

Enterprise Reference Architecture

Addressing key challenges facing EA and enterprise-wide adoption of SOA

Ahmed Fattah

This paper relates my experience in developing and applying an approach that successfully used Reference Architecture to bridge the gap or disconnect between Enterprise Architecture and Project-level Architecture. Although there is a huge amount of information about Reference Architecture, there is considerable confusion about what it is and how it can be used effectively. I analysed and positioned various types of RAs within a simple classification scheme that cleared up some of this confusion. The scheme delineates a category of Reference Architecture which I refer to as Enterprise Reference Architecture (ERA). The paper discusses how ERA can be used to address some of the challenges facing EA including the enterprise-wide adoption of SOA.

Reference Architecture (RA) classification scheme

Although I have used Reference Architectures for a long time, I was surprised recently when reviewing the staggering range of usage of the term. Because I knew what I meant, I assumed that some of these usages must be erroneous. I then realised that this was not the case, at least for the many instances I have surveyed: the culprit is the malleability of the term 'architecture' itself. Anything you can think of can have an architecture and so by extension a RA. I tried to find a way to classify this huge number of 'things' called Reference Architecture and I found useful two dimensions: breadth of Coverage and level of Abstraction.

Coverage

Coverage or applicability indicates the area where a RA can be useful. Some RAs cover only presentation, integration or security aspects of solutions, whereas others cover an end-to-end enterprise solution. What would lie at the extremes of this dimension? At the narrow one, an Architecture Pattern which is consistent with the view that a RA is an aggregation of Architecture Patterns. At the other, we should expect end-to-end architecture that covers both business and IT aspects.

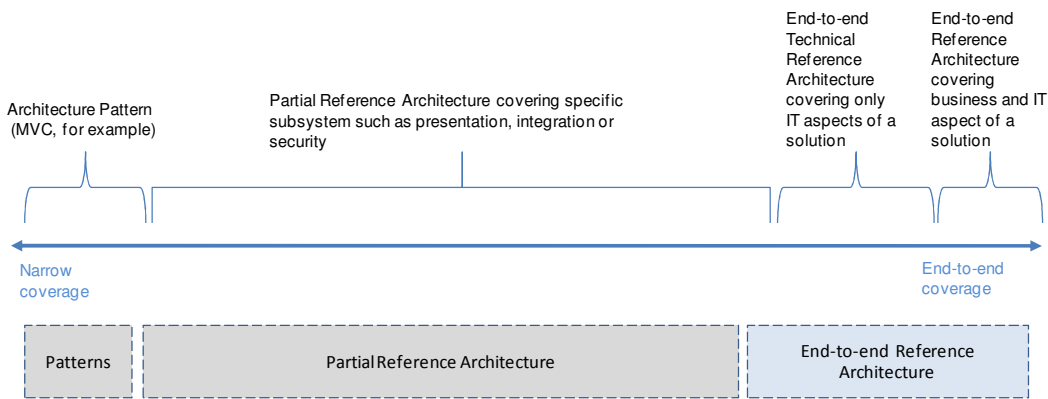


Figure 1: Reference Architecture coverage ranges from very narrow (Architecture Pattern) to full end-to-end that includes both business and IT aspects of a solution

Abstraction

Level of Abstraction reflects how concrete or specific a given RA is. It indicates the gap between the RA and a *Solution Architecture* based on it. It is a qualitative measure but very valuable in assessing how pertinent a given RA is for developing an implementable solution. I believe that one source of confusion about RAs is that some IT practitioners view all RAs as the same. Because of this, they may form an incorrect impression about the value of RAs in general based on an encounter with a particular RA.

As with Coverage, it is useful to examine the extremes of this dimension. It is easy to identify the concrete end because a fully implemented solution is as solid as you can get. It is not as easy to define the other because it seems there is no limit of how abstract, conceptual or generic a RA can be before it is not called a RA¹. This is unfortunate because it would be handy if there were a cut-off point where we stop calling an artefact a RA, similar to the Coverage dimension where there is a demarcation between a RA and an Architectural Pattern.

Another useful way to think about this dimension is in terms of *Architectural Decisions*. At the concrete/specific extreme, 'all' the Architectural Decisions have been made. At the other, very few Architectural Decisions are likely to have been made. Architectural Decisions are in inversally proportional to degrees of freedom. With few decisions being made there are many possibilities that the architecture can take and purposes it can serve. With more Architecture Decisions being made the architecture becomes more specific and concrete [Zimmerman, 2007].

