Using Capabilities in Enterprise Architecture Management

Whitepaper – Version of 2009-12-18

Wolfgang Keller
objectarchitects, Liebigstr. 3, 82166 Lochham, Germany
Email: wk[at-nospam]objectarchitects.de
http://www.objectarchitects.biz

Disclaimer: While the author used his best efforts in preparing this paper, he makes no representations or warranties with respect to the accuracy or completeness of the contents of this paper and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. The advice or strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. The author shall not be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential or other damages.

Abstract

Capabilities are one of the later "hot topics" in Enterprise Architecture Management. They first appeared towards the beginning of this decade in the form of Microsoft's Motion effort to develop a business engineering method. In the meantime, also other large consulting companies use capability based methods of all kinds for IT investment planning and also application portfolio management.

The method is quite simple and straightforward but still powerful. Opposed to that, for whatever reasons the literature on capability assisted enterprise architecture is still quite scarce. This paper will explain a few of the basic mechanisms behind capability based modeling in pattern form. If you have read it, you will be able to judge, whether capabilities are something that can help make your enterprise architecture efforts more effective. The paper demonstrates how to make straightforward use of capabilities and demonstrates how organizations can profit from capabilities today, even though formal research on capabilities is still ongoing.

Introduction

The term capabilities are being used quite a lot in discussions about IT investment planning and also enterprise architecture (see e.g. [Brede+06, CamKal09, Homan08, Ritz08]). It is somehow strange that on the one hand people are using the term in an inflationary fashion – while on the other hand the literature base on capability based IT investment planning in enterprise architecture is rather small. Maybe because a lot of the players, who actively work with capability based planning, still consider it some form of competitive advantage.

This paper will first give a short clarification of relevant terms such as *capability* and *capability model* (or call it list of capabilities) in as much depth as you need it as a practitioner. You will then get account of typical use cases for capability based planning in enterprise architectures and IT planning in general in pattern style. Via the use of three patterns we will demonstrate how capabilities may be used to prepare maps that ease decision making (Heat Mapping, Footprinting) and how you can prepare the base by mixing your enterprise's capability model (Mix the Models)

What are Capabilities?

If you ask three people about capabilities in enterprise architecture or even business capability management (BCM) you might get five answers. In such a situation it is always a good idea to start by looking up the basic terms in a dictionary:

Merriam Webster's Dictionary defines capabilities as follows

> Main Entre: **ca·pa·bil·i·ty**
> **1 :** the quality or state of being capable; *also* **:** ability
> **2 :** a feature or faculty capable of development **:** potentiality
> **3 :** the facility or potential for an indicated use or deployment <the capability of a metal to be fused> <nuclear capability>

Wikipedia [http://en.wikipedia.org/wiki/Capability] gets you a bit closer to the point

> **Capability** is the ability to perform actions. As it applies to human capital, capability is the sum of expertise and capacity. ()

And **Forrester Research** [CamKal09] is even a bit more helpful if it comes to relating capabilities to IT planning.

> A **business capability** defines the organization's capacity to successfully perform a unique business activity. Capabilities:
>
> - Are the building blocks of the business,
> - represent stable business functions,
> - are unique and independent from each other,
> - are abstracted from the organizational model,
> - capture the business interests.

You might say that this is not really revolutionary new. Some people might argue that they do not see a real difference to functional decomposition here. Comparing capabilities, components, and e.g. functional decomposition is indeed a rewarding exercise for ongoing academic research. But as you will see, you need not go through such an exercise in order to get practical benefit from using capabilities in your enterprise architecture management.

Some academics might get pretty "annoyed" at this point. They would argue that capability is not really precisely defined here. From an academic's standpoint this is true – from a practitioner's standpoint it can be observed that various consulting firms and enterprises are successful at using capabilities also without the ultimately precise definition. The above level of precision proved to be sufficient for practical use, facilitating communication between business and IT.

Capabilities and Domains

Others might argue that top-level capabilities are nothing else than the top-level domains of the enterprise architecture of a company. This comes close to the truth. The following figure can illustrate this a bit.

