symmetric relationships across either side of each service.

Cross-link this with Daniel Pink's work on motivation in human enterprises[192]: in particular, the importance of purpose – the 'vision' or 'promise' that holds the enterprise together and drives it into action.

If we link all of these together, this suggests that we could view everything in the enterprise in terms of services that connect 'vertically' in terms of shared purpose and associated values; and that connect 'horizontally' in terms of exchange of 'that which is valued', with relationships that link before, during and after the main-transactions. Or, in visual form, as the classic supply-chain:

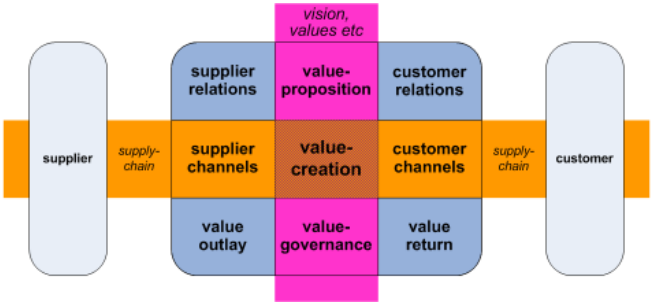


Rethink the functions of the service in terms of a simple 3×3 matrix:

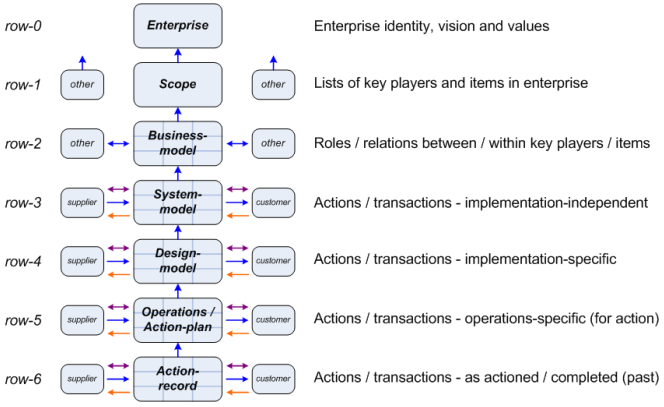
- supplier-facing, self, customer-facing
- before, during, after

[192] http://www.danpink.com/books/drive

Which, in visual form, gives us:



Apply the extended-Zachman from Part 1 of this series[193]. Each service could be described in terms of any one of the extended-Zachman levels-of-abstraction:

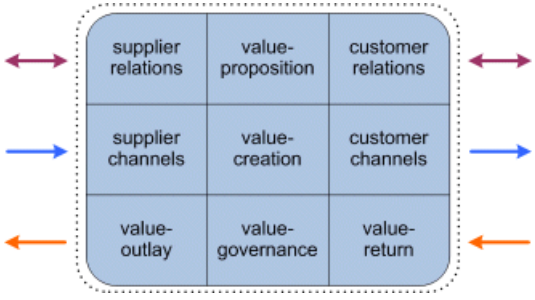


And the *service-content* for each service in terms of a single row within the extended-Zachman frame:

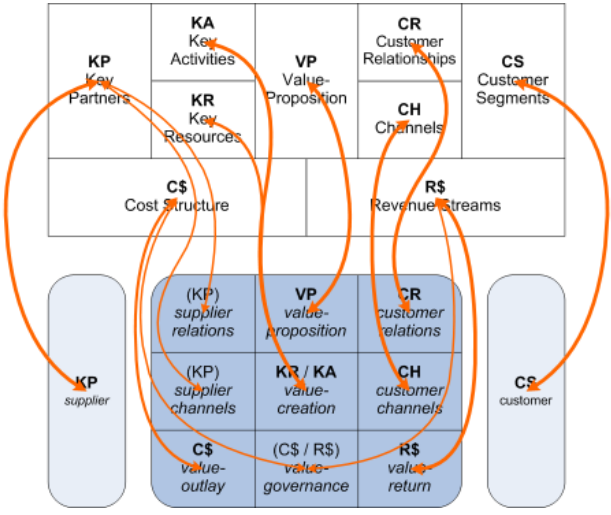
[193] http://weblog.tetradian.com/metaframeworks-pt1-ext-zachman/

