Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Using A Commercially Successful Empirical Enterprise Framework : Deterministic Service Oriented Architecture Service Levels

Ian McKeachie*, Ljubo Vlacic**

**Intelligent Control Systems Laboratory, Griffith University, Nathan Campus, Brisbane, Australia; (e-mail: ism@ datat3.com.au).*
*** Intelligent Control Systems Laboratory, Griffith University, Nathan Campus, Brisbane, Australia;(e-mail: l.vlacic@ griffith.edu.au).*

Abstract: This paper presents a summary of an integrated empirical enterprise framework and an aspect of its practical use. Development of the empirical framework, over some ten years, has been driven by management strategic goals and the need for successful commercial outcomes. What is offered as novel is the integrated empirical framework approach for a Service Oriented Architecture (SOA). The approach uses service interfaces' definitions at several logical levels within the Enterprise with mapping to component services thus allowing deterministic Service Level Agreement derivation using Failure Mode Effects Analysis (FMEA). As the empirical framework can incorporate arbitrary quantitative metrics, the technique allows management of risks with accurate budgeting, verifiable cost benefit design and comparison of proposed changes to existing enterprises on a cost benefit basis. Emphasis on integration of strategic frameworks and management paradigms with application and SOA components has been found to be critical to commercial success.

INTRODUCTION

A large body of knowledge of Enterprise Integration Architectures, Modelling and Methodologies (EIAM&M) has been developed addressing various types of Enterprises. Most viewpoints have been either static or cycle based, with the major drivers being design and implementation of Computer Integrated Manufacturing (CIM) plant and major engineering projects, including Virtual Enterprises. (Bernus & Nemes, 1996, pp 377-450; Williams, Rathwell & Li, 2001)

While a focus on interoperability and an agreed ontology is essential for EIAM&M in these types of industry, there was a lack of uptake in many other enterprises (Whitman, 1999; Whitman & Huff, 2001). Practical observation in consulting shows the major requirements are financial and economic efficiency and risk mitigation, rather than interoperability.

It is now generally empirically evident that specialisation for different industry types with heuristic based EIAM&Ms is necessary, as is a dynamic, evolutionary or living model, with cycle based models comprising a subset of these architectures. (Whitman, L., Ramachandran, K. & Ketkar, V., 2001)

Initial development of an empirical framework, beginning in 1995, was performed in isolation from the main body of knowledge by extending standard software engineering, capacity planning and failure mode effects and risk analysis.

As the scope of the study widened, it was quickly found that an integrated approach to the enterprises' requirements was essential. The level of integration necessary covered not only architectural areas such as technical design, business processes and capacity planning, but such areas as:

- Integration with management enterprise strategy and objectives, with feedback on objective success.
- Quantitative risk analysis and mitigation.
- Informal and social networks.
- Quantitative Cost benefit analysis.

A major driver was to allow reasonably accurate modelling of the effects of mergers, changes, new business initiatives or predicted organic growth for arbitrary risk or economic metrics.

Many of the various structures, views, process models, etc. used are not new, having been analysed in detail by various experts and built into standard reference models such as GERAM (ISO/IS 15704:2000), CIMOSA(CEN ENV 40003), PERA(Williams, 1992), etc.

However these frameworks do not intrinsically provide suitable dynamic or time varying economic or financial metrics, or integrated views that can be used for risk assessment or mitigation. These must be 'bolted on' by adding extra dimensions to the frameworks.

What is offered as novel in the empirical enterprise framework is successful integration of some features from these standard frameworks within a business, management and commercial viewpoint. A detailed presentation of the empirical framework is provided in Dynamic Enterprise Modelling for Knowledge Worker Industries (McKeachie & Vlacic, 2007a).

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.2719

This paper presents the application of the empirical framework to deterministic Service Oriented Architecture design and validation.

EMPIRICAL FRAMEWORK SUMMARY

Management Context

The empirical framework, with its populated data structures, is a management tool. It enables management to view the enterprise, define and measure arbitrary metrics and perform what if manipulation.

The empirical framework may be used to either :

- *Model* an existing enterprise by population of the data structures from the enterprise.
- Define a postulated target enterprise *architecture* by populating the data structures with a desired configuration.