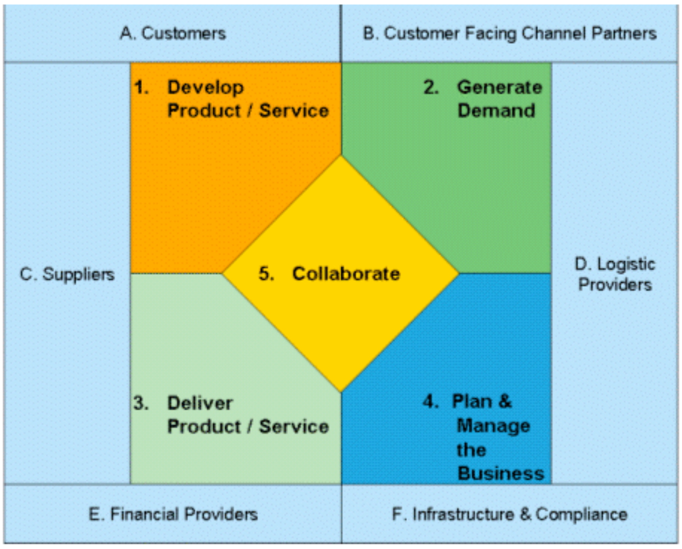

A. Customers
B. Customer Facing Channel Partners
1. Develop Product / Service
2. Generate Demand
C. Suppliers
5. Collaborate
D. Logistic Providers
3. Deliver Product / Service
4. Plan & Manage the Business
E. Financial Providers
F. Infrastructure & Compliance

Figure 1: Microsoft's top level capability model (from [MerTob06]): Compare this to your domain model for your company's domain model, and you will likely be able to produce some form of mapping

If you compare the above rather generic picture, which Microsoft is using for any industry to your own company's domain model you will typically find a mapping between your top-level domain model and the above model. The disturbances are in the details: Let's assume that risk management is a top-level capability of insurance. Then underwriting (the acceptance of risk) can be seen a sub-capability of risk management. If you see the company more from a logistics chain perspective then you would say that Underwriting is an area of your processing capability. You could now spend lots and lots of time discussing whether to put underwriting in the processing corner or the "manage the business" area of the model. Experience says that this is not really worth the effort. At some level in the hierarchy of capabilities, the views will merge again. No matter what the path from the root of a capability tree to an Underwriting capability looks like: You will find an Underwriting Capability in an insurance capability model. And again: For academics this might be annoying. In different capability models for the same industry you will find very similar capabilities at very different places in the capability tree. If you ask practitioners they will tell you that capabilities are still a useful planning instrument as long as one company agrees on a single tree of capabilities for their EAM (enterprise architecture management). This need not be a "universally true" capability tree for an industry. Most practitioners will not invest in finding a universal capability tree. Instead they will use what is available, Mix the Models (see that pattern below) and will start practical work.

Capabilities as a common Language for Business and IT

At this point in the paper you may still argue, that the definition of capabilities is still a bit fuzzy. Hence let's try to go through a few more descriptive statements on capabilities.

Capabilities are not only IT

If you look at how Microsoft describes capabilities [Kumar06] with figures the likes of Figure 2 than you will see that it is the clear intent, that capabilities comprise not only the IT system part of what an enterprise does but capabilities describe a chunk of what the business is able to do in all aspects: People, processes, and also IT – or call it the technology used to support the activities needed to offer a capability.

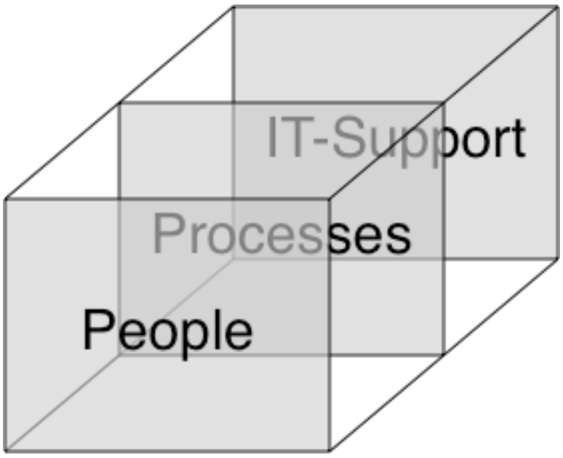


Figure 2: Capabilities are not just technology but cover all aspects: People, processes and the technology used to support them.

In many businesses you will also find a large percentage of capabilities that might not be automated by IT at all. Maybe such areas are just supported by the usual office support solutions the likes of e-mail, word processing and spreadsheets, but no serious specialized application packages. Examples are the "*capability to imagine new products*" or the "*capability to report regulatory compliance*".

