Chapter 4
More About EAP

«Finally, in conclusion, let me say just this.»
Peter Sellers

Abstract The first part deals with how EAPs can help in structuring the Enterprise Architecture. The second section deals with the application of the patterns and how they evolve over time. In the last two sections, we take a look into the future and motivate the use of EAPs in upcoming trends in ICT.

4.1 More Reasons for Using EAPs

In Chap. 2, we described how EAPs might help in structuring the overall Enterprise Architecture and consolidate existing environments. In the following sections, we give some thoughts about further reasons for using EAPs.

4.1.1 Identifying Fields of Action

We expect that the one who reads the most about EA patterns is the Enterprise Architect. He who defines, plans, aligns, and pushes forward the architecture of his enterprise may use the pattern catalog as an inspiration as well as a comparison to find «his» patterns in his enterprise. There may already be many patterns in place, but also multiple instances of a pattern might be a helpful indicator that work needs to be done. How does your Enterprise Architecture compare to patterns? This may be used as a guidance on where action is needed; for instance, if you observe reoccurring installations all resembling a certain pattern. A typical case may be found with the occurrences of different access management installations.

T. Perroud and R. Inversini, *Enterprise Architecture Patterns*,
DOI: 10.1007/978-3-642-37561-3_4, © Springer-Verlag Berlin Heidelberg 2013

4.1.2 Facilitating Discussions

The noblest purpose of a pattern is to facilitate the discussion between the various stakeholders. The Enterprise Architect finds himself often in the role of a mediator between various requirements and expectations. In order to guide the discussion toward meeting the various requirements, he may use the pattern catalog first to find suitable patterns. Then he discusses the chosen pattern with the various stakeholders. To do this, he focuses on the architectural layer where the discussion partner is most likely to be at ease. He uses the business architecture layer to talk to the stakeholders representing the business and the technical architecture layer for the network or system engineer who has to build the system. The layer in between, the information system layer, should be the glue between these two.

Whenever a problem needs to be solved that can be described with one of the following questions, the use of patterns may be helpful:

- We want to implement a new ICT strategy, how do we start? Patterns may help by getting a quick overview of the action needed and by providing glue between the business processes and the ICT infrastructure.
- The various stakeholders, especially the business oriented and the infrastructure oriented, do not talk the same language. If you hear repeatedly the sentence «this solution does not meet my requirements» and the answer of ICT is something like «you did not specify this requirement», patterns are a wonderful means to provide a common understanding of needs, requirements, and infrastructure.
- A long-term program to implement a business process is established. Patterns help in every phase of the program to gain oversight and direction and to stay focused.

4.1.3 Working in Projects

While patterns are helpful at a strategic or governance level, the use of Enterprise Architecture Patterns also alleviates the dire everyday work, consulting and guiding projects and trying to fulfill their needs in an ordered and foresighted way without negatively impacting the blueprint of the Enterprise Architecture. As with every project, you should be clear about the requirements of the various stakeholders. Therefore, it is necessary to conduct a stakeholder analysis right at the start. The patterns may serve as a catalog where the Enterprise Architect can show various examples to the stakeholder asking him «did you mean something like this?» When starting the process of clarifying the requirements and proposing solutions, the catalog is very helpful to locate interesting patterns. By looking at the interfaces provided at the three architectural levels, the architect is able to identify and propose promising patterns. As the project progresses, he may use the patterns as a guideline to check whether the project stays focused and is likely to accomplish its goals.

4.2 It Depends…

As every company is unique, with its own identity, communication culture, governance, and business, there is no magic formula we can give you on how to implement the patterns. It depends on many factors that may even be unique for your enterprise. But reading and trying to understand the purpose and goals of the pattern may already be the first steps to leverage your Enterprise Architecture – independently of the current maturity. Your own debate with the patterns and the mental mapping with your environment may already help you to structure things in a better way.

We introduced a language to describe the patterns that we think may be helpful for the common understanding. This does not mean that the terms we use may be the right one for your specific environment. You may want to map our terms to your glossary and definitions. This will first help you to better understand the patterns (and to question them) and second enable the discussion with your colleagues.