These are two distinct uses. The model is where we are *now*; the enterprise architecture is where we want *to be*. It is usual to maintain these two populated frameworks in parallel.

Abstract Flows, Transforms and Interfaces

The Snapshot

At an instant in time, an enterprise is viewed as an integrated system, comprising the following abstract components:

- Atomic Generalized Business Processes, which transform, create or consume inputs and which generate or provide outputs. This can include 'Storage', 'Sources' and 'Sinks'.
- Atomic Generalized Flow (of information or material) between the Business Processes, each flow having flow properties such as rate, volume, probability distribution, etc.

Time function deltas can be incorporated in either Processes or Flows, either as simple constants or statistical functions.

The Enterprise logical and physical topology is represented by a set of directed graphs connecting Flow and Processes. This shows the current static state of the connections between and functions of the Business Processes. While the connectivity is static, the Flows between processes contain time dynamic information.

Metrics of any arbitrary sort may be specified by derivation from the model matrices (eg time through process, overall resource inputs, costs, risk & reliability, etc.) or by adding extra matrices.

Evolution : Variable Views, Variable Topology

The Snapshot now becomes a stepwise dynamic model, where the steps may be as small as required.

Operators may be used to perform a Generalized Transform or mapping of the model from the current snapshot form as represented by the components described above, to another form.

Again, this postulate of a Generalised Transform is as essential as the Generalised Process and Flow postulates are. This is a generalized Linear operator on the model vector space. This transformation may be achieved in one step or any number of smaller transforms.

By specifying the actual transform operator(s), a number of actual operations on the Business may be modelled, as well as various views projected.

The level of representation may be chosen as required, from atomic flows and processes (e.g. a manual process, single threads in a PC, etc.), up through composite (e.g. departmental, summary) form to an overall enterprise representation as a black box with inputs and outputs. This may be done by a suitably chosen projection transform operating on the most detailed level, then on the next, etc.

The model may be evolved in time by these transforms.

The topology may be changed by an operator acting on each component matrix in the model. This is used to step the model through reengineering/reconfiguration. This allows a 'what if' comparison to be performed, in conjunction with any given metric.

This also allows, given a postulated end form and a series of small transforms, the project plan steps to achieve that end form.

In conjunction with suitable metrics, the model topology can be optimized for those metrics.

Projections & Interfaces; Boundary Conditions

Between regions within the framework structure of generalised flows and transforms, as well as at the boundaries of the enterprise and the outside world, it is both possible and necessary to define interfaces. These are boundaries defined by either the nature of flows across the interface, or by a boundary interface transform.

In the case of a boundary transform, this is a projection, which may or may not be lossy (in an information sense) and may be single directional or bi-directional.

In the case of an interface defined by flows across the boundary, there is no loss, and the generalised flows map in each direction across the boundary.

These generalised components may also be used to:

1. Define the enterprises interaction with the external environment.
2. Define how virtual enterprises interact with their component enterprises.
3. Define internal interactions, including bi-directional services at different levels.
4. Produce views of the enterprise, or sections of the enterprise with varying levels of detail.
5. Map the enterprise modelled in this manner into (and from) other enterprise model representations.
6. These components may also be evolved in time in parallel with the flow and transform components.

Figure 1 below presents the highest or most abstract view of the empirical framework and the conceptual connectivity and mapping between detail layers.

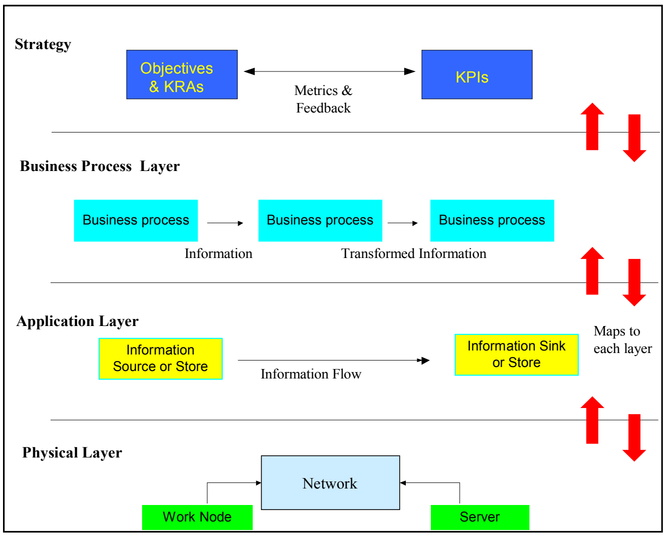

Figure 1 – High Level Enterprise View

Empirical Framework and SOA

Within the empirical framework, it has been found quite useful to consider the mapping between layers, as well as to external entities (i.e other enterprises, and the external environment: regulatory, financial, etc.) to be abstract interfaces.