	Assets	Functions	Locations	Capabilities (action)	Capabilities (agent)	Capabilities (skill-level)	Events	Decisions	
Asset-types	*What*	*How*	*Where*	*(Who)*			*When*	*Why*	*Decision/skill-types*
Physical	Phys	Phys	Phys	Phys	Phys	Rules	Phys	Rules	Rule-based
Virtual	Virtual	Virtual	Virtual	Virtual	Virtual	Algor'm	Virtual	Algor'm	Algorithmic
Relational	Reln	Reln	Reln	Reln	Reln	Guideln	Reln	Guideln	Guidelines
Aspirational	Aspn	Aspn	Aspn	Aspn	Aspn	Princpl	Aspn	Princpl	Principle-based
Abstract			Time						

And if we summarise the service itself visually as follows:



Then, if we rethink Business Model Canvas somewhat in terms of *reciprocation* (symmetry and balance), we would end up with a crossmap between Business Model Canvas and this 'service-canvas' somewhat as follows:



It's a bit messy, but it works. Sort-of: enough for sensemaking about services at any level, anyway. Yet it still doesn't quite feel complete…

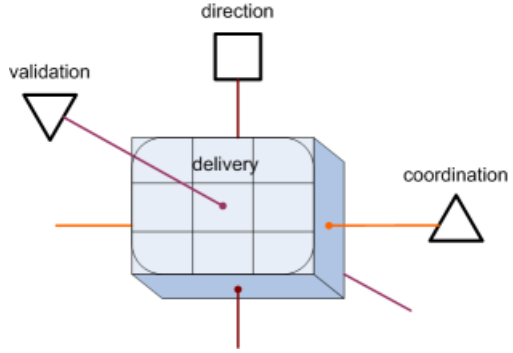
Link across to work I'd done more than a decade or more ago on 'viable systems', based on Stafford Beer's Viable System Model[194] (VSM). If systems are to be viable, especially over the longer term, then there are distinct interrelationships that are needed between each 'component' in the system. Apply *recursion* and/or *reflexion* to go to any scale, and reframe 'component' as 'service': hence, in terms of an individual service not so much as 'Viable *System* Model' as 'Viable *Service* Model'.

[194] http://en.wikipedia.org/wiki/Viable_system_model

If so, every service-entity:

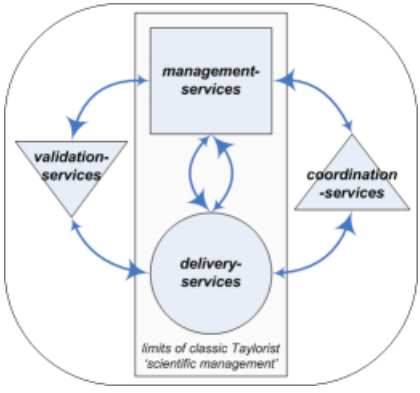
- *delivers* some kind of service (VSM 'system-1′)
- *coordinates* with other services (VSM 'system-2′)
- *directs* and/or is directed ('managed') by other services (VSM 'system-3′, 'system-4′, 'system-5′)
- keeps *on track to value(s)* (VSM 'system-3*')

Or, in visual form:



Or, in terms of services' relationships to each other, as described in my book *The Service-Oriented Enterprise*[195]:

