The single most important principle of the systems approach is that it is applied to an engineered system context and not just to a single system (INCOSE 2012). The systems approach includes models and activities useful for the understanding, creation, use, and sustainment of engineered systems to enable the realization of stakeholder needs. Disciplines that use a systems approach (like systems engineering (SE)) consider an engineered system context that defines stakeholder needs, and look for the best ways to provide value by applying managed technical activities to one or more selected engineered systems of interest (SoI).

Generally, four specific types of engineered system contexts are recognized in SE:

- product system
- service system
- enterprise system
One of the key distinctions between these system contexts pertains to the establishment of how and when the SoI boundary is drawn.

### Contents

- Engineered System-of-Interest
- Applying the System Context
- Product System Context
- Service System Context
- Enterprise System Context
- References
  - Works Cited
  - Primary References
  - Additional References

### Engineered System-of-Interest

We use the idea of an engineered system context to define an engineered SoI and to capture and agree on the important relationships between it, the systems with which it works directly, and any other systems with which it works. All applications of a systems approach (and hence of SE) are applied in a system context rather than only to an individual system.

A system context can be constructed around the following set of open system relationships (Flood and Carson 1993):

- The **Narrower System-of-Interest** (NSoI) is the system of direct concern to the observer. The focus of this system is driven by the scope of authority or control with implicit recognition that this scope may not capture all related elements.

- The **Wider System-of-Interest** (WSoI) describes a logical system boundary containing all of the elements needed to fully understand system behavior. The observer may not have authority over all of the elements in the WSoI but will be able to establish the relationships between WSoI elements and NSoI elements.
The WSol exists in an **environment**. The immediate environment contains engineered, natural, and/or social systems, with which the WSol (and thus some elements of the NSol) directly interact for the purpose of exchanging material, information, and/or energy to achieve its goals or objective.

A **Wider Environment** completes the context and contains systems that have no direct interaction with the SoI, but which might influence decisions related to it during its life cycle.

"Some Theoretical Considerations of Mathematical Modeling" (Flood 1987) extends this context to include a **meta-system** (MS) that exists outside of the WSol and exercises direct control over it.

The choice of the SoI boundary for particular activities depends upon what can be changed and what must remain fixed. The SoI will always include one or more NSol but may also include WSol and an MS if appropriate, such as when considering a service or an enterprise system.

**Applying the System Context**

For lower-level and less-complex systems, the WSol can represent levels of a product system hierarchy. An example of this would be an engine management unit as part of an engine, or an engine as part of a car. The WSol in a system context may encapsulate some aspects of SoS ideas for sufficiently complex systems. In these cases, the WSol represents a collection of systems with their own objectives and ownership with which the NSol must cooperate in working towards a shared goal. An example of this would be a car and a driver contributing to a transportation service.

This view of a SoS context being used as a means to support the engineering of an NSol product system is one way in which a systems approach can be applied. It can also be applied directly to the SoS. Examples of this include a flexible multi-vehicle transportation service or transportation as part of a commercial enterprise. In this case, the NSol aspect of the context no longer applies. The WSol will consist of a set of cooperating systems, each of which might be changed or replaced to aid in the synthesis of a solution. The context may also need to represent **loose coupling**, with some systems moving in or out of the context depending on the need, or **late binding** with systems joining the context only at, or
close to, the delivery of the service.

Thus, a context allows a reductionist view of the SoI that is of direct concern to an observer, as it provides for the system relationships and influences that are needed to maintain a holistic view of the consequence of any actions taken.

**Product System Context**

The distinction between a product and a product system is discussed in the article *Types of Systems*.

A product system context would be one in which the SoI is the product itself. The wider system context for a product system can be a higher level of product hierarchy, a service, or an enterprise system that uses the product directly to help provide value to the user. A significant aspect of a product systems context is the clear statement of how the product is intended to be used and ensures that this information is given to the acquirer upon delivery. The customer will be required to accept the system, typically through a formal process, agreeing not to go against the terms of use.

If a systems approach is applied to a product context, it is done with the purpose of engineering a narrow system product to be integrated and used in a wider system product hierarchy or to enable the delivery of a wider system service directly to a user by an enterprise.

This view of the relationship between product and service is specific to product systems engineering. While some engineering of the acquirer's static service system may occur, it is done with a product focus. The definition of service system in a service systems engineering context describes a more dynamic view of service systems.

**Service System Context**

Services are activities that cause a transformation of the state of an entity (people, product, business, and region or nation) by mutually agreed terms between the service provider and the customer (Spohrer 2008). The distinction between service and a service system is discussed in the article *Types of Systems*.

A service system context is one in which the SoI is the service system. This SoI contains all of the technology, infrastructure, people, resources, etc. that are needed to
enable the service. The WSoI describes the enterprise providing the service as well as its relationship with other services that impact the success of the enterprise.

If a systems approach is applied to a service system, it is done with the purpose of engineering a service system to enable the outcomes required by an enterprise to satisfy its clients. When operating in the service system context, all options to provide the service must be considered, providing that they fit within the constraints of the enterprise. This will include interfaces to other services, people, and resources in the enterprise. If an option for providing the service makes use of existing products or resources within or outside of the enterprise, it must be ensured that they are available for this use and that this does not adversely affect other services. Part of getting the right service may require the negotiation of changes to the wider enterprise context, but this must be by agreement with the relevant authority.

For a service system, and also when considering the service system context, the value is realized only through service transactions. The end-user co-creates value at the time of the request to use the service. For example, to make a flight reservation using a smartphone, the service system is composed of many service system entities (the caller, the person called, the smartphone, the access network, the core Internet Protocol (IP) network, the Internet Service provider (ISP), the World Wide Web (WWW), data centers, etc. All these are necessary to enable the service. When a caller makes a reservation and then books the flight, the value has been created.

This definition of a service system, as associated with dynamic Information Technology (IT) services, is discussed further in the article Service Systems Engineering.

**Enterprise System Context**

The distinction between an enterprise and an enterprise system is discussed in the article Types of Systems.

An enterprise system context is one in which the SoI is the enterprise system. This system contains all of the technology, infrastructure, people, resources, etc. needed to enable the service. The WSoI describes the business environment within which the enterprise sits.

It is to be noted that an enterprise context is not equivalent to an organization according to this
definition. An enterprise includes not only the organizations that participate in it, but also the people, knowledge, and other assets, such as processes, principles, policies, practices, doctrines, theories, beliefs, facilities, land, and intellectual property that compose the enterprise.

An enterprise may contain or employ service systems along with product systems. An enterprise might even contain sub-enterprises. Enterprise systems are unique when compared to product and service systems in that:

- they are constantly evolving
- they rarely have detailed configuration controlled requirements
- they typically have (constantly changing) goals of providing shareholder value and customer satisfaction
- they exist in a context (or environment) that is ill-defined and constantly changing

The enterprise systems engineer must consider and account for these factors in their processes and methods.

Both product and service systems require an enterprise system context to create them and an enterprise to use the product system and deliver services, either internally to the enterprise or externally to a broader community. Thus, the three types of engineered system contexts are linked in all instances, regardless of which type of system the developers consider as the object of the development effort that is delivered to the customer.

References

Works Cited


3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.


**Primary References**


**Additional References**


