Enterprise Systems Engineering Key Concepts

From SEBoK
Enterprise Systems Engineering Key Concepts

Lead Authors: *James Martin, Bud Lawson, Alan Faisandier*

The purpose of traditional systems engineering (TSE) is to bring together a diversity of discipline experts to address a wide range of problems inherent in the development of a large, complex "single" system (Blanchard and Fabrycky 2010; Hall 1989; Sage and Rouse 2009). Enterprise systems engineering (ESE) expands beyond this traditional basis to "consider the full range of SE services increasingly needed in a modern organization where information-intensive systems are becoming central elements of the organization's business strategy" (Carlock and Fenton 2001, 242-261). The traditional role of systems engineering (SE) is heavily involved in system acquisition and implementation, especially in the context of government acquisition of very large, complex military and civil systems (e.g., F22 fighter jet and air traffic control systems).

ESE encompasses this traditional role in system acquisition, but also incorporates enterprise strategic planning and enterprise investment analysis (along with others as described below). These two additional roles for SE at the enterprise level are "shared with the organization's senior line management, and tend to be more entrepreneurial, business-driven, and economic in nature in comparison to the more technical nature of classical systems engineering" (Carlock and Fenton 2001, 242-261).

Contents

Closing the Gap

ESE practices have undergone significant development recently.

> *Today the watchword is enterprise systems engineering, reflecting a growing recognition that an 'enterprise' may comprise many organizations from different parts of*

government, from the private and public sectors, and, in some cases, from other nations. (MITRE 2004)

Rebovich (2006) says there are “new and emerging modes of thought that are increasingly being recognized as essential to successful systems engineering in enterprises.” For example, in addition to the TSE process areas, MITRE has included the following process areas in their ESE process (DeRosa 2005) to close the gap between ESE and PSE:

- strategic technical planning,
- enterprise architecture,
- capabilities-based planning analysis,
- technology planning, and
- enterprise analysis and assessment.

These ESE processes are shown in the context of the entire enterprise in the figure below (DeRosa 2006). The ESE processes are shown in the middle with business processes on the left and TSE processes on the right. These business processes are described in the article called Related Business Activities. The TSE processes are well documented in many sources, especially in the ISO/IEC/IEEE 15288 standard (2015).

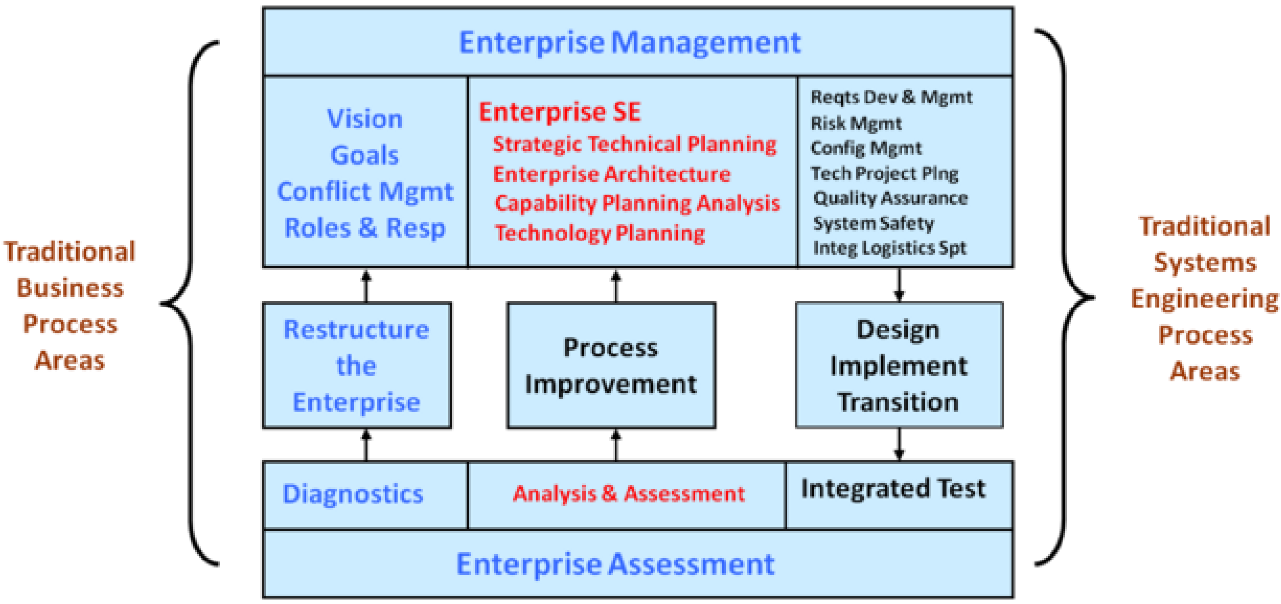

Figure 1. Enterprise SE Process Areas in the Context of the Entire Enterprise (DeRosa 2006).