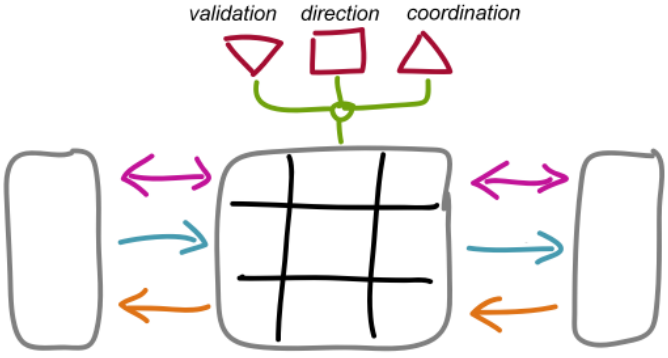
[195] http://tetradianbooks.com/2008/12/services/



Note the limits of Taylorism[196], which often doesn't acknowledge either cross-silo coordination or the need to link to pervasive values. For a viable model, we'd need to include those latter elements.

Add these relationships to a 'back-of-the-napkin' sketch of that 'service-canvas':

[196] http://weblog.tetradian.com/yet-more-on-no-jobs-for-generalists-3a/

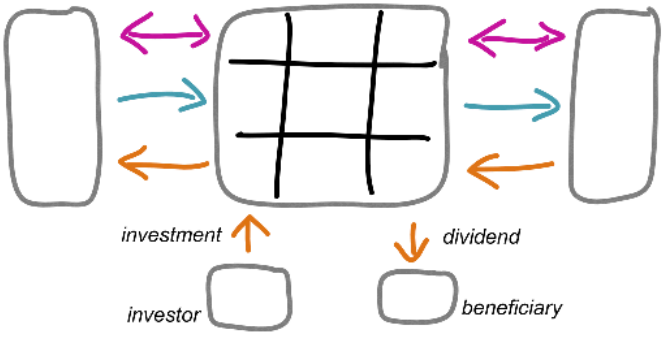


Note too that each service – especially at larger scale – is likely to have its own investors and beneficiaries, who interact via different forms of value[197] than those flowing through the 'horizontal' supply-chain or value-web[198].

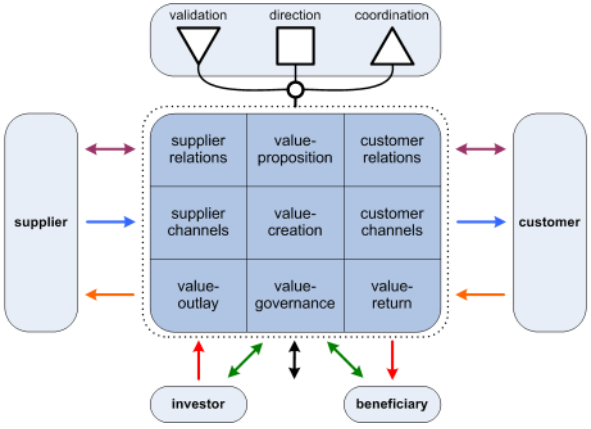
Add these too to a 'back of the napkin' sketch:

[197] http://weblog.tetradian.com/money-price-and-value-in-ea/

[198] http://weblog.tetradian.com/modelling-mixed-value-in-enterprise-canvas/



And if we bring all of these together, into expanded form, we have an **Enterprise Canvas**:

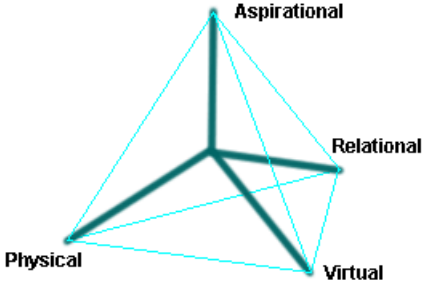


This can be used to summarise the content, relationships and interdependencies of *any* service at *any* level. Whenever we look at a Business Model Canvas, or an entity in a BPMN or Archimate or UML model, or a data-element in a

data-schema, we can use this frame to bridge the gaps and link across all of the different views.

Useful, yet still not enough… what happens *between* services matters every bit as much as the services themselves.