As you will also find, capability names start with a verb in most cases: E.g. you will not find a capability "*Contract System*" as you might find it in an IT architecture document. You will find a capability "*Manage Life Insurance Contracts*", which comprises far more than just the IT system, but also People and processes. If you take a small scale or a start-up operation, the capability "Manage Life Insurance Contracts" might not even be fully automated with an IT system.

Other aspects that can be bundled into a capability are (e.g. according to [Kumar06]):

- Performance metrics and cost information for a capability,
- Service level agreements for a capability,
- Compliance criteria,

and other attributes you might need to interconnect the capability in your business architecture models. If you have practiced application portfolio management this will sound familiar to you – except that the subject of interest is a business capability instead of an IT service or an IT application.

What is the Advantage of Managing Capabilities over Managing just Applications?

It is also clear why having capabilities, as the subject of discussion is an advantage over a mere discussion of IT systems or an application portfolio. If you discuss an application portfolio you are in danger to omit important business aspects. You are endangered to discuss and monitor only such capabilities that are already implemented or supported by some IT application.

The things that will contribute to your future business success might not yet be implemented in an application. Or how some enterprise architect would express: Your application portfolio represents the past. To express future need you need additional concepts. Capabilities serve the purpose of reasoning about future sources of success much better than reasoning about merely the application portfolio.

Use Cases for Capability Based Planning

Depending on the hierarchy levels you use for the decomposition of your capabilities you end up with catalogs (or call them trees) of something up to 2.000 and more capabilities – grouped hierarchically from the domains down to maybe 5-7 levels. The question now is: What would you do with such capability models or capabilities list. Typical use cases according to [Ritz08, CamKal09] are:

- ***Investment decisions:*** You can assess your capabilities whether they have any strategic importance for your business strategy and business model. Then you can also check the IT support for them and compare the quality of the IT support to the strategic importance. You will for sure find mismatches: Strategic capabilities with gaps in IT support and unimportant capabilities with best in class top notch IT support.

- ***IT / business alignment:*** The above can also be used, to conduct a gap analysis of the IT support of a whole enterprise. You will find strategically important capabilities that lag in high quality IT support and also waste of money.

- **Outsourcing decisions:** Non-strategic capabilities with high operation costs are prime candidates for any wave of outsourcing.

- ***IT demand management:*** Is more or less a variant of the above. If you match IT demands against capabilities you will be very fast at rating whether they contain more or less unimportant features or whether they drive strategically important capabilities.

The main strong point of capability management is, that it creates a common language for business and IT to discuss common matters of where to direct the money and where not to direct the money in an enterprise.

Pattern Roadmap

In the main part of the paper you will find three patterns that illustrate the handiness of the capabilities approach:

- Heat Mapping: Is a very universal approach to produce easy to read material that helps prepare decisions – e.g. for the above use cases of capability based IT planning. Heat maps show "*hot spots*" in your IT landscape and help you communicate them.

- Footprinting: Is a variant of Heat Mapping that can be used for application planning and comparing applications in a setting with e.g. many subsidiaries that all need application support. Applications have a footprint that can be expressed in terms of capabilities. By making such footprints visible, you can easily compare applications and make their scope easily debatable.

- Mix the Models: Shows you how to obtain a capability list that you will need as the base for all further planning using capabilities. In most cases you will not be able to download a "*ready to use*" capability list but you will have to compile one for your purposes. Mix the Models shows you how to typically do that.

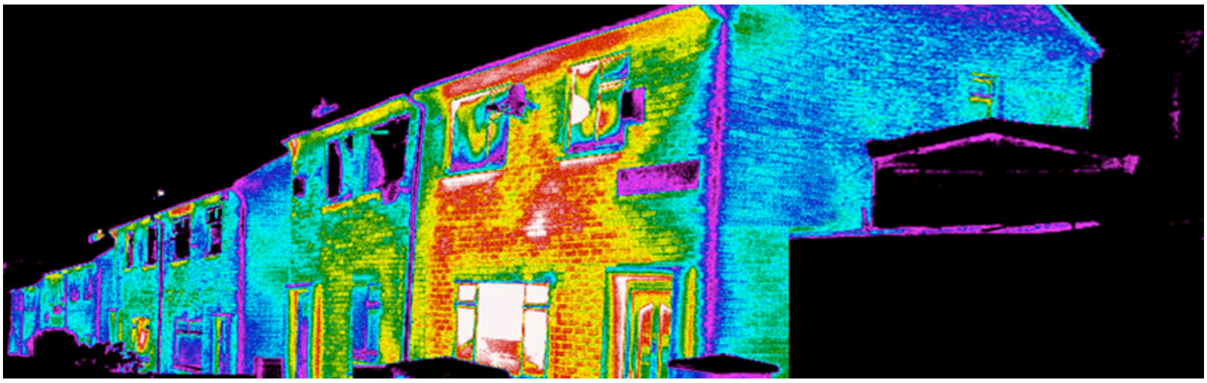
Normally you would start with the capability list. But as this is the most exhausting part we start by showing you the handy things that can be done with capability based reasoning before we show you the effort that you will need to play on that piano.