Even if the patterns are not implemented, they may nevertheless help you in gaining more insight into a certain topic. The patterns provide a «tour d'horizon» on a certain topic. This may help you to first gain an oversight and then find the starting end of the golden thread on how to get into the topic. Take for example ResourcesAreScarce (S3): The Business View and the Data and Application View gives you a list of possible tasks and related applications and services that may be of interest in human resource management. Depending on your environment, you may already have some elements in place. Perhaps you did not have a structured overview, as described in the pattern, or some of the elements may be missing. In this case, the pattern can serve as a reference, an information source, or a possible to-be architecture, even if you do not implement it.

4.3 Patterns Within Patterns

Several patterns within Enterprise Architecture Patterns seem to exist. We are not going to discuss these in deeper detail but would like to point them out nevertheless as they give a strong indication of the importance of some of the Enterprise Architecture patterns or show commonly known and applied paradigms of ICT.

A high-level pattern that characterizes a type of business is the «Acquire-Transform-Publish» sequence. This pattern is for example shown in Fig. 8.8 of the InformationChest (S2) pattern. This is a typical sequence found in many business processes. Examples are governmental agencies gathering statistical data, companies performing customer surveys, or a search engine provider.

At the business level, a frequently occurring schema is the process of login/logout that surrounds the actual business action. This typical structure that can be observed as soon as privileges for an operation are slightly elevated, is shown in

Fig. 4.1. This fact emphasizes the importance of the StoreMyIdentities (I1) and LetMeAccess (I2) patterns.

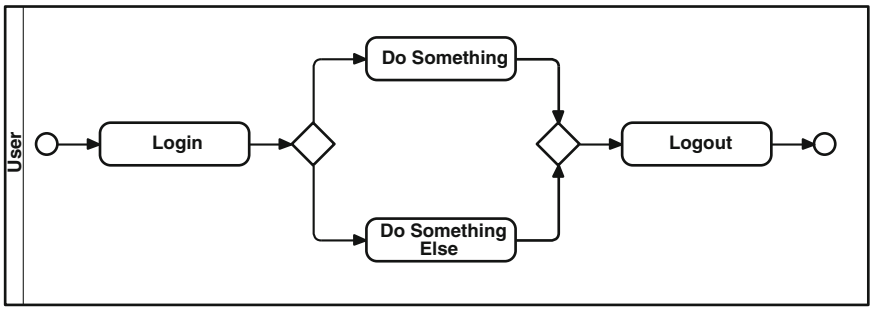

Fig. 4.1 Login/logout structure

Another pattern within patterns at the business level is the search and selection of information that afterwards is modified, enriched, or simply worked upon, see Fig. 4.2. As this pattern is used by all other patterns dealing with information (and we are talking about Information Technology, so it is ubiquitous), it is defined as a business process in the InformationChest (S1) pattern and implicitly used in all other patterns or applications.

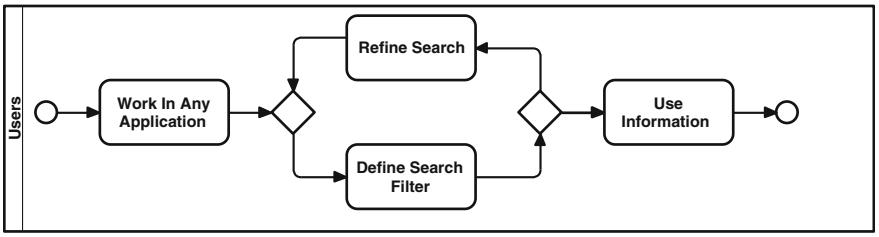

Fig. 4.2 Search pattern

One commonly encountered archetype within patterns is the distinction between frontend and backend at the application level. Often, the frontend is facing the customer whereas the backend is facing internal applications and business logic as well as the further processing of the request that has been placed by the customer. This characteristic has its origins at the business level and can also impact the technical level. It is common that both tiers access the same datasets sometimes with equal rights, but very often users in the backend have more rights as they process the data entered in the frontend. The separation between frontend and backend not only has advantages by simplifying deployment but also in terms of security (Fig. 4.3).