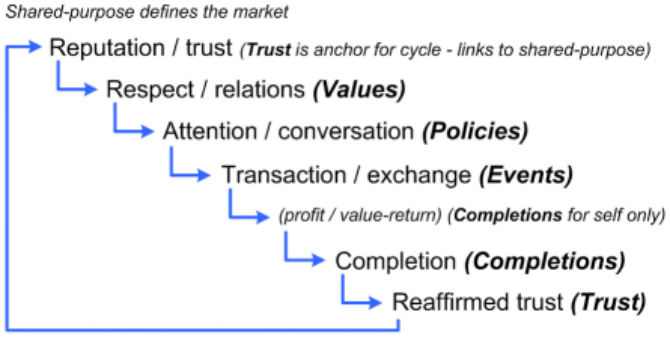
Apply the *tetradian* to those flows or relationships:



What it indicates is the *asset-types* that are exchanged or linked across each of the 'flows' – the almost-infinite variety of asset-type combinations indicated or implied by the left-most column in the *service-content* map:

	Assets	Functions	Locations	Capabilities (action)	Capabilities (agent)	Capabilities (skill-level)	Events	Decisions	
Asset-types	*What*	*How*	*Where*	*(Who)*			*When*	*Why*	*Decision/skill-types*
Physical	Phys	Phys	Phys	Phys	Phys	Rules	Phys	Rules	Rule-based
Virtual	Virtual	Virtual	Virtual	Virtual	Virtual	Algor'm	Virtual	Algor'm	Algorithmic
Relational	Reln	Reln	Reln	Reln	Reln	Guideln	Reln	Guideln	Guidelines
Aspirational	Aspn	Aspn	Aspn	Aspn	Aspn	Princpl	Aspn	Princpl	Principle-based
Abstract			Time						

But what about the timing or sequence of those interactions? For this, we could crossmap the *service-cycle* above to the 'not-quite VPEC-T' frame in the Five Element model in Part 3 of this series[199]:



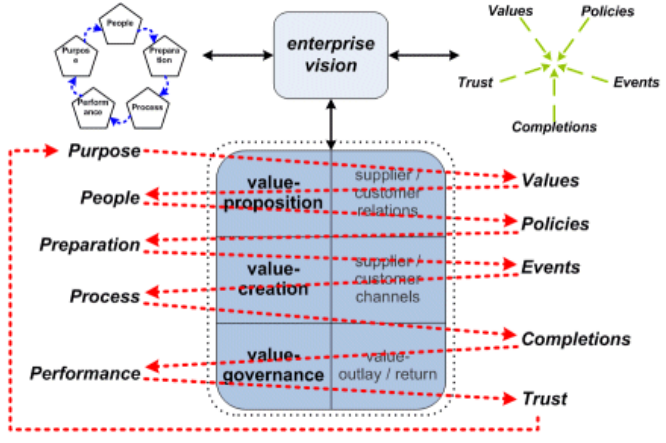
Although each flow could carry *any* type of asset, the sequence does also emphasise different parts of the 'canvas':

- trust / values / policies: *before* – Value Proposition, Supplier/Customer Relations
- policies / events / completions: *during* – Value Creation, Supplier/Customer Channels
- completions / trust: *after* – Value Governance, Value Outlay, Value Return

So if we crossmap the whole Five Element set onto the Enterprise Canvas, turned round to a different view to

[199] http://weblog.tetradian.com/metaframeworks-pt3-five-elements/

distinguish 'inside' from 'outward-facing', we get a frame that summarises the full **Enterprise Canvas dynamics**:



Which, overall, is quite a lot. But it works. Sort-of. :-)

Originator-relations

On Business Model Canvas, I've been careful to make it clear that Enterprise Canvas doesn't purport to be a 'replacement', but a related yet definitely different frame that addresses a significantly different type of challenge: not business-models, but whole-of-context mapping and integration. Being careful and explicit about that point has made relations with Alex Osterwalder a lot easier. We do disagree in some areas – in particular, about the importance of 'between the boxes[200]', which I don't think is addressed

[200] http://weblog.tetradian.com/between-the-boxes/

enough in Business Model Canvas – but otherwise it's all easy, and mutually-supportive.