The Patterns

Pattern: Heat Mapping

Example

Imagine you want to know the spots where your house could use some investments in additional insulation. You could call a so-called infrared building survey.



Now imagine you want to plan investments in an application landscape. You want to know where it might be worth putting money. Given hundreds of applications, maybe more than a thousand capabilities where do you start? You need something to connect you strategy to capabilities and later applications. You need a quick bird's eye's view where to start. You need to communicate facts to your management in a visual, easy to understand style.

Problem

What is the equivalent to an infrared building survey in Enterprise Architecture or IT investment planning?

Forces

Managers want one slide – technically oriented architects want the details: For management meetings like investment committees, it is quite handy to have arguments ready on a single slide. At least never on a 10 square feet poster. On the other hand technical managers, like enterprise architects want accurate facts. They want things to be true or right. Based on detailed facts. Technically minded people would typically say that PowerPoint slides used by management are inaccurate, while managers would say that the details provided by the technical people would blur their vision and hence hamper their ability to decide.

Sometimes also managers want a drill down: The fact that managers want a one-slide overview does not mean that they would not like to have all the facts at hand if they feel like it. Hence whatever planning method should allow drill downs like thermography allows zooming.

Time consumption versus accuracy: The “higher the resolution” of the images that also allow zooming in, the more expensive the technology to make the pictures. This holds for thermography, photography, and this also holds for the use of capability based heat maps in investment planning. You can apply any level of functional decomposition in order to increase accuracy. The drawback is raising costs.

Solution

Produce a so-called heat map of your capability landscape.

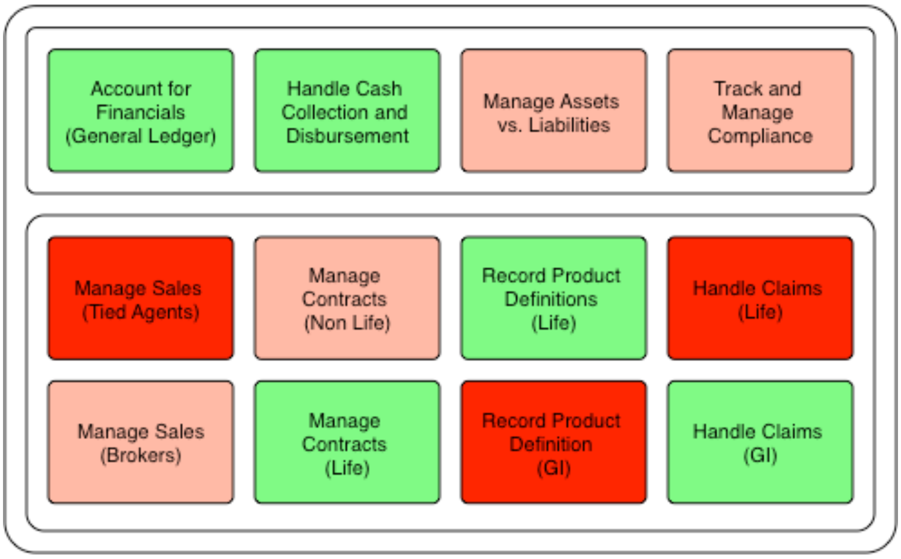

Account for Financials (General Ledger)
Handle Cash Collection and Disbursement
Manage Assets vs. Liabilities
Track and Manage Compliance
Manage Sales (Tied Agents)
Manage Contracts (Non Life)
Record Product Definitions (Life)
Handle Claims (Life)
Manage Sales (Brokers)
Manage Contracts (Life)
Record Product Definition (GI)
Handle Claims (GI)

Figure 3: Heat map at Domain Level – could also be refined at capability level. Read color coding could e.g. stand for high maintenance cost combined with low customer satisfaction for the implementation and support of a capability.