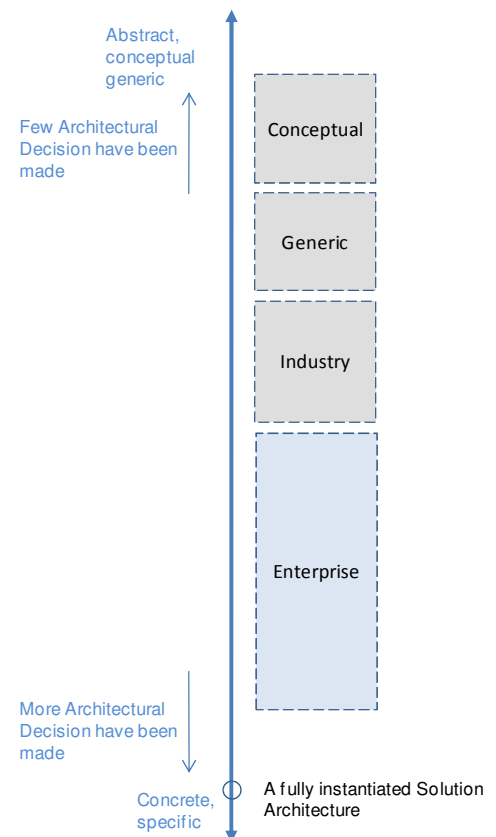


Figure 2: Reference Architecture can be defined at varying levels of abstraction from the conceptual and generic to the concrete and specific.

The Reference Architecture classification scheme

The classification scheme presented above is useful in sorting the myriad of RAs that are available, determining which are useful in a given situation, and how different RAs relate to each other.

¹ Some of the RAs at this end of the dimension tend to use the term Reference Model but this is not universal.

The diagram below shows some RA examples and where they fit in the scheme.

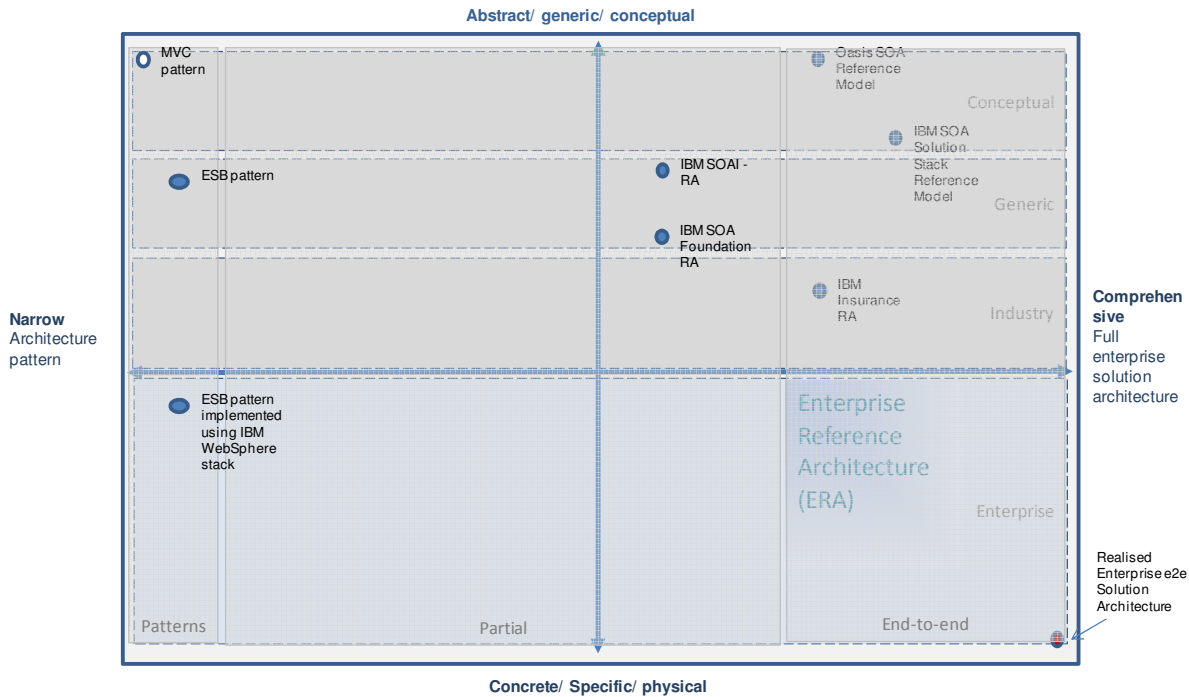


Figure 3: The Reference Architecture classification scheme can be used to position various instances of Reference Architectures according to their Coverage and Level of Abstraction and consequently how they are best used

I contend that thanks to the versatility of Reference Architectures, they can span the gap between EA (abstract and generic) and the Project-level Business Solution Architecture for a given business problem (concrete and specific). The RA classification scheme presented above shows that there are many types of RAs which range in coverage and abstraction. All these (at least the good ones) are useful but, based on my experience, when the RAs are made close enough to Solution Architectures their value to projects becomes enormous. There are a number of factors that determine the optimal closeness of a RA to an actual Solution Architecture. First, the more specific a RA is, the less flexible it becomes. Second, there is, of course, cost associated in moving from the generic and abstract to the specific and concrete.