On this re-use of the extended-Zachman frame, it's much as described in Part 1 of this series: we 'agree to disagree', but the core of it is that Zachman himself is a *really* nice guy who seems to go out of his way to make everything easy and comfortable for everyone.

On VSM (Viable System Model), Stafford Beer is sadly no longer with us: it would have been good to explore this with him. The core difference from 'standard' VSM here is that I've greatly expanded the role of the VSM 'system-3star', from 'random-audit' to full coverage of all values-issues. This does make sense, because the original VSM was designed only to describe *information*-systems and information-flows, whereas the Enterprise Canvas must cover *all* asset-types, flow-types and relationship-types – and an expansion of 'system-3star' fits exactly with the logic of the VSM. Some of the lead VSM practitioners – particularly Patrick Hoverstadt[201] – do disagree with me on this: but it's still a reasonably-amicable 'agree to disagree' rather than a full-blown 'unhappy-originator' problem.

On this adaptation of VPEC-T, it's as described in Part 3 of this series: another careful emphasis that this *isn't* the same as Nigel Green and Carl Bate's original model. (See 'More on 'Not-quite VPEC-T'[202]' for more detail on this.)

[201] http://www.amazon.co.uk/Fractal-Organization-Creating-Sustainable-Organizations/dp/0470060565

[202] http://weblog.tetradian.com/more-on-not-vpect/

The rest of it is 'all my own work', mostly. I do seem to argue a lot with myself, it's true, but otherwise relations with that originator seem fine!

Applications

This is one framework that's out there and being used in a lot of different ways.

It's described in depth in my book *Mapping the Enterprise*[203], and the blog-posts here through which it was developed have been published as the ebook *Enterprise Canvas: the Tetradian weblogs*[204].

It's also appeared here in quite a few other forms and variations, such as the 'This'[205] game[206] – see the post 'Using the 'This' game in EA modelling[207]' for a live example – and as the Enterprise Canvas service-viability checklist[208], using the simplified notation[209] for Enterprise Canvas.

It's starting to gain a much broader user-base: see the post 'Tools in action[210]' for an early example. I'd also recently visited a client who have been using it as a core framework

[203] http://tetradianbooks.com/2010/11/ecanvas/
[204] https://leanpub.com/tp-ecanvas
[205] http://weblog.tetradian.com/this-exploratory-game-for-service-oriented-ea/
[206] http://weblog.tetradian.com/more-on-the-this-game-for-ea/
[207] http://weblog.tetradian.com/using-the-this-game-in-ea-modelling/
[208] http://weblog.tetradian.com/ecanvas-as-service-viability-checklist/
[209] http://weblog.tetradian.com/simplifying-ecanvas/
[210] http://weblog.tetradian.com/tools-in-action/

across the whole of their enterprise-architecture for almost a couple of years now – exactly as I'd hoped and intended right from the start. Feels good.

Overall, quite a good advert for metaframework techniques, I'd say?

Lessons-learned

Probably the single most important lesson-learned here is that over the years – and like just about every other practitioner – I've accumulated quite a broad toolkit of frameworks and model-types: some very interesting and useful things can happen when we put them together in new and different ways!

That's all of the metaframework examples in this series: over to you for any comments on this example, or on the series as a whole?

Source (Tetradian weblog)

- *Date*: 2012/11/14
- *URL*: metaframeworks-pt5-ecanvas[211]

[211] http://weblog.tetradian.com/metaframeworks-pt5-ecanvas

- *Comments*: (none)
- *Categories*: Complexity / Structure, Enterprise architecture, Knowledge
- *Tags*: decision-making, Enterprise architecture, enterprise canvas, frameworks, metaframework, paradigm, sense-making