You need not base heat mapping on a catalog of capabilities. You can also use it on a catalog of applications arranged in an EAM application map or any other set of interesting items, but we will use capabilities for the purpose of this paper. What you can use as a base to produce such heat maps is a list of capabilities together with a set of attributes that you use to go through the situation of your company in a bunch of expert sessions, where experts will e.g. tell you how important they rate a capability for your enterprise’s success.

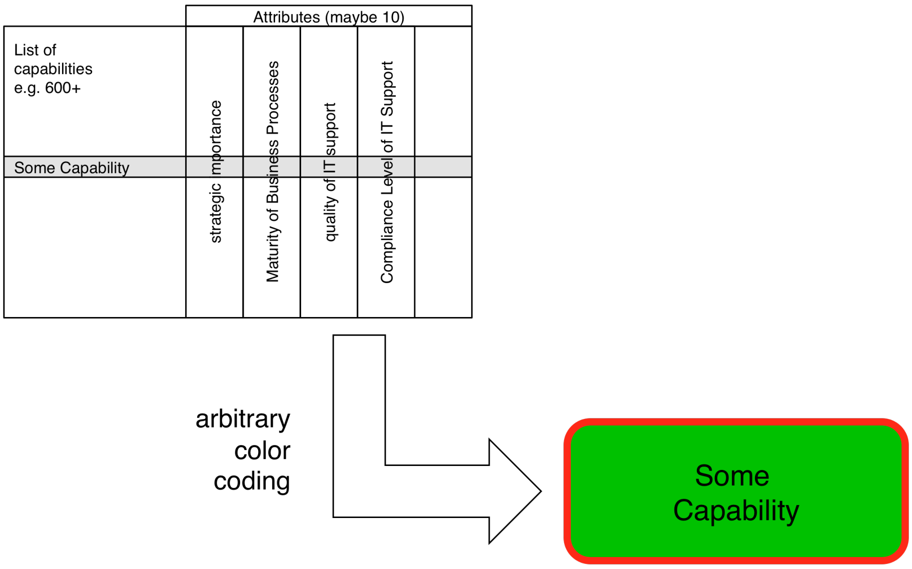


Figure 4: From evaluated Capabilities to a Heat Map

Management could judge whether the capability has any strategic importance. Users can say, what they think about the quality of IT support. Enterprise architects can contribute how many apps are used to implement the capability describing the level of redundancy.

Once you have such an information base, you can use it to generate far more than one heat map. If you define that

Color on map = some function ({attributes of capability})

then you can generate an arbitrary set of heat maps, depicting different aspects, depending on the attributes your management is interested in.

Variants

Heat mapping allows a high variability in itself. You can define arbitrary sets of attributes that you want included in the evaluation and you can define an arbitrary evaluation function that assigns colors or other map attributes to the graphical representation of the capability on a heat map.

Microsoft Motion frequently uses border color to indicate the result of a second evaluation function [HeatMap06]. alfabet also makes the size of boxes variable to indicate e.g. application costs [CamKal09]. There are hardly any limits to one’s fantasy.

Consequences

You can satisfy both: managers and technical people if you invest enough effort in evaluating capability catalogs down to the level that you want or need for a decision. Just evaluate the capabilities down to the level that you need considered.

You can drill down to an arbitrary level – as deep as you have you capabilities modeled.

The method is ***as time consuming or accurate as you want it to be***. You can start with a high level of capabilities and model or evaluate deeper where you assume a “hot spot”.

Related Patterns

Heat Mapping can be refined to various other usages: One has been documented below as Footprinting.

Known Uses

The Microsoft Business Engineering Method set, also known as Motion [Homan08, HeatMap06] makes extensive use of configurable heat maps. alfabet also implements heat mapping, as has been demonstrated in [CamKal09]. Besides, the Business Domain Based Capability Roadmap [TUM09] is a specialization of Heat Mapping used by companies such as Nokia and Siemens. McKinsey also makes extensive use of Heat Maps and capability based planning (see example maps in [Ritz08]).

Pattern: Footprinting

Example

Imagine you have to execute a globalization strategy for a large scale financial service company. You will have to synchronize operations in regions such as Northern America, LATAM (Latin America), EMEA, (Europe Middle East Africa), APAC (Asia Pacific). You will need to compare solutions that have grown across your subsidiaries in the various regions. Those subsidiaries have completely different histories. Hence they will have different solutions, implementing different capabilities. Besides you also evaluate packages from software vendors.

Problem

How do you get a sufficiently accurate overview in a scenario where maybe a dozen systems implements a few hundred capabilities each?

