

Service Oriented Enterprise Architecture Modelling

Growth of Enterprise Architecture Modelling

Whether economic conditions are challenging or favourable, organisations should always seek to maximise the value from their IT investments, identify and remove waste and ensure that future investment decisions are based on what the business needs in the long term (strategic) as well as the short term (tactical).

Developing and maintaining a model of the Enterprise Architecture (EA) has become an established approach to achieving these objectives. This paper introduces a new way of modelling the enterprise, based on a service-oriented approach. It is simple yet powerful in nature, can be adopted incrementally, and provides tangible benefits to organisations seeking to understand the current state of their business and IT or implement strategy through projects and change programmes.

Why Service Oriented?

The use and application of EA modelling can vary widely, depending on the specific needs of an organisation. Typical applications include:

- Strategy development
- Current state modelling (architectural review)
- Technology asset management
- Project delivery

In many cases, architects have used drawing tools such as PowerPoint or Visio to capture architecture models. However, models built using these tools often prove difficult to maintain as the architecture evolves. More fundamentally, each architect will often have their own terminology and way of representing complex relationships between elements of the architecture. This lack of consistency in representation and terminology compromises the value of the model, as translation of the model by the author is often required.

There are tools and methodologies to address these representation issues. However, many of these tend to focus on a sub-set of the EA modelling requirements; some being too high-level and some being too low-level. This fragmentation in the tool marketplace reflects the variety of definitions across the IT industry of what enterprise architecture is. This is understandable, given the many ways in which an organisation can exploit enterprise architecture modelling.

Indeed, the experiences of most end user companies and across the IT industry had shown that there is still a considerable challenge in this space - to devise a flexible approach to enterprise architecture modelling that caters for all of the different perspectives.

Some of the key benefits of Service-Oriented Architecture (SOA) in software architecture relate to the ability to clearly structure and maintain the definitions and relationships of components that comprise a system, thereby providing a foundation for managing complexity and change. These benefits are realised through adherence to a consistent set of principles applied at both low levels and high levels of abstraction.

If an enterprise can be considered to be a system of inter-related components varying in nature and granularity we believe that, by applying service-oriented principles to enterprise architecture modelling, the same benefits related to managing change and complexity can also be gained.

Taking a Service-Oriented Approach

Whether service-oriented or not, it is widely agreed that an enterprise architecture framework should allow an organisation to describe their enterprise by capturing and relating the business, application (or system), information and technology elements that exist within it. It is also common practice to categorise these elements into architecture layers with clearly defined dependencies (see Figure 1).

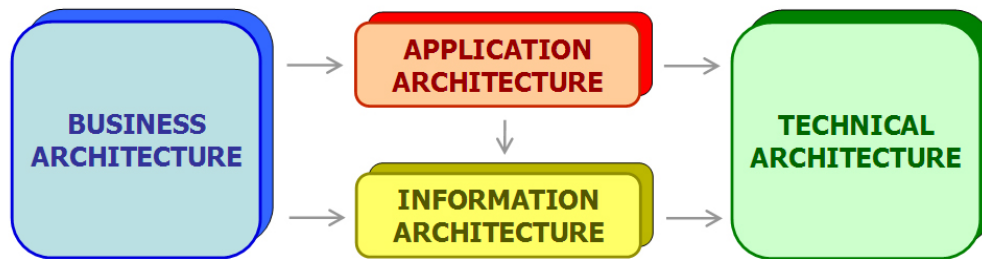


Figure 1. Enterprise Architecture Layers

For any enterprise architecture framework, it is vital to understand how the components contained within the different architectural layers relate to one another to gain a complete enterprise view of an organisation. These relationships have been formalised in terms of a service-oriented meta-model and a well-defined set of modelling rules.

Within each architectural layer, the service-oriented meta-model comprises a number of architecture constructs that exist at different levels of abstraction. These constructs provide architects with the different *Views* of the enterprise that they require, depending on the task at hand.