SE is viewed by many organizations and depicted in many process definitions as bounded by the beginning and end of a system development project. In MITRE, this restricted definition was referred to as TSE. Many have taken a wider view seeking to apply SE to the “whole system” and “whole life cycle.” For example, Hitchins (1993) sets out a holistic, whole-life, wider system view of SE centered on operational purpose. Elliott and Deasley (2007) discuss the differences between development phase SE and in-service SE.

In contrast to TSE, the ESE discipline is more like a “regimen” (Kuras and White 2005) that is responsible for identifying “outcome spaces,” shaping the development environment, coupling development to operations, and rewarding results rather than perceived promises (DeRosa 2005). ESE must continually characterize the operational environmental and the results of enterprise or SoS interventions to stimulate further actions within and among various systems in the enterprise portfolio. Outcome spaces are characterized by a set of desired capabilities that help meet enterprise objectives, as opposed to definitive “user requirements” based on near-term needs.

Enterprise capabilities must be robust enough to handle unknown threats and situations in the future. A detailed description of previous MITRE views on ESE can be found in a work by Rebovich and White (2011).

Role of Requirements in ESE

TSE typically translates user needs into system requirements that drive the design of the system elements. The system requirements must be “frozen” long enough for the system components to be designed, developed, tested, built, and delivered to the end users (which can sometimes take years, and in the case of very large, complicated systems like spacecraft and fighter jets, more than a decade).

ESE, on the other hand, must account for the fact that the enterprise must be driven not by requirements (that rarely can even be defined, let alone made stable), but instead by continually changing organizational visions, goals, governance priorities, evolving technologies, and user expectations. An enterprise consists of people, processes, and technology where the people act as “agents” of the enterprise:

> *Ackoff has characterized an enterprise as a 'purposeful system' composed of agents who choose both their goals and the means for accomplishing those goals. The variety of people, organizations, and their strategies is what creates the inherent complexity and non-determinism in an enterprise. ESE must account for the concerns, interests and objectives of these agents.* (Swarz, DeRosa, and Rebovich 2006) (See also Complexity)

Whereas TSE focuses on output-based methodologies (e.g., functional analysis and object-oriented analysis), ESE is obligated to emphasize outcomes (e.g., business analysis and mission needs analysis), especially those related to the enterprise goals and key mission needs.

Enterprise Entities and Relationships

An enterprise “system” has different entities and relationships than you might find in a product/service system (see note 1). These can be usefully grouped into two categories: asset items and conceptual items. An example of an asset is hardware and software. Examples of conceptual items are things like analysis, financial elements, markets, policies, process, and strategy.

> *Note 1. An “enterprise system” should not be confused with the enterprise “perceived as a system.” An enterprise system is a product (or service) system used across the enterprise, such as payroll, financial accounting, or enterprise resource planning applications, and consolidated data center, data warehouse, and other such facilities and equipment used across one or more organizations.*

Products and services are sometimes treated as “assets” as shown in the figure below (Troux 2010). This categorization of enterprise items comes from the semantic model (i.e., metamodel) used in the Troux Architect modeling tool for characterization and analysis of an enterprise architecture. Other enterprise entities of interest are things like information, knowledge, skills, finances, policies, process, strategy, markets, and resources, but these are categorized as "concept" items (in this particular schema). Further details on how to use this metamodel's entities and relationships are provided by Reese (2010).