Forces

Ease of use versus sufficient accuracy: a method that allows quick comparisons should also allow enough accuracy so that you do not end up with faulty decisions because the tool was hiding important facts. Ideally you should be able to drill down to the lowest level of detail that you have judged during evaluation, if you need or want to.

Speed versus accuracy: Management is impatient – Engineers often want detail and time in order to prepare optimum decisions. A method that allows comparing e.g, software platforms should be able to support both: Quick, dirty, and fast and also slower, and accurate. Maybe the quick check should be able to evolve into more detail later.

Solution

Compare the solution footprints …

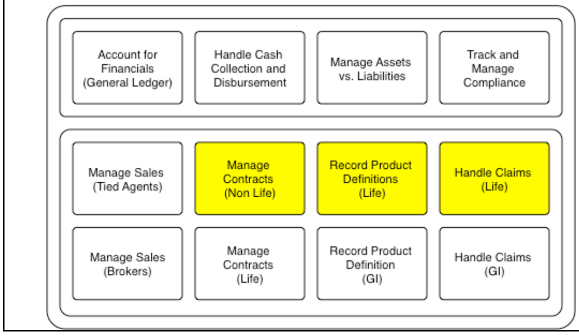
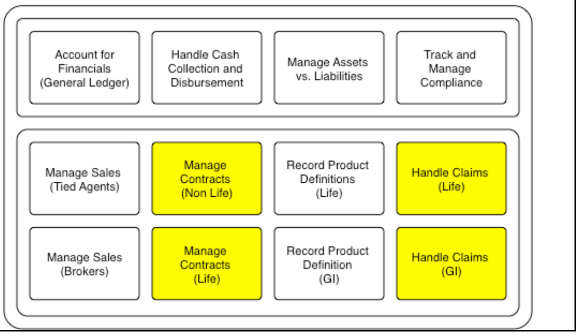
System A Life contract administration including product server capability	System B Life and non-life contract administration excluding product server capability
 Account for Financials (General Ledger) Handle Cash Collection and Disbursement Manage Assets vs. Liabilities Track and Manage Compliance Manage Sales (Tied Agents) Manage Contracts (Non Life) Record Product Definitions (Life) Handle Claims (Life) Manage Sales (Brokers) Manage Contracts (Life) Record Product Definition (GI) Handle Claims (GI)	 Account for Financials (General Ledger) Handle Cash Collection and Disbursement Manage Assets vs. Liabilities Track and Manage Compliance Manage Sales (Tied Agents) Manage Contracts (Non Life) Record Product Definitions (Life) Handle Claims (Life) Manage Sales (Brokers) Manage Contracts (Life) Record Product Definition (GI) Handle Claims (GI)

Figure 5: Comparing two system footprints

Expressed in capabilities implemented. Use a common catalog of capabilities. Check the capabilities that are implemented by each solution. Use color coding as used in Heat Mapping to visualize which capabilities are implemented by a solution (and which are not).

Variants

Instead of simply using a Boolean value for *"X implements capability Y"* you can also use a scale for the degree quality to which X implements Y.

Consequences

Ease of use: Once you have a common capability model, applying it for Footprinting is really straightforward and easy. If you have a Heat Mapping tool anyway, you can also use it for Footprinting.

Fast and accurate enough: You can use Footprinting down to any level of you capability list you define. You can also later refine the evaluation if you used a coarse footprint to kick out a few of the evaluated candidates and invest in more detail only for the likely candidates of your evaluation.

Related Patterns

Footprinting can be seen as a special form of Heat Mapping, using one property (is implemented in area X (in solution X)) and hence a trivial coloring function. If you map a given application to a capability catalog you end up with its footprint – and you can easily visualize that using a "degenerate" heat map.

In order to be effective at Footprinting you need a high quality capability model of the area you want to work in. You can Mix the Models from various sources.

You can also combine Footprinting with Kiviat diagrams [EAMPat09]

Known Uses

Alfabet is able to visualize heat maps and hence also footprints [CamKal09]. Footprinting is used in strategic planning by various financial.

Pattern: Mix the Models

Example

Imagine you need a catalog of business capabilities that you need to start a capability based planning activity. You will need to come up with the around 1000 or so titles for capabilities that will allow you to start a capability based planning exercise. The capabilities should be structured in 3-5 or even more levels.

Problem

How do you get a good enough capability list and domain model for your planning efforts?

