Chapter 2
The Systems Engineering

Abstract Many definitions of the Systems Engineering are proposed, and analyzing some differences between them help to highlight contents and open issues, as they are described in this chapter. Nevertheless, a short historical outline could be helpful in appreciating some characteristics of this methodology, which are even more detailed by a wide literature, herein briefly described. Furthermore, an overview of technical standards dealing with the Systems Engineering is added, to define a roadmap for a deeper education in this field.

2.1 A Definition in a Nutshell

If one reads the definition proposed by the International Council on Systems Engineering (INCOSE), this interdisciplinary approach is described as "*a mean to enable the realization of successful systems*" and "*focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem*". Moreover "*it integrates all the disciplines and specially groups into a team effort forming a structured development process that proceeds from concept to production to operation*". It "*considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs*" (Walden, 2015).

The above and precise definition probably is the fastest way to describe at least the most important contents of the Systems Engineering (in the following simply SE), which doesn't substitute all the techniques of engineering the systems in a more general meaning, but provides new means to do this by following a more systematic approach. It can be immediately realized that some highlights are characteristic of the proposed approach and even cited by the above definition. Object of design must be a *system*, being naturally based on the interaction among different *components*, to be *assembled* together and *connected*. Success of product developed is directly related to the *customer satisfaction*, although suitable *metrics* to measure that satisfaction have to be defined. Product must provide only *functions*

E. Brusa et al., *Systems Engineering and Its Application to Industrial Product Development*, Studies in Systems, Decision and Control 134,
https://doi.org/10.1007/978-3-319-71837-8_2

strictly *required*, since its conceptual design, i.e. very early in the whole development path. A clear sequence of operations in development is suggested, thus starting from requirements, to define the design contents till the synthesis, being a first intermediate goal of the whole activity. *Validation* is then proposed as a necessary step to investigate how much the product fits the customer needs. This interpretation leads to the description of a *process*, from concept to operation, being referred to as *structured*, i.e. fairly well defined to be linear, repeatable and effective, and strictly oriented to *quality* and to the *user*. Moreover, the two main worlds of *technology* and *business* are both considered, thus suggesting the need for a clear *modeling* the technical contents as well as the business foreseen for that product.

Despite of the brightness of sentences, implementation of the SE is never so straight, especially in the hardware domain. It might happen that the real contents of those sentences are poorly understood, if the Reader doesn't know some goals which motivated the creation of the SE and some principal pillars of this approach. Screening those two issues may help in implementing it more easily.

2.1.1 Main Goals

Nowadays systems conceiving, designing, developing and integrating are all key strategic goals of the product development, especially in industrial manufacturing, when a direct material processing technology is exploited. Practically speaking, some needs could be detected.

Suitable tools for handling complexity. Product is today a *system of subsystems*, *components and parts*, and often embeds some kind of *smartness* and some *capability of communicating*. Consequently the number of functions exploited is increasing, together with the number of interfaces. As an example, one may think about electronic control units and sensors, which are often applied to mechanical and electromechanical elements, thus creating a full mechatronic product. It has to be manufactured through the contribution of several technical competences and a suitably design of each energy conversion exploited by the system has to be performed. This leads to a certain *complexity*, poorly manageable through the tools widely used in the past.

Traceability over the whole lifecycle. Nature of new products intrinsically requires a longer support after delivery, to perform monitoring, maintenance and control, and to assure a regular service. Cost might grow up fairly fast for the manufacturer, if the whole *Product Lifecycle Development* is never completely considered to clearly foresee all the actions due after market. Only a clear *traceability* in the development (from requirements to part numbers) of both the product elements and functions, seen as a direct result of the allocation of requirements, could allow covering the system lifecycle.