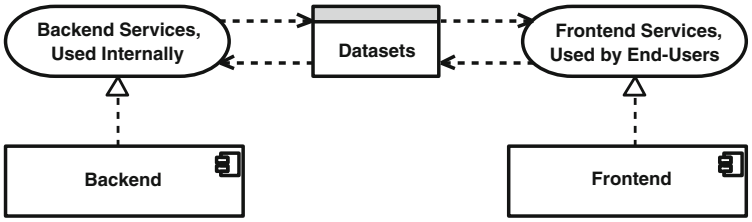

Fig. 4.3 Frontend/backend pattern

At the technical level, the combination of a web server that stores the data in a database, which in turn has access to a storage system, can be seen in nearly every pattern. It is most common but has many implications on various levels. A distinction between these elements decouples the elements and improves manageability, security, and robustness. Changes at any element can be done more easily than in a tightly coupled environment (Fig. 4.4).

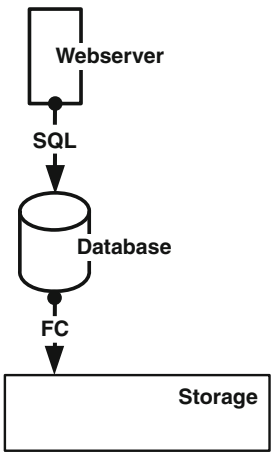

Fig. 4.4 Web server/database pattern

Through all levels of most patterns an administration schema can be seen. As most applications need to be managed, a management or administration interface with the necessary technology must be defined. At a relatively high abstraction level, these have many similarities and, in fact, they can be implemented in a uniform way by defining specially protected administration systems that are accessible for administrators only and that are restricted to the management of systems. This pattern exists on all levels, from the business layer view to the application view down to the technology view. Basically an actor, e.g., a system administrator or an application administrator has to fulfill administration tasks such as modifying configuration parameters. This is done using a management application that can be standalone or embedded into the functionality of the

normal application. On the infrastructure level, a gateway system is often used that regulates access for performing any management tasks and management functions. It protects the application or the system from unauthorized modifications. Figure 4.5 shows such an administration pattern on the application view.

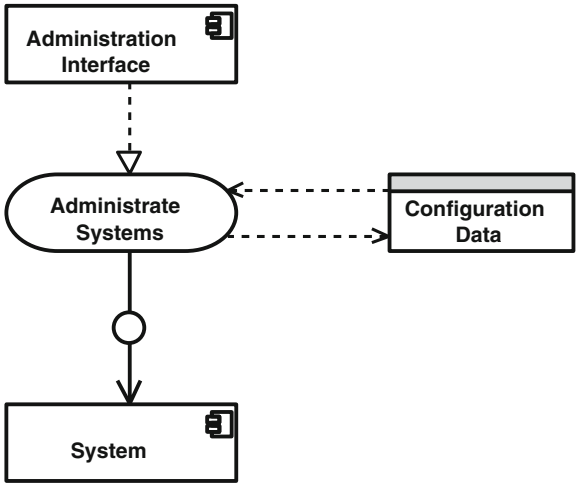

Fig. 4.5 Administration pattern

4.4 (De-)composition

Understanding the nature of Enterprise Architecture is an ongoing process of decomposition and composition of the various parts. At first glance, it is a nearly impenetrable jungle of business processes depending upon applications and infrastructure elements. After having observed the architecture, different recurring patterns are emerging. First, they may be diffused and unclear and this is where the pattern catalog may help out. It helps to clarify and identify different key elements in the infrastructure of the Enterprise Architecture, thus enabling you to decompose the jungle into understandable and manageable bits (and bytes). By having identified the patterns in your own Enterprise Architecture and comparing them to the proposed pattern, you should now be able to compose the new infrastructure by adopting the patterns and changing the Enterprise Architecture into something that is well understood and maintainable and that serves the business best.

4.5 The Life Cycle of a Pattern

EAPs and the real-world architecture evolve and change over time. This may be, for example due to technological advancement (who does nowadays acquire a fax to use them in a business process?) or changes in business processes

(«it was common to do it that way, the best practice are nowadays to do it this way»). The patterns are the blueprints to transform the Enterprise Architecture but are only useful when taking the real-world situation into account. New facts in the real world influence the patterns and the patterns will lead to a re-Architecting of the real world. The first application of the patterns will probably have the largest impact on re-Architecting efforts, but the more the Enterprise Architecture matures, the less re-Architecting efforts will be necessary.