Forces

Speed versus quality: Typically you do not have "all the time" when your management wants you to start developing a strategic plan based on capabilities. Developing a catalog of maybe a thousand headlines, as needed for a typical international enterprise, and discussing this in your company with "all the stakeholders" may take far more than a year. Typically these are time spans that a management does not accept. On the other hand, if you start a brainstorming in a small group under high time pressure, the result might arrive faster – but the quality level will suffer: The catalog might be incomplete, redundant, might not capture the strategically important capabilities and the likes.

Patience of your management versus accuracy: In a management workshop you will not get the time to discuss 2000 single capabilities. You might get enough time to discuss 50-70 important top-level capabilities and their respective state and importance to the enterprise.

Full custom versus reference model: When you are in a situation to produce a capability model for your enterprise, you can try to "invent" the whole catalog, which means that you try to derive the capabilities from some theoretical or other models by applying whatever business engineering techniques. This will typically take lots of time and the results might not have the quality of best in class models after a first effort. On the positive side, such models have the potential to be relatively well tailored to your specific enterprise. You can also think of buying a capability list from a consulting firm the likes of Microsoft who have reference capability models for a large choice of industries. If you buy a list, it will be ready for your project within even a week – if you hire the consultants from such a firm.

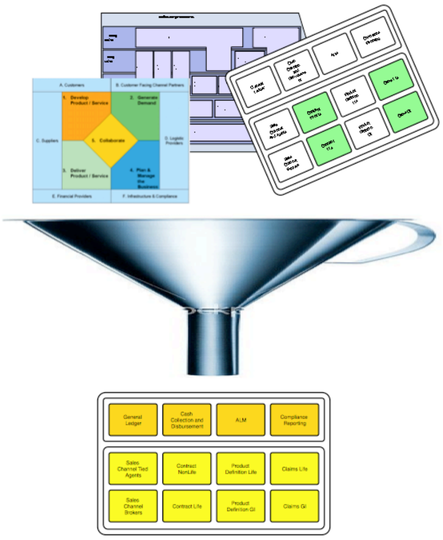
The downside of this is, that you get a list of capabilities for the average of the industry. If you want to focus on non-standard capabilities, you might miss some that are important for your business.

Clean design versus pragmatism: If you scan literature, you will find business engineering methods that would allow you to develop a domain model, capability model and also service model more or less from scratch by sitting down with a group in front of a whiteboard and applying design principles [Quasar08]. On the other hand you can also draw pragmatically the models from various sources and design your model by discussing at an expert level. Those experts will also have design principles in their mind, when synthesizing their company model – but they will not start with a blank sheet or blank whiteboard. In contrast to a method consultant who is good at thinking tools, they will also know the industry they are operating in plus also some methodology.

Good enough and fast versus perfect and slow: You could spend years researching the perfect capability catalog for your company. But at the end of the day management will not pay you for producing a 100% solution but they will pay you for fast results. Besides the perfect catalog would be outdated because business changes faster than engineers can apply real methodological engineering. The usual 80% of the maximum possible results by employing maybe 10-20% percent of the effort for a 95% perfect solution should do the job. This is well known as applying the Pareto principle.

Solution

Mix your capability models from various sources …



Typically you will start with a **domain map** that describes the top level of your company. Find your own domain map by considering the following sources:

- **Folklore from your own company:** Often companies do have domain maps from previous planning efforts.

- **Industry maps:** In many industries you will find industry reference models maintained by groups who foster knowledge exchange and promote best practices in an industry. eTOM is a typical example of such models for the telecom industry. If you know nothing about the airline industry where would you look for reference models? IATA for example! If you need reference material on insurances you might want to ask your countries association of insurers.

- **Standard software solution maps:** Often there is also standard software available for your industry. A very prominent example is SAP. SAP has got so called solution maps for a lot of industries, which will also show typical domains and how SAP software is ideally applied to them. These SAP solution maps are often a very good source if you try to structure an industry or enterprise.

- **Sector consultants' maps:** Consulting companies typically organize themselves according to industry sectors, the likes of banking, telecom, chemical process industry, or e.g. manufacturing industry. Typically they also have capability lists from previous engagements with previous customers. Often they also invest some effort in consolidating such lists or models in order to have a reusable asset for future assignments. By hiring such consultants you can also get a grip on their repositories.

- **Generic maps:** On a high level – to start with – you can use models like Porter's value chain or the below top-level domain map by Microsoft (as presented above in Figure 3).