The mappings across these boundaries are of two types. There is the business mapping, where the higher level defines what the lower level has to do to support the business strategic objectives. The second type is an abstract element that is considered to be a service primitive or element, which involves a component of abstract flow – which may be in either direction.

Domains at the same level, such as server, network and storage, or data and applications communicate across horizontal level interfaces.

Services that cross a boundary in the upward direction [towards the business process architecture in the case of the application domain, for example] are mapped to lower level or component services that are necessary to support the higher level service.

The same tenet applies in the other direction. For example, the "management" services, starting at the top strategic layer bifurcate into more elemental management or control streams as lower levels are reached.

This mapping must be consistent and not discontinuous. Any discontinuity generally also means that what enterprise management thinks is happening is in fact not. The shop floor is disconnected from management. (McKeachie & Vlacic, 2007a, p68)

It is the paradigm of generic interfaces comprising bi-directional generic services that gives the empirical framework the ability to decouple the different domains, and hence the actual SOA components, and therefore insulate the enterprise from future shock.

SOA METHODOLOGY

Summary Case Study

While we very rarely need to use the full power of the empirical model framework, in the model driven design [analysing and defining the component elements] of an SOA for a large government department we *did* need to go into considerable detail [which unfortunately can only be briefly summarised in this paper].

The process that was required may be summarised as:

- A current state enterprise audit was performed. [Where are we now?]
- The empirical framework was populated with this information, partly by automated data capture, to create a model of the enterprise.
- The model was analysed for discontinuities between layers, and domains, particularly in the application and business process layers. This revealed several issues that needed to be addressed in parallel with the SOA design.
- Extract the application and business process common services that were candidates for elements of SOA services; propose a set of these services, with logical and physical topology requirements. [What is the Service Oriented view of the enterprise business processes?]
- Analyse and state the necessary non-functional requirements for the end business processes that will be supported (availability, location, response time, costs, etc.) [e.g. What is the business continuity that is necessary; end to end performance, etc.?]
- Map the lower level service elements from the technology domain up to the business process level. [How do we provide the SOA high level services from the available low level elements?]

- Analyse and state what levels of non-functional specification would then exist (Gap analysis) [Does this mapping provide the necessary business continuity; performance, etc.?]
- Adjust the mapping, topology and grouping of the lower level services as necessary

It was necessary to iterate the last three tasks twice to optimize costs versus end to end Service Level Agreements.

Besides the SOA services definition and construction from lower level elements, we also analysed the model for overall:

- Capacity
- Performance
- Availability & Reliability
- Recovery time and data loss points
- Interdependence and de-coupling
- Business risk

The empirical framework also allowed these parameters to be predicted by statistical and FMEA analysis on a cost benefit basis. (.McKeachie & Vlacic, 2007b)

SOA Location in Empirical Framework

To reach the necessary level of detail, the high-level empirical framework has to be expanded as required. The first step is figure 2 below.

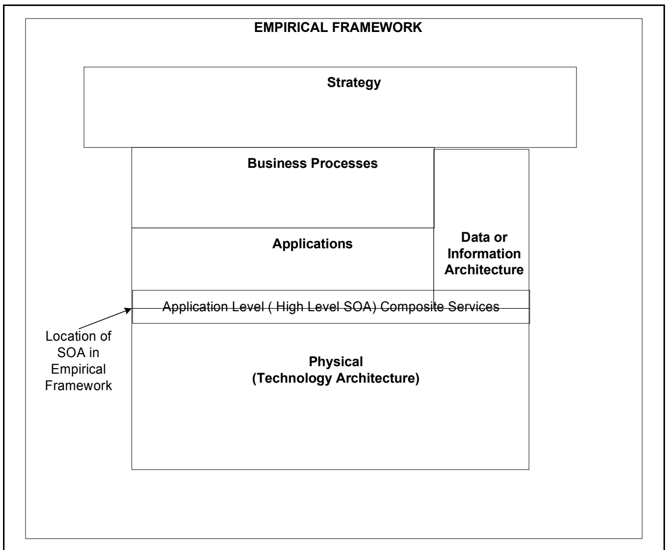

Figure 2 – SOA Location.