The remainder of the paper focuses on the right lower corner (End-to-end & Enterprise) which I call Enterprise Reference Architecture (ERA).

Enterprise Reference Architectures (ERAs)

An ERA is a blueprint for the Solution Architecture of a number of potential projects within an organisation that embodies the EA principles, policies, standards and guidelines. In other words, an ERA is a Solution Architecture with some of the Architectural Decisions already made and others left open. ERAs have the following characteristics that distinguish them from other Reference Architectures and make them of great value for making EA effective in an organisation:

- They resemble actual Solution Architectures. This means that the effort to apply them by project-level architects is relatively low.
- They are applicable to a number of potential business solutions within the organisation. As mentioned earlier there is a trade-off between making an ERA close enough to an actual solution to lower the effort in applying it, and the number of business solutions where it is applicable. This tension combined with the cost of developing the ERAs provides the balance that determines the optimal Level of Abstraction and Coverage of an ERA within an organisation.

- Ideally, the development of an ERA should be funded directly by the business to address specific business objectives. This is a crucial aspect of ERA that can make or break the whole approach. It is possible to apply the approach discussed here by funding it from regular EA funds
- However, funding the effort of developing an ERA directly by the business for specific business objective as I did in another case study can profoundly transform the way architecture is practiced in the organisation. I refer below to this as Program-level Architecture.
- One key source of knowledge, experience and reusable components for the development and construction of ERAs must come from existing projects by way of harvesting suitable proven components and patterns.
- Architects who develop the ERAs are the best people who can adapt these ERAs for use in projects. This is key in changing the not-very-healthy typical dynamics of the relationship between EA and projects where projects view EA not as help but an obstacle to go around. By providing the ERA's architects as a paid-for resources from the EA to projects, or alternatively by seconding Solution Architects to the EA group to develop ERAs, the relationship between projects and the EA group can be greatly improved.

Program-level and Project-level Architecture

The ERA approach that I am outlining here aims to alter the typical relationship between EA and Project-level Architecture. It proposes replacing what is depicted on the left in Figure 4 below with that shown on its right. Interpretation by projects and conformance policing by Enterprise Architects are typically the only means to shape business solutions. In the new proposed relationship (on the right) ERA embodies the EA guidance and is used to help projects develop their Project-level Architecture. In addition, funding the development of an ERA directly by the business for a specific business initiative (a program of projects) can profoundly transform the way architecture is practiced in the organisation. The ERA for a business initiative or a program of projects is a Program-level Architecture.