The life cycle of a pattern begins with the identification in the «as-is» architecture. Identifying means to look out for similarities. A prerequisite to be able to identify patterns is certainly that one must have a description of the «as-is» architecture.

The similarities could be found on every architecture level – be it the business, data and application, or technology level. There are patterns that are obvious, take for example the StoreMyIdentities (I1) or LetMeAccess (I2) patterns. Others may occur only in a certain industry, like a stock exchange application.

The next step consists in documenting the pattern. You may do that according to our proposal in this book or develop an own documentation scheme that suits you best. The important thing in documenting the patterns is to know who will read it and what objectives you want to reach. As described in Sect. 2.8, we recommend beginning with a simple version of the pattern and then refining it over time.

The pattern will help you to define the scope of a project to realize it. It represents, therefore, a part of the to-be architecture and will serve you as a guideline for the implementation. During the implementation you will probably gain new insights. This can lead to modifications of the pattern – and you may make it available to the Enterprise Architecture community.

4.6 Where to Begin: A Question of Maturity

We believe that two ways of applying these patterns exist:

- You have a project to implement and want to help the project leader or solution architect by giving some guidance from the point of view of Enterprise Architecture. We will describe this approach in Chap. 5: The patterns are being applied to solve the problems arising in a project.
- You want to structure your overall Enterprise Architecture, consolidating existing environments, reduce costs, streamline your different architectures, etc. In short: do your job as an Enterprise Architect. This is what we describe in the following:

So, where to begin? For every pattern the three parameters, *Complexity*, *Connectivity,* and *Invariance* are defined according to our opinion. Each parameter can be set with one out of three possible values: Low, Medium, and High. The exact meaning of these parameters and values are defined in Sect. 2.11. This leads to the following Table 4.1:

Table 4.1 Invariance, complexity, and connectivity for every pattern

	Complexity	Connectivity	Invariance
FromSupplierToCustomer (S4)	High	High	Low
StoreMyIdentities (I1)	High	High	Medium
LetMeAccess (I2)	High	High	Medium
InfoChest (S2)	High	High	Medium
WorkTogether (B1)	High	Medium	Low
UnderControl (I3)	Medium	High	Low
Financials (S1)	Medium	High	Medium
KnowYourCustomer (B3)	Medium	Medium	High
ResourcesAreScarce (S3)	Medium	Medium	Medium
TalkToMe (B5)	Low	Medium	Medium
ForYourEyesOnly (S5)	Low	Medium	High
YouHaveMail (B4)	Low	Medium	High
VendingMachine (B2)	Low	Low	High

In Table 4.1, the patterns are sorted according to their complexity and connectivity and not according to the type and number of the pattern. This is due to the fact that there is no correlation or causality between the type of the pattern and the three parameters taken together.

The *Invariance* parameter is the measure for our confidence in having found a generic pattern. The higher the value, the more confident we are. When looking at the last row we can see a tendency: The higher the complexity and connectivity, the less generic the pattern is to our understanding. This is quite obvious as an increasing connectivity and a high complexity makes different implementations of a pattern much more likely. Another tendency that can be deduced is that the simpler the business layer, the lower the complexity.

Using patterns to consolidate your Enterprise Architecture means that you need an overall view of the topic of patterns you want to implement: As the green-field approach in most cases is not an option, you first have to gain an understanding of what is already in place. Take for example the StoreMyIdentities (I2) pattern: Chances are high that there are already many different instantiations in your architecture. To be able to document the «as-is» architecture of your identity management environment, you will need to get the necessary resources and management commitment. If the Enterprise Architecture function is not yet well established, you will encounter many obstacles that are not directly related to the implementation of the pattern. The justification of the existence and goals of the Enterprise Architecture will be questioned and you will have to do a lot of lobbying for your cause. The less mature the Enterprise Architecture, the more difficult it is to handle architecture topics that affect the overall architecture. In relation to Table 4.1 this means that it is certainly easier to start with the patterns at the bottom of the list, if the Enterprise Architecture function is still in its infancy. If the maturity of the Enterprise Architecture is low, you should do the following: Begin with patterns where you do not need a high commitment from

