Enterprise Architecture and Systems Engineering

Peter Webb

BMT Defence Services Limited, United Kingdom, pwebb@bmtdsl.co.uk

Abstract: Factors such as the end of the cold war and increasing globalisation in terms of competition and social responsibility have significantly increased the complexity of events addressed both by governments and by industry. Furthermore, their responses are increasingly more constrained by economic and environmental issues. It is clear that the large complex “socio-techno-economic” systems (i.e. enterprises) that may comprise as well as create and deliver these responses must be both agile and efficient. An approach to the analysis, design and specification of such enterprises is presented which draws on state of the art Enterprise Architecture concepts and Systems Engineering techniques. The resulting EA/SE method enables clear justification of design, definition of interfaces and derivation of validated requirements. Comparisons are drawn to Zachman and ISO 15704 - GERA concepts.

1 INTRODUCTION

This paper describes how enterprise architecture concepts and systems engineering techniques have been combined to provide a new approach to the analysis, design and specification of enterprises. It has been named the Enterprise Architecture / Systems Engineering (EA/SE) method.

Enterprise architecture provides a rich but generic framework that guides the construction of enterprise models. Furthermore, it enables stakeholder needs and expectations to be identified and traced to technological design while facilitating integration efforts.

Systems engineering provides the tools and techniques to create “well engineered” enterprise architectures. In particular, it uses modelling as a means for analysis, improvement and validation of proposed solutions. Sys-

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35621-1_43](https://doi.org/10.1007/978-0-387-35621-1_43)

tems engineering also supports possible integration between interfacing enterprises or systems by identifying and defining interfaces.

The systems engineering core process first defines desired behaviour and determines structural options. Then the defined behaviour is allocated onto each structural option to create a range of potential solutions. Finally a trade-off is undertaken against pre-defined effectiveness measures in order to identify a near-optimal solution.

In this way an enterprise architecture description is supported by a number of products comprising diagrams, models and logic statements underpinned by a data repository. Where necessary architecture descriptions can be easily translated into requirement sets that are coherent, consistent, complete and validated as correct.

2 ENTERPRISE ARCHITECTURE

EA/SE considers an enterprise to be an organisation of people performing processes and supported by technology that cooperate to create and deliver solutions to customers (Fig.1). These solutions comprise products and/or services that may themselves be whole or part enterprises. Furthermore, an enterprise can interface with others to form a larger integrated enterprise.

Figure 1: Composition of an Enterprise

Examples of enterprises include procurement, support or design project teams and company organisations (people biased) as well as industrial facilities and ships, aircraft or other transport systems (technology biased).

The behaviour and structure of an enterprise is defined by its architecture. If an enterprise is to create and deliver quality solutions that, where practical, meet all stakeholders' needs and expectations within cost, schedule and risk

constraints then it should have an efficient, enabling architecture. Such enterprise architectures may be considered “well engineered”.

The EA/SE approach does not analyse the behaviour and structure of a whole enterprise at once. Instead, it progressively analyses the enterprise from successive degrees of abstraction, building the complete enterprise architecture in a structured manner. These degrees of abstraction provide perspectives of the architecture and are analogous to supporting layers where each layer defines the context and provides input data for the layer beneath (Fig. 2). The analysis activity starts at the most abstracted layer and works down through the layers. The systems engineering core process outlined in section 1 above is applied once to each layer.

Figure 2: EA/SE Layers of Abstraction

Table 1: Comparison of EA/SE Abstraction with ZEAF Perspectives and Lifecycle Phases of ISO 15704 (GERA) and Pre EN ISO 19439

EA/SE Abstraction	ZEAF Perspective	ISO 15704 - GERA & EN ISO 19439 Lifecycle Phase	
Capability / Market	Not Addressed	Domain Identification	
Asset / Business Unit	Not Addressed	Concept Definition	
System Context	Scope (Contextual) - Planner	Requirements Definition	
System Concept	Enterprise Model (Conceptual) - Owner	Preliminary Design	Design Specification
Logical Representation	System Model (Logical) - Designer	Detailed Design	
Physical Realisation	Technology Model (Physical) - Builder	Implementation Description	

The EA/SE layers of abstraction reflect the “perspectives” of the Zachman Enterprise Architecture Framework (ZEAF) (Zachman, <http://>). However, EA/SE adds two higher layers while missing the lowest ZEAF perspectives dealing with “Detailed Representation” and “Functioning Enterprise”.