Once you have your domain maps you will continue to break down the capabilities to a level 2-3. Typically the internal sources start to dry up here and you will need to mix stuff from industry maps the likes of eTOM and also sector consultants' stuff with generic capability lists of some consulting firms.

Variants

There are too many variants to be counted. Any combination of sources for your capability model could be seen as a variant of the pattern, depending on what sources are accessible to you in your industry.

Consequences

If you mix **your capability model from various sources** this has the following typical consequences:

Speedy and good enough: Typically you or your management will set you a time box. Mixing your capability model enables you to cope with various lengths of time boxes. If your management is in a hurry they can rely more heavily on external material.

If they want to save money and have more time, you can buy less consulting and rely more on slower sources like employing internal experts.

Sufficient individuality: Even models from sector consultants are good enough in many cases. If there are some special capabilities needed, they can often be found in generic capability lists or also in capability lists for slightly different industries. If you have strategically important additional capabilities you might know them anyway (if they are considered important in your company). Hence a mixed capability model will be "individual enough" in most cases.

Pragmatic Design: Design always deals with tradeoffs between requirements that are often also tough to quantify. This is why you will often get results at least as good as the guys with the blank whiteboard by getting a group of domain experts together why exchange domain patterns – but do not challenge them from scratch. You can use the domain engineering rules to challenge your models and explain them. But you should not start from scratch with a blank whiteboard. If you mix models this will not happen to you.

Related Patterns

A good catalog of capabilities is the common base for the other patterns described in this paper such as Heat Mapping and Footprinting.

Known Uses

There is one international telecom provider based in Germany, who has mixed their capability models from sources such as their own group domain model, eTOM and also capabilities lists from a large software firm. A second large scale Telco operating in Germany also makes extensive use of eTOM plus own considerations. There are more than a handful of insurance companies who have developed their domain architectures based on VAA, an initiative of the German Association of insurers GDV.

Concluding Remarks

If you have suggestions how to improve this paper, please feel free to address the author. Good questions will help improve the next version of this paper. Remarks concerning language are also welcome as the author is not a "native speaker". Any help welcome.

References

[Brede+06] Bredemeyer, D.; Krishnan, R.; Lafrenz, A.; Malan, R.: Enterprise Architecture as Business Capabilities Architecture; Whitepaper; Bredemeyer Consulting, available via http://bredemeyer.com/pdf_files/Presentations/EnterpriseArchitectureAsCapabilitiesArch.pdf (link verified 2009-11-12)

[CamKal09] Cameron, B.; Kalex, U.: Webinar (Web Seminar) on Business Capability Management; Forrester Research & alfabet AG, June 2009. Available via http://www.alfabet.de/news/veranstaltungen/webinar_driving_productive_it_investment (link verified 2009-11-13).

[EAMPat09] Buckl, S.; Ernst, A.; Lankes, J.; Matthes, F.: Enterprise Architecture Management Pattern Catalog: Single Pattern: Kiviat diagram showing conformance level. Available via http://eampc-wiki.systemcartography.info/wikis/eam-pattern-catalog/v-83 (link verified 2009-11-15).

[Homan08] Homan, L.: Enterprise Architecture; Connecting Business and IT, Talk at GI-Jahrestagung, Munich, September 9th, 2008

[HeatMap06] Microsoft Services: Microsoft Motion Heat Mapping Tool: *blogs.microsoft.co.il/files/folders/2034/download.aspx (link verified 2009-11-14)*

[MerTob06] Merrifield, R.; Tobey, J.: Motion Lite: A Rapid Application of the Business Architecture Techniques Used by Microsoft Motion; available via http://msdn2.microsoft.com/en-us/library/bb736727.aspx (Link verified 2009-11-14).

[Kumar06] Dinesh Kumar; Motion – A Framework for SOA Adoption in the Enterporise; Microsoft TechEd Conference, Pasadena 2006.

[Quasar08] Engels, G. et al.: Quasar Enterprise; dpunkt Verlag, Heidelberg 2008.

[Ritz08] Ritzenhöfer, G.: SOA-basierte IT-Strategien für Banken, Talk at GI-Jahrestagung, Munich, September 9th, 2008.

[TUM09] Buckl, S.; Ernst, A.; Kopper, H.; Marliani, R.; Matthes, F.; Petschownik, P.; Schweda, C. M.: *EAM Pattern for Consolidations after Mergers*. In: Workshop on Patterns in Enterprise Architecture Management (PEAM 2009), Kaiserslautern, 2009.