Table 1. Asset Domain and Concept Domain Categories for Enterprise Entities. (Troux 2010)

Asset Domains

Application and Software Domain
Data Domain Document Domain Infrastructure and Hardware Domain IT Product Domain IT Service Domain Location Domain Organization Domain Product and Service Domain Services Portfolio Management Domain

Concept Domains

Analysis Domain
Financial Domain General Domain Information Domain IT Architecture Domain Knowledge and Skill Domain Market Domain Policy Domain Process Domain Resource Domain Strategy Domain Timeline Domain Transition Domain

The application/software and infrastructure/hardware domains are likely the most familiar to systems engineers (as illustrated in the figure below). The application/software domain contains things like the deployed software itself, plus applications, modules, servers, patches, functions, and messages. The infrastructure/hardware domain contains things like the hardware itself, plus networks and different kinds of hardware like computing hardware, cabinets, and network devices. There might be different subtypes of computing hardware like computers, servers, desktops, laptops, and mainframes. You can see from this elaboration of these domains that an enterprise architecture "schema" can be quite extensive in the kinds of things it can model.

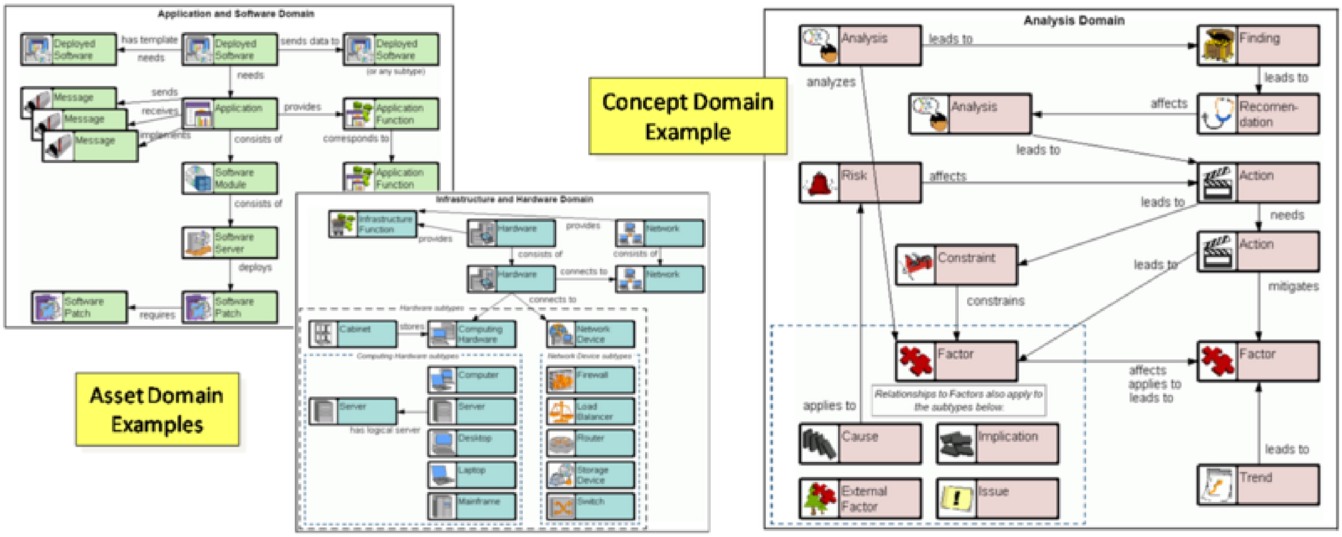

Figure 2. Example of Enterprise Entities & Relationships (Troux 2010). Reprinted with permission of Copyright © 2010 Troux Technologies. All other rights are reserved by the copyright owner.

The less technical domains would be things like policy, market, strategy, transition, financial, knowledge and skill, and analysis. In a typical enterprise architecture schema like this, there could be over a hundred types of modeling objects grouped into these domains. The examples give above are from the Troux Semantics metamodel used in the Troux Architect modeling tool for enterprise architecture activities. Other enterprise modeling tools have similar metamodels (sometimes called “schemas”). See Reese (2010) for more details on how to use the metamodel shown in the figure above.

Enterprise Architecture Frameworks & Methodologies

Enterprise architecture frameworks are collections of standardized viewpoints, views, and models that can be used when developing architectural descriptions of the enterprise. These architecture descriptions can be informal, based on simple graphics and tables, or formal, based on more rigorous modeling tools and methods. ISO/IEC 42010 (2011) specifies how to create architecture descriptions.

These frameworks relate to descriptive models of an enterprise, with conventions agreed in particular communities. There are various frameworks and methodologies available that assist in the development of an enterprise architecture.