upper management. This can be done for patterns with a low to medium complexity and connectivity. VendingMachine (B2) and ForYourEyesOnly (S5) are in our opinion the most easy patterns to implement and do not represent a large risk. If you do not need these patterns, take the first one that suits you, beginning at the bottom of the list. Doing so allows you not only to implement patterns successfully, but also to gain broad acceptance for the Enterprise Architecture. This finding correlates also to our rating of the invariance of a pattern. In other words, when choosing first a pattern with a high invariance and therefore a low complexity and connectivity you are on the safe side. However, if your Enterprise Architecture has already gained some level of maturity and you have at least partial support by top management, the gain of implementing a pattern with high complexity and connectivity such as StoreMyIdentities (I1) or LetMeAccess (I2) may be of greatest benefit for the enterprise as many applications profit and as many other patterns can unleash their full potential. If you want to measure the maturity of your Enterprise Architecture, take for example (NASCIO 2007) or (ACMM 2007) as a guideline.

There is certainly no magic formula that will tell you which pattern you should implement first. Every environment is different and has its own specificities to consider. Assessing the maturity of the Enterprise Architecture is one way of getting into the topic. Another possibility is shown in Chap. 5.

4.7 Less Is More

While writing this book and carrying on our jobs, we observed an interesting phenomenon that may give a hint on how the «as-is» landscape of an Enterprise Architecture evolves. We believe that the evolutionary stage gives an indication of the maturity level an enterprise has concerning Enterprise Architecture in general and of the use of EA patterns.

What can often be observed is that for many of the patterns, especially Infrastructure and Support Patterns, a variety of unconnected instances may exist. In the first stage; the instances multiply as every project is confronted with such requirements and is forced to find a solution on its own. It may seem to be an unsolvable task to introduce some organization into this jungle. In the next stage, the instances within one pattern begin to harmonize and one after the other is given up or integrated in the instance with the highest potential. This process can be unguided, driven by the pressure of cost saving and complexity or it can be guided by a strategy, which of course is the preferable way. After some time, a form that is often very likely to these presented in this book has evolved. Simultaneously with the reduction of instances within one pattern, links between different instances of the same pattern begin to form and are getting more and more important. Figure 4.6 shows such an evolution. At the initial stage, two different Business Patterns have their own instance of the same support and the same Infrastructure Patterns. For example, the VendingMachine (B1) pattern (or an

application resembling this pattern) and the KnowYourCustomer (B3) pattern both use their own instances (named *a*, *b*, *c*) of the InformationChest (S2) and the StoreMyIdentities (I1) pattern. In fact, there exist three StoreMyIdentities (I1) instances as the identities are fragmented into internal and external users. During the transitional phase, a consolidation of the various instances can be observed. While the instances of the Support Pattern are migrated into a new application, the identity management is concentrated into one of the existing instances (I1c). When a level of maturity is reached, the connections are well-defined and simple. The new Support Pattern has evolved and is being used by both Business Patterns; the same applies for the Infrastructure Pattern, from which only one instance exists that is being used by all patterns.