Models creation, digitalization, reusability and automatic documentation. A deep, complete and shared *documentation* is needed to make aware all the users and the operators about the details of the system operation, maintenance and even failure. This task is required to consolidate the "know-how" of the manufacturer, managing the turn-over of people, but even to support the user in service. Moreover, documentation and *data* should be recovered, collected and shared through a unique data management system, with dedicated repositories, controlled access to data bases under high security standards. This requires a preliminary digitalization of the information, then a network of connections, as well as an efficient data elaboration. *Models*, instead of pure documents, are easily reused in new products or versions of the same system, shared and stored as well as embedded in different documents to be automatically produced. These documents and models can be shared among operators of the same company when developing, but even together with suppliers and customers to enhance the activities of acquisition together with service and maintenance in operation.

Reducing costs of the system development, human mistakes, and late re-engineering activity. The SE is so widely growing up within the industrial world because of its effectiveness in handling some multidisciplinary engineering projects, by decomposing the complexity of systems, but even thanks to the so-called "left shift" in the resources consumption over the lifecycle.

As Fig. 2.1 shows, if one describes the resources usually spent for each step of the project to develop the system, conceptual stage covers a small percentage of the total cost, while tests, production and final assessment to disposal are fairly expensive. By converse, the SE tends to improve the efficiency of the early steps of development by increasing significantly the resources applied, by already considering for each element of the system the required test, production and assessment needs, to prevent the risk of a late detection of defects which might require a complete re-engineering, thus causing a demand of a huge amount of resources, being several times those already spent at that time, during the development.

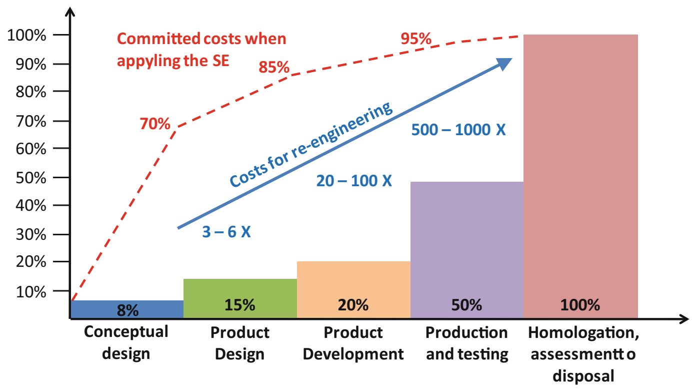

Fig. 2.1 Comparison between the usual distributions of the percentage of total costcovered for the system development and that proposed by the SE (Defense, 2001)

In this description a peak in costs occurs earlier in the project development and in time, but several costs usually foreseen for the late steps are anticipated. Therefore the peak is somehow shifted to the left side along the time axis, thus motivating the typical expression above mentioned. It can be remarked that a deeper conceptual activity might help in discovering problems before that any prototype is manufactured. Moreover, since the largest amount of money is spent in the early part of development, decisions are driven by a number of analyses, and, usually, lead to avoid a trial and error approach, typically more expensive and randomly effective.

Those needs actually are strictly related to some typical goals, as the design for safety and the quality in production, but they could be even associated to a bright *process of change configuration management* and to the two activities of *verificating the requirements* and *validating the system*, respectively.

In practice, the SE helps in *decomposing the complexity* of the problem in several issues and somehow in different *views*, thus exciting a careful activity of concept design and a complete elicitation of requirements. In addition, integrating all the models which might be used to describe and analyze the system in a same *virtual platform* improves their correlation and helps to create a holistic perception of the whole system. Models allow defining *a rationale* to be reused and assessed through different projects and driving the product designer through several tasks, thus making easier to account for all the required steps.

Some problems of *intrinsic safety* of product might arise in presence of a number of subsystems and components, if failure modes, causes and effects cannot be effectively predicted. Moreover, detecting clearly and fast any correlation between a failure occurring into a designed part and a requirement which motivated its inclusion within the overall system architecture is often rather difficult. Finally system prognostics, diagnostics and reliability can be effectively provided only through a clear *monitoring of events*. Therefore, the SE tools help in the safety analysis development, by showing how requirements, functions, components and parts are linked each other within the system architecture, thus making the *traceability* of requirements clear. Simultaneously, some issues of the so-called *functional modeling* allows defining actions, actors and relations, thus allowing