Furthermore, the EA/SE layers are analogous to the life cycle phases of the Generic Enterprise Reference Architecture (GERA), (ISO15704, 1999) and (pre EN ISO 19439, 2002). These relationships are shown in Table 1 with typical outputs of EA/SE in Table 2.

Table 2: Typical EA/SE Outputs by Abstraction Layer

EA/SE Abstraction	Typical EA/SE Outputs
Capability / Market	<ul style="list-style-type: none"> – Analysed threats and incidents with desired countermeasures and responses. – Allocation to capability / market areas – Outline capability (user) requirements.
Asset / Business Unit	<ul style="list-style-type: none"> – Analysed capability behaviour – Detailed user requirements – Allocation to assets / business units – Outline system requirements
System Context	<ul style="list-style-type: none"> – Analysed asset / business unit behaviour – Detailed system requirements – Allocation to systems – Outline system architecture
System Concept	<ul style="list-style-type: none"> – Analysed system behaviour – Detailed system architecture – Allocation to sub-systems – Preliminary design
Logical Representation	<ul style="list-style-type: none"> – Analysed sub-system behaviour – Detail design Allocation to Objects – Outline build
Physical Realisation	<ul style="list-style-type: none"> – Analysed object behaviour – Allocation to components – Build description

Confusingly, in the ZEAF each “perspective” is divided into “abstractions” that deal with six fundamental interrogatives. Within EA/SE, each layer of abstraction (i.e. perspective) is similarly divided to address the six interrogatives, but they have been renamed as foci and are defined as follows:

- a) What entities are being processed? i.e. inputs and outputs;
- b) How are entities processed? i.e. functions (tasks, activities);
- c) When do processes occur? i.e. events and times (sequence);
- d) Where are processes occurring? i.e. locations and environmental conditions;
- e) Who is involved in processes? i.e. organisation, jobs, roles, skills (depending on perspective);

- f) Why are processes undertaken in the way they are? i.e. principles, standards.

The EA/SE approach aims to answer each of these interrogatives: Analysis of behaviour addresses the interrogatives “what”, “how” and “when”, while the consideration of structure addresses “where” and “who”. Similarly, consideration of motivation and purpose addresses the interrogative “why”. The abstraction layers and foci combine to form the EA/SE framework as shown in Table 3.

Table 3: The EA/SE Framework

EA/SE Abstraction	Behaviour			Structure		
	What?	How?	When?	Where?	Who?	Why?
Capability						
Asset						
System Context						
System Concept						
Logical Representation						
Physical Realisation						

The output from analysis is a description of the Enterprise Architecture model at the respective layer of abstraction. Combining the descriptions for each layer provides a description of the complete Enterprise Architecture model. In fact, the description of each architecture layer can take the form of a requirement specification for the layer beneath.

3 SYSTEMS ENGINEERING

The Systems Engineering Core Process adopted by EA/SE is adapted from Oliver, et al, (1997). The associated process flow diagram is reproduced in Fig. 3. The overall process description given in their book has been adapted in relation to the EA/SE approach and is included in the following paragraphs.

Step 1 evaluates and categorises available information for input to the modelling steps. It identifies and eliminates information redundancy, inconsistency and obvious incompleteness. In all but the first abstraction layer this is simply a matter of accepting context information output from

is simply a matter of accepting context information output from analysis in the preceding layer.

Figure 3: The Six Steps of the Systems Engineering Core Technical Process

Steps 2, 3, and 4 are concurrent activities. They can be ordered; however in practice it is found that they are highly inter-dependant and iteration is required. In other words, as understanding progresses in one of the tasks, it suggests changes in the other two.

Step 2 defines the criteria for optimisation. These are effectiveness measures and are written against the behaviour needs expressed in the input/context data. They can be represented by perhaps three to fifteen key requirements, even for large complex systems. They are the criteria that mean success or failure and against which options are assessed during trade-off analysis (see step 5).

Step 3 defines the behaviour that is desired. Behaviour is a rigorous description of what is to be done. It includes the functions to be performed, the sequencing control of those functions and the inputs and outputs from the functions. Two partial views of behaviour can be created diagrammatically: The first view, that tends to be scenario specific, shows functions and sequence, e.g. Functional Flow Block Diagrams, IDEF3 or Catalyst Process Charts. The second view is not scenario specific; it shows functions, inputs and outputs, e.g. Data Flow Diagrams, N-squared charts, or IDEF0. In addition, Entity-Relation diagrams can be used to simplify or “normalise” the

inputs and outputs. These views provide static models of behaviour. Either alternatively, or in addition, an executable model, such as Coloured Petri Nets, can be used to analyse behaviour.