Urbaczewski and Mrdalj (2006) provide an overview and comparison of five prominent architectural frameworks, including:

- the Zachman Framework for Enterprise Architecture (Zachman 1992),
- the Department of Defense Architecture Framework (DoDAF) (DoD 2010),
- the Federal Enterprise Architecture Framework (FEAF) (FEA 2001),
- the Treasury Enterprise Architecture Framework (TEAF) (US Treasury 2000),
- and The Open Group Architectural Framework (TOGAF) (TOGAF 2009).

References

Works Cited

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall.

Carlock, P., and R. Fenton. 2001. "System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations." *Systems Engineering*. 4 (4): 242-261.

CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*, version 1.1. Washington, DC, USA: Federal Chief Information Officers Council.

DeRosa, J.K. 2005. "Enterprise Systems Engineering." Presented at Air Force Association, Industry Day, Day 1, August 4, 2005, Danvers, MA, USA.

DoD. 2010. *DoD Architecture Framework (DoDAF)*, version 2.0. Washington, DC: U.S. Department of Defense (DoD).

Elliott, C., and P. Deasley. 2007. *Creating Systems that Work--Principles of Engineering Systems for the 21st Century*. London, England, UK: Royal Academy of Engineering.

FEA. 2001. "Federal Enterprise Architecture – Practical Guide, version 1.0, February 2001." Available: https://secure.cio.noaa.gov/hpcc/docita/files/a_practical_guide_to_federal_enterprise_architecture.pdf.

Friedman, G., and A.P. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition." *Systems Engineering*. 7 (1): 84-96.

Hall, A.D. 1989. *Metasystems Methodology: A New Synthesis and Unification*, 1st ed. Oxford, UK: Pergamon Press.

Hitchins, D. 1993. *Putting Systems to Work*. New York, NY, USA: John Wiley & Sons.

ISO/IEC/IEEE. 2015.*Systems and Software Engineering - System Life Cycle Processes*.Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers. ISO/IEC/IEEE 15288:2015.

Kuras, M.L., and B.E. White. 2005. "Engineering Enterprises Using Complex-Systems Engineering." Annotated presentation at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 10-15, 2005, Rochester, NY, USA.

MITRE. 2004. *MITRE 2004 Annual Report". McLean, VA, USA: MITRE Corporation.*

Rebovich, G. 2006. "Systems Thinking for the Enterprise: New & Emerging Perspectives." Presented at IEEE/SMC International Conference on System of Systems Engineering, April 2006, Los Angeles, CA, USA.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Reese, R.J. 2010. *Troux Enterprise Architecture Solutions*. Birmingham, UK: Packt Publishing Ltd.

Sage, A.P., and W.B. Rouse (eds). 2009. *Handbook of System Engineering and Management*, 2nd ed. New York, NY, USA: John Wiley & Sons.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." Proceedings of the 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

TOGAF. 2009. "The Open Group Architecture Framework," version 9. Accessed September 7, 2011. Available: http://www.opengroup.org/togaf/.

Troux. 2010. *Metamodeling and modeling with Troux Semantics*, version 9. Austin, TX, USA: Troux Technologies.

Urbaczewski, L., and S. Mrdalj. 2006. "A Comparison of Enterprise Architecture Frameworks." *Issues in Information Systems*. 7 (2): 18-26.

US Treasury. 2000. *Treasury Enterprise Architecture Framework,* version 1. Washington, DC, USA: US Department of the Treasury Chief Information Officer Council.

Zachman, J.A. 1992. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal*. 31 (3): 590-616.

Zachman, J.A. 1987. "A Framework for Information Systems Architectures." *IBM Systems Journal*. 26 (3): 276-92.

Primary References

Kuras, M.L., and B.E. White. 2005. "Engineering Enterprises Using Complex-Systems Engineering." Annotated presentation at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 10-15, 2005, Rochester, NY, USA.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." Proceedings of the 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

Additional References

Journal of Enterprise Architecture. Available: http://www.globalaea.org/?page=JEAOverview.

Minoli, D. 2008. *Enterprise Architecture A to Z: Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, An Auerbach Book.

TRAK. 2011. "TRAK Enterprise Architecture Framework." Accessed September 7, 2011. Available: http://trak.sourceforge.net/index.html.

Vernadat, F.B. 1996. *Enterprise Modelling and Integration - Principles and Applications*. London, UK: Chapman and Hall.

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.3, released 30 October 2020