Figure 2 is a typical IT-centric view. The physical layer boundary has moved up into the lower levels of the application layer and is now referred to as the technology architecture. The higher application layer has been split between data structures (information architecture) and data manipulation (applications). The SOA location is across this interface.

An important point is that these detail changes to the empirical framework component layers did not break any enterprise wide metrics. These could still be evaluated as part of the design and verification process.

This process was now continued down to a level of detail sufficient for actual design, specification and implementation of the SOA. The necessary SOA services were built up from groupings of the empirical frameworks' lower level elemental flows, supporting the application and hence business process level.

Figure 3 below shows this detailed breakdown, which is the one used by the IT implementation and management team for the implementation of the SOA .

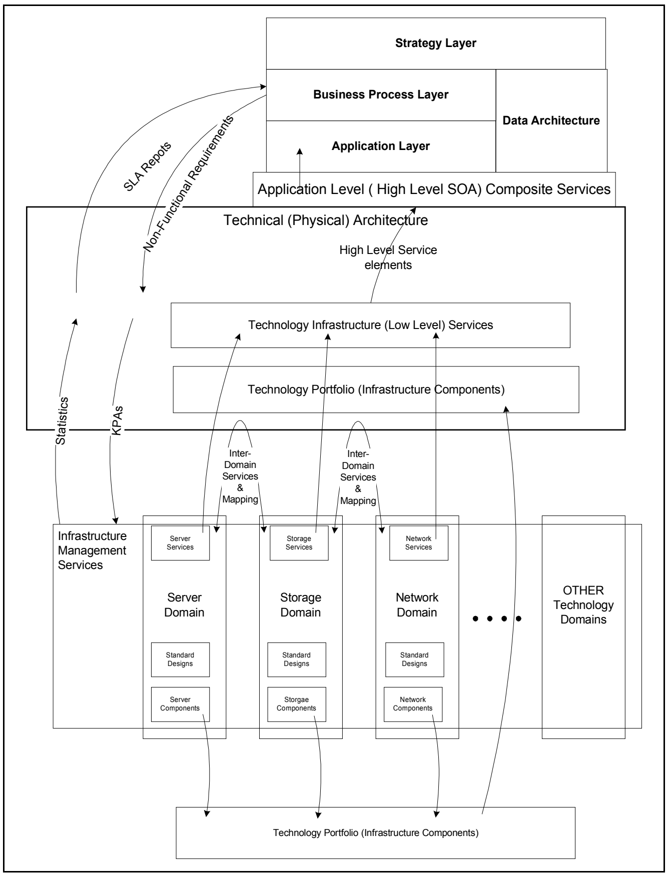

Figure 3 – Detailed Framework Technology Layer

The diagram above shows the detailed relationships between the empirical framework technology domains. Each domain is a portfolio of real infrastructure elements.

At the application level, only those services in which a business unit or application owner have an interest were included, with all additional details included in the lower level domains.

At a lower level, the Server, Network , Storage, etc. Domains provide primitive elements or more atomic services.

Note an SOA service does not map straight through to the low level. It is necessary to proceed from the SOA service, for example business process, through the SOA provided services to the lower level services, defining the low level service(s) that will indirectly support the higher level services.

SOA Core View

The diagram below expands the detail in the Application Level (High Level SOA) Services that provided the core of the SOA. This detailed view is shown in figure 4 below.

