Enterprise Systems Engineering Key Concepts

From SEBoK
Enterprise Systems Engineering Key Concepts

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, “single” (Blanchard and Fabrycky 2010; Hall 1989; Sage and Rouse 2009). (ESE) expands beyond this traditional basis to “consider the full range of SE services increasingly needed in a modern where information-intensive systems are becoming central elements of the organization’s 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 of government 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 strategic ning 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

1 Closing the Gap
2 Role of Requirements in ESE
3 Enterprise Entities and Relationships
4 Enterprise Architecture Frameworks & Methodologies
5 References
   5.1 Works Cited
   5.2 Primary References
   5.3 Additional References

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 areas, MITRE has included the following process areas in their ESE process (DeRosa
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).

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, 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 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 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)
Reprinted with permission of Copyright © 2010 Troux Technologies. All other rights are reserved by the copyright owner.

<table>
<thead>
<tr>
<th>Asset Domains</th>
<th>Concept Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application and Software Domain</td>
<td>Analysis Domain</td>
</tr>
<tr>
<td>Data Domain Document Domain Infrastructure and Hardware Domain IT Product Domain IT</td>
<td>Financial Domain General Domain Information</td>
</tr>
<tr>
<td>Service Domain Location Domain Organization</td>
<td>Domain IT Architecture Domain Knowledge and Skill Domain Market Domain Policy Domain</td>
</tr>
<tr>
<td>Domain Product and Service Domain Services</td>
<td>Process Domain Resource Domain Strategy Domain Timeline Domain Transition Domain</td>
</tr>
<tr>
<td>Portfolio Management Domain</td>
<td></td>
</tr>
</tbody>
</table>

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.

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 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


International Symposium, July 9-13, 2006, Orlando, FL, USA.


Primary References


Additional References


< Previous Article | Parent Article | Next Article >

SEBoK v. 2.0, released 1 June 2019

Retrieved from