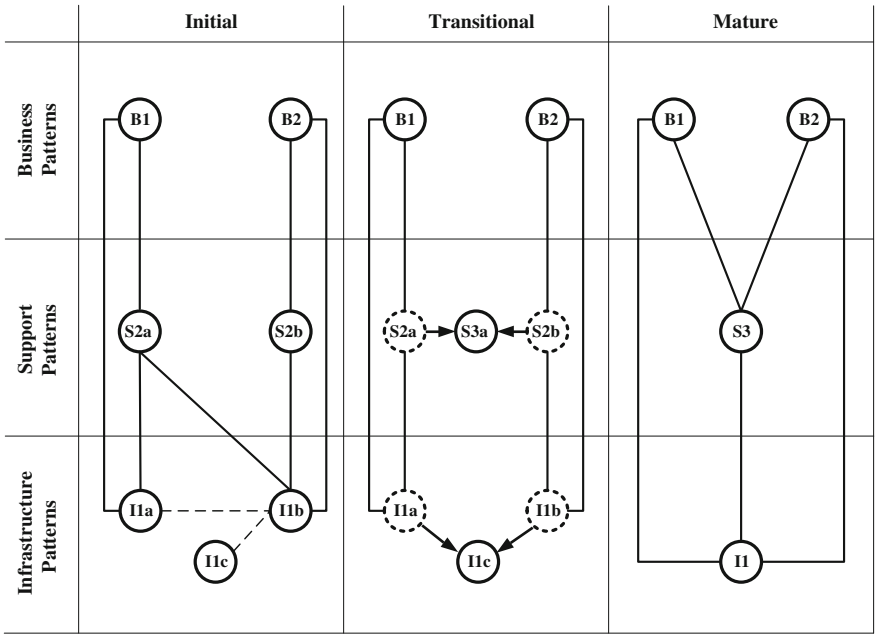

Fig. 4.6 Evolution of the number of instances

4.8 EAPs and Actual Trends in ICT

During the writing of the book, we were asked several times, whether the patterns would become obsolete with the upcoming of *cloud computing*. We are convinced that even in a cloud computing environment, be it software-as-a-service (SaaS), platform-as-a-service (PaaS), infrastructure-as-a-service (IaaS), or even business-as-a-service (BaaS), the patterns can be applied. Identity and access management, collaboration, or customer management outcomes must be solved even in a cloud

computing environment. Moreover, we think that new patterns may arise. On-demand capacity management and pay-per-use services are two examples that could be investigated further. The just-in-time provisioning of computing resources can even be leveraged by the use of EAPs: The patterns will not only help in provisioning single computing resources, but entire solutions with networks of resources according to a pattern.

The announced demise of Services-Oriented Architecture (SOA) in 2009 by Anne Thomas Manes (Manes 2009) and her call to concentrate on *services* is still valid. Service orientation is still a topic, but perhaps with less euphoria and more (economic) realism. The EAPs support the paradigm of service orientation: Architecture bricks are decoupled through the use of interfaces, and applications communicate using services. The services are coarse grained and have a high degree of reusability.

Mobility is one of the major trends in ICT: The access to business information on the road, concepts like Bring Your Own Device (BYOD) or the convergence of data and telephony on the smartphone are now radically changing our ways of communicating together. People are constantly online, checking their e-mails, organizing meetings, or approving application when outside of the company. We tried to take this trend into account when describing EAPs: You will find for example references to mobility in all Business Patterns.

The upcoming of *social media* within companies will open new communication and information paths. The first companies have already begun – for example, by formulating policies that will ban e-mails and substitute it with social media technologies (wikis, blogs, electronic pin boards, etc.). WorkTogether (B1), KnowYourCustomer (B3), or ResourcesAreScarce (S3) are all EAPs that refer to the communication between persons, be they employees, partners, or customers. You will find in all these patterns references to the use of social medias as possible means of communication.

The promises of the hype called *big data* to unveil new insights in the ever-growing amount of data will be put to test in the next few years. Besides the mastering of the necessary technologies to cope with real-time analysis of huge amounts of data, we will probably also need new skills of experts who are able to investigate these data. This may lead to new processes and ways of dealing with information and subsequently to new Enterprise Architecture Patterns.

4.9 The Future

We are convinced that Enterprise Architecture is one of the most important fields for most enterprises as otherwise they risk losing control over cost and orientation in the field of Information Technology and data architecture. In order to reduce the complexity and to have a common understanding between the various stakeholders, the concept of patterns may be of high value.

If you have come this far in the book, we would like to encourage you to follow the path of working with patterns and to share new or derived patterns with the community.

References

ACMM (2007) United States Department of Commerce: enterprise architecture capability maturity model. http://ocio.os.doc.gov/ITPolicyandPrograms/Enterprise_Architecture/PROD01_004935. Accessed 28 Feb 2012

NASCIO (2007) National Association of the State Chief Information Officers. http://www.nascio.org/publications/documents/nascio-eamm.pdf. Accessed 21 June 2013

Manes A (2009) SOA is dead; long live services. Burton Group blogs. http://apsblog.burtongroup.com/2009/01/soa-is-dead-long-live-services.html. Accessed 4 Mar 2012

http://www.springer.com/978-3-642-37560-6

Enterprise Architecture Patterns
Practical Solutions for Recurring IT-Architecture Problems
Perroud, T.; Inversini, R.
2013, XI, 320 p., Hardcover
ISBN: 978-3-642-37560-6