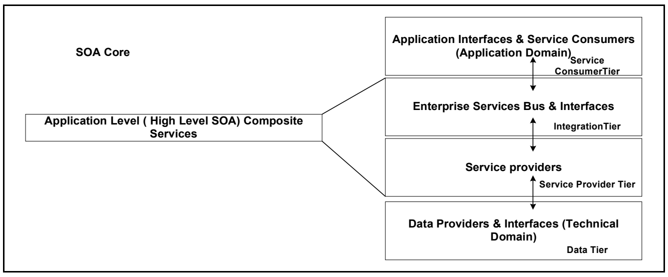

Figure 4 –SOA Core View within empirical framework.

The Integration layer, also known as an Enterprise Service Bus (ESB), provides the main abstraction and virtualisation layer that enables de-coupling of the SOA. It operates by providing transparent transport for messages between the service providers and the service consumers, as well as data responses. This part of the SOA is responsible for message, request and response queuing and providing in-order delivery of each. It included transform interfaces as necessary (to translate data, requests and messages between different logical entities' required formats)

At the lower level, the data tier is accessed by service providers, in response to service requests from the service consumers via the ESB.

Under the empirical framework SOA architecture, legacy applications support was by providing ESB interfaces to the legacy applications. These interfaces emulate the existing legacy interface to the data structures.

This facility has allowed the enterprise to migrate their higher level applications in a staged manner, minimising operational impacts. This process is currently continuing as the major legacy application is being retired.

CONCLUSIONS

By applying classical engineering and safety based FMEA mapped to the business process components, as well as end to end operations, it was possible generate real cost benefit metrics for financial or economic risk assessment and mitigation. The advantage of this application of FMEA techniques is the ability to provide quantitative assessment and management of enterprise wide risks, as well as SOA quality of service.

Using this approach, defining what Service Level Agreements (SLAs) are possible with a given topology, physical and technical components was an exercise in statistical FMEA. Probability rules for chains of connections could be applied to each element in the chain to validate a given design. (McKeachie & Vlacic, 2007b).

It was also possible to take the service delivery chains, including hardware and software elements, directly from the empirical framework data structures and export these (with their dependencies) into a formal IT Implementation Library (ITIL) Configuration Management Database (CMDB).

The ability to confirm and analyse the continuity in service flow and connection to upper business processes, including management strategy, from lower technology architecture components and outward between technology domains has allowed deterministic assessment of system design [management objectives as well as capacity, performance, cost, risks and reliability]

An integrated approach is one of the major commercial advantages of the use of the empirical framework, with further practical application of these techniques in consulting work being used to refine the empirical framework.

REFERENCES

Bernus, P, Nemes, L, & Williams, T.J. (Eds), 1996, *Architectures for Enterprise Integration*, Chapman & Hall, London.

CEN ENV 40003 – CIMOSA

ISO/IS 15704:2000 GERAM

McKeachie, I., & Vlacic, L., 2007a, Dynamic Enterprise Modelling for Knowledge Worker Industries, a chapter in *Enterprise Modelling in Adaptive Technologies and Business Integration: Social, Managerial and Organizational Dimensions*, Cunha. M., Cortes, B., & Putnik, G., (Eds), 2100 Books, Hershey.

.McKeachie, I., & Vlacic, L., 2007b, Enterprise Cost/Benefit Risk Analysis Using FMEA, an article in *Encyclopedia of Networked and Virtual Organizations,* Cunha. M., & Putnik, G., (Eds), IDEA Group, Hershey .

Whitman, L. 1999, *A living Model of the Enterprise*, Doctoral Dissertation, University of Wichita

Whitman, L. & Huff, B., 2001, On the Use of Enterprise Models, *Intl Jnl of Flexible Manufacturing Systems*, **13 2** pp195-208.

Whitman, L., Ramachandran, K., & Ketkar, V, 2001, A Taxonomy of a Living Model of the Enterprise, *Proceedings of the 2001 Winter Simulation Conference,* pp 848-855. [Peters, B.A., Smith, J.S., Medeiros, D.J. & Rohrer M.W. (eds)]

Williams, T.J, Rathwell, G. A & Li, H (Eds), 2001, A *Handbook on Master Planning and Implementation for Enterprise Integration Programs* – Based on the PERA and Purdue Methodology, Purdue Laboratory for Applied Industrial Control. Retrieved on April 25th 2003 from iies.www.ecn.purdue.edu/IIES/PLAIC/PERA/Publications