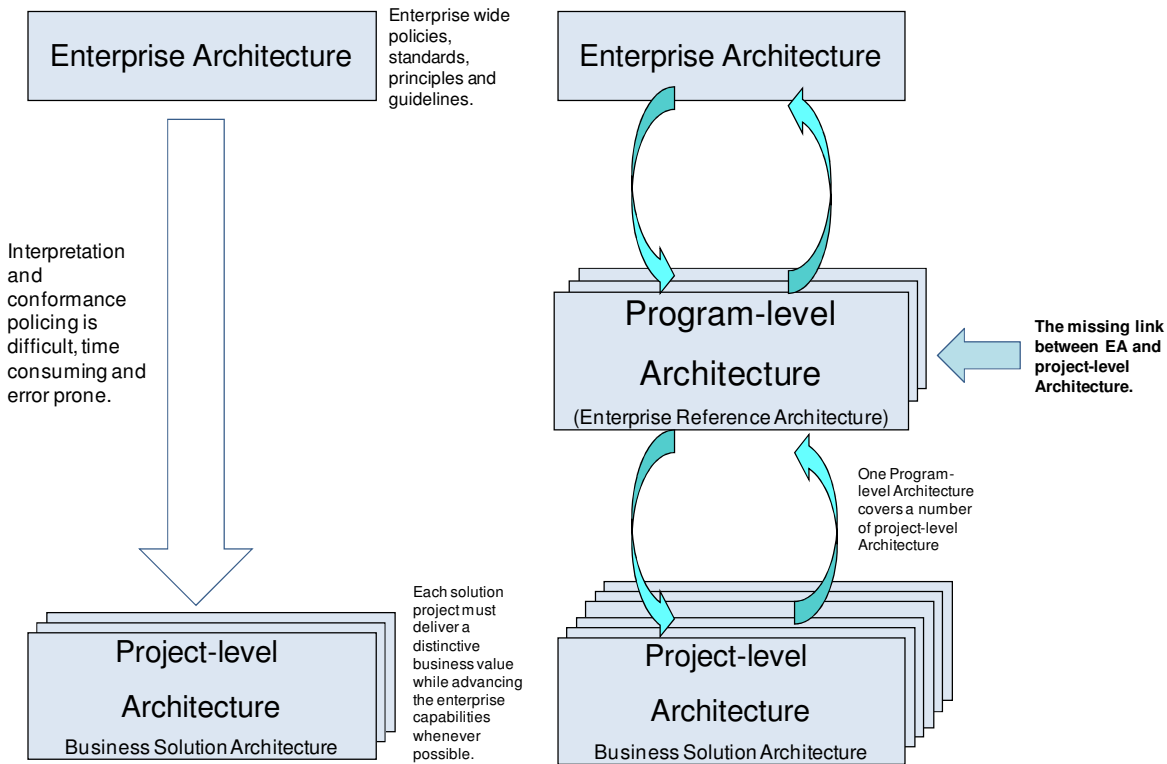


Figure 4: Typical and new relationship between Enterprise Architecture and project-level architecture

Service Oriented ERA

Although the ERA approach proposed in this paper applies to all architecture styles, it's more compelling and easier to apply with SOA because of SOA's emphasis on standardisation and reuse. One can actually find that all (or at least the majority of) the operational applications within an enterprise can be covered with a single Service Oriented ERA. Another positive contribution of SOA is the expansion of ERAs to cover both business and IT. The business aspects of an ERA should cover business processes (and of course Business Services), organisation and business information architecture.

Figure 5 shows the hierarchy of the SOA RAs that can be adopted and applied within an organisation culminating in a small number of ERAs that can be used to guide projects in creating SOA business solutions.

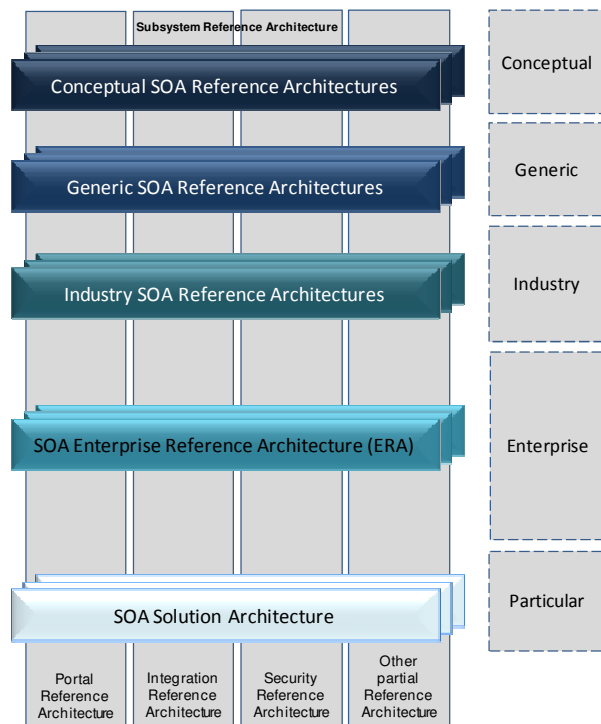


Figure 5: Hierarchy of SOA RAs from conceptual to specific

Conclusions and next steps

The paper overviewed an approach for enhancing the effectiveness of EA practices with the use of Enterprise Reference Architecture (ERA).

ERAs are Reference Architectures that embody the principles, policies, standards and guidelines of the enterprise in a form that is easily applicable to a set of business solutions. ERAs are ideally developed with large business initiatives.

The approach described here was developed and has been applied successfully in a number of practical situations.

I hope that some readers find the whole approach or some aspects of it useful for their organisations or clients. I welcome all feedback regarding the structure, contents or experiences related to applying the concepts discussed in the paper.

I am presently enhancing the approach and the concept of ERA based on a number of customer situations that I am currently involved in.

Acknowledgement

I would like to thank James Ward, Peter Budimir, Glenn Altern, Kathryn Kennedy, Kenny van Tong and Brian Kent for their input, provided through many discussions and/or reviews of earlier versions of this paper.

Ahmed Fattah

IBM

afattah@au1.ibm.com

References

[Zimmerman, 2007] Reusable Architectural Decision Models for Enterprise Application Development, Olaf Zimmermann et al, *Software Architectures, Components, and Applications*, 2007, <http://www.springerlink.com/content/e4j1473502644438/>