When the desired behaviour is modelled separately from structure, then alternative structures can be readily identified. Step 4 defines these alternate structure models. In either step 3 or step 4, an allocation of modelled behaviour onto structure is made so that each option exhibits the same emergent behaviour. Further options can be derived through partitioning the modelled behaviour in different ways for each structure.

Step 5, trade-off, selects among the alternative architectures. The best feasible design is selected based on the effectiveness measure values defined in step 2. This is a critical, or key, best practice in the engineering of complex systems because it finds a near-optimal solution while guaranteeing the desired emergent behaviour. One possible branch from Step 5 is iteration back to the beginning, made necessary when no alternative architecture meets the effectiveness criteria. When this occurs, either steps 1 to 5 are repeated to find feasible solutions, or effectiveness criteria are relaxed so that a previous non-feasible solution is accepted.

Step 6 creates a plan for further modelling and implementation when a feasible and near optimal architecture has been found. The plan takes into account identified issues, risk reduction, etc. It may also include a description of the architecture at the related layer of abstraction in the form of requirements for the next layer.

The six systems engineering core steps are applied once for each layer of abstraction forming a complete Enterprise Architecture model. As stated previously, this analysis activity should start at the most abstracted layer and work down through the layers since each layer defines the context and provides input data for analysis in the layer beneath.

4 CONCLUSION

A method for analysing, designing and specifying enterprises that can comprise complex technological or business systems has been presented. Known as Enterprise Architecture / Systems Engineering (EASE), it combines enterprise architecture principles with systems engineering techniques.

EA/SE simplifies the modelling of Enterprise Architectures by using a relatively small number of constructs from the systems engineering process and applying these successively to layers of abstraction. The result is a rich but simple model that enables user and other stakeholder needs to be traced to technical design. Furthermore, justifiable and validated requirements can be derived from the model against each layer of abstraction.

5 REFERENCES

- ISO 14258, (1999), *Concepts and rules for enterprise models*, available from <http://www.mel.nist.gov/sc5wg1/>
- ISO 15704, (1999), *Industrial automation systems - Requirements for Enterprise Reference Architectures and Methodologies*, <http://www.mel.nist.gov/sc5wg1/>
- Oliver, D. W. Kelliher T. P. Keegan Jr, J. G. (1997), *Engineering Complex Systems with Models and Objects*, downloadable free from <http://www.incose.org/newsline.html>
- Pre EN ISO 19439, (2002), *Enterprise Integration - Framework for Enterprise Modelling*, CEN TC 310, WG1, ISO TC 84 SC5 WG1.
- Zachman Enterprise Architecture Framework, <http://www.zifa.com>

6 ADDITIONAL REFERENCES

- Levis, A. H Wagenhals, L.W. (2000), *C4ISR Architectures 1: Developing a Process for C4ISR Architecture Design*, Journal of INCOSE Vol. 3(4), Wiley ISSN 1098-1241, available from <http://viking.gmu.edu/http/courses.htm>
- US Chief Information Officers (CIO) Council, (1999), *Federal enterprise Architecture Framework*, Version 1.1, available from <http://www.cio.gov/>
- US Chief Information Officers (CIO) Council, (2000), *A Practical Guide to Federal Enterprise Architecture*, version 1, available from <http://www.cio.gov/> or directly from http://www.itpolicy.gsa.gov/mke/archplus/ea_guide.doc
- US Chief Information Officers (CIO) Council, (2000), *Architecture Alignment and Assessment Guide*, available from <http://www.cio.gov/>
- US DOD, (1997), *C4ISR framework* Version 2.0, available from <http://viking.gmu.edu/http/c4isrAFI/archfwk2.pdf>
- Vernadat, F. B. *Enterprise Modelling and Integration: Myth or Reality*, available from <http://www.cit.gu.edu.au/~bernus/taskforce/>
- Vernadat, F. B. (1997), *Enterprise Modelling Languages*, ICEIMT'97 Enterprise Engineering and Integration - International Consensus, Springer-Verlag, <http://www.mel.nist.gov/workshop/iceimt97/pap-ver3/pap-ver3.htm>