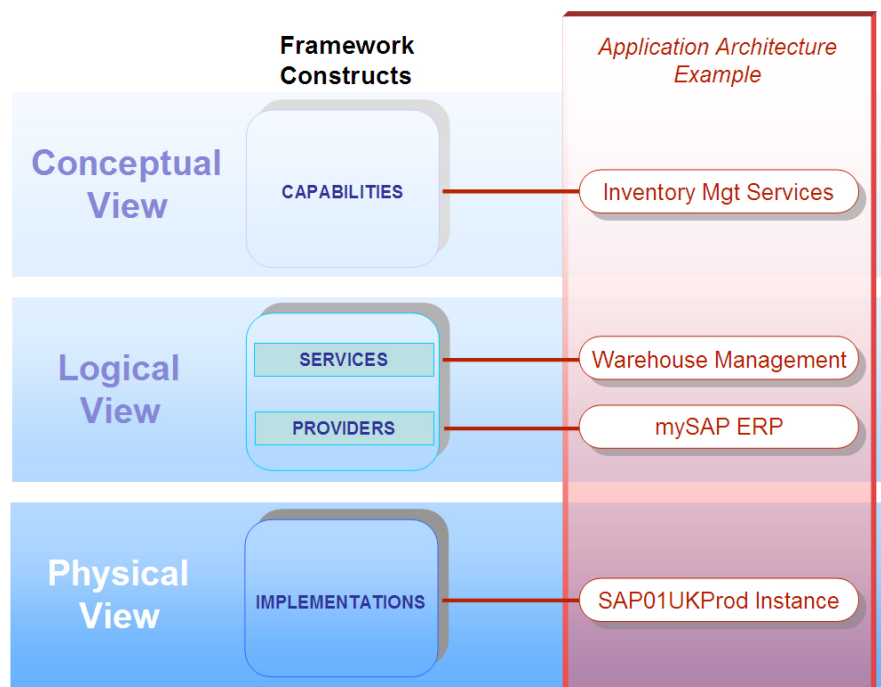


Figure 2. High Level Meta-Model and Example

The *Conceptual View* provides the highest level of abstraction and defines the *Capabilities* provided by a particular layer. These Capabilities describe the ‘what’ for a given layer.

The *Logical View* defines the set of *Services* that realise the Capabilities of the Conceptual View. In addition, components that implement these services are defined as *Providers*. Together the Services and Providers describe the ‘how’ for a given layer.

The *Physical View* defines the *Implementations* of the Providers from the Logical View that physically exist within an enterprise. This view describes the ‘who’ and ‘where’ for a given layer.

These views and their associated modelling rules are applied consistently across the architecture layers to form an Enterprise Architecture framework. This framework has been designed to be implemented in any modelling tool with a flexible meta-model. Indeed, it has been implemented in a number of popular CASE tools and architecture modelling tools to date.

What are the Benefits?

The framework has been applied in a number of end-user organisations from several market sectors and with varying levels of experience in enterprise architecture. To ease its adoption, many have superimposed their own IT terms and architecture languages onto the framework - e.g. Application Layer \equiv Systems Layer. Nonetheless, in all cases the fundamental concepts and principles of the architecture framework have been preserved. After only a short period of support and familiarisation, the in-house architecture teams of these organisations have been able to apply the framework to meet a variety of different modelling objectives, including:

- Business strategy development, identifying opportunities and risks
- IT strategy development
- Programme management for large scale business change
- Impact analysis for decommissioning, mergers and acquisitions
- Gap analysis when developing new business propositions
- Identification of redundancy for rationalisation and cost reduction exercises

In addition, the service-based nature of the framework also delivers many of the classic benefits associated with a service-oriented approach to software architecture. In the context of enterprise architecture modelling this means:

- A single modelling paradigm with a simple, minimised set of modelling constructs, applied consistently, whatever the modelling objectives.
- Well-defined, joined up layers with formal definitions and added heuristics for deciding how to categorise a specific architecture component into a particular layer.
- A clean division of organisational responsibilities within and between layers. For example, concisely and accurately defining the service contract between an end-user organisation and its out-sourcing partner (see Figure 3).
- A repeatable approach ensuring quality and enabling architects to focus on architecture design choices rather than representation during modelling.
- Relationships defined in terms of service interfaces provide loose-coupling between layers. This in turn allows decomposition of a problem space - supporting team working and an incremental approach. For example, with a project that provides a gap analysis of business strategy to existing business systems, an organisation can focus on identifying the Business and Application Capabilities that are present for a particular business area. Although some may consider this to be an incomplete picture, the information that this model conveys, will provide value to the project at an early stage.

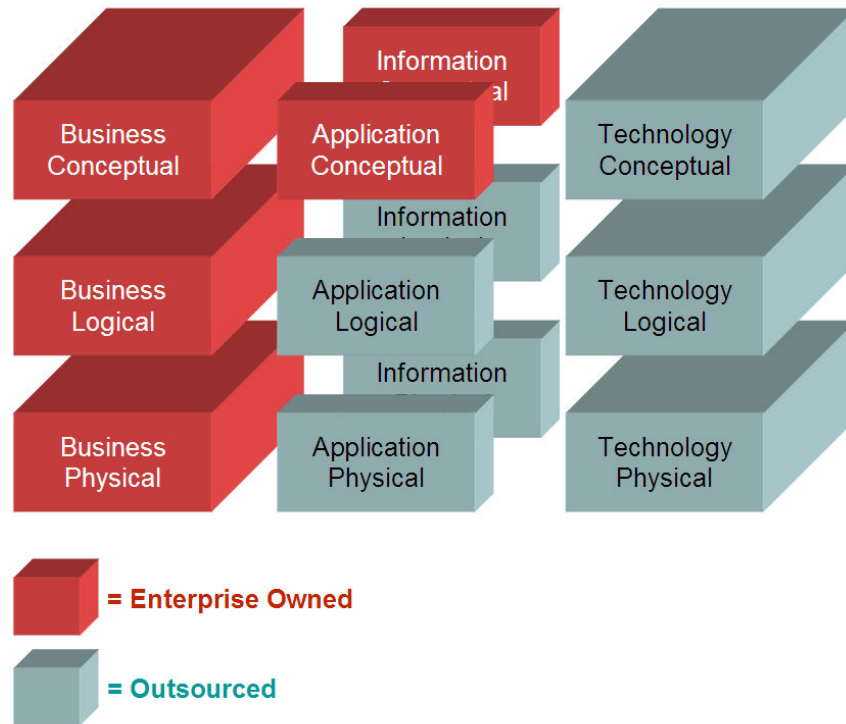


Figure 3. Defining Organisational Responsibilities

Delivering Enterprise Architecture

Enterprise Architecture is an invaluable tool for any organisation looking to improve their ability to manage and implement changes to their business and IT systems. However in practice, the complexity and scale associated with many of today’s enterprise architecture frameworks have made them difficult to introduce into existing project delivery and governance processes. Moreover, in many cases, few obvious or tangible benefits are realised in the early stages of adoption when quick wins are vital to sustaining executive sponsorship.

By applying the tenets of SOA to enterprise architecture, organisations are able to gain a better understanding of the elements that make up their enterprise - business, application, information and technology assets. Inherent complexities can be managed by hiding detail and deferring further exploration for a time when it is needed. This, in turn, allows organisations to take a more controlled and incremental approach to enterprise architecture development, giving focus to areas of higher priority or value as required.

The framework described in this paper is a realisation of a service-oriented approach to enterprise architecture. With a comprehensive, yet intrinsically simple meta-model, it provides organisations with a clear, repeatable and pragmatic method of applying the disciplines of enterprise architecture to address the issues that are most important to them at any given point in time.

*For more information please contact: info@e-asolutions.com
<http://www.enterprise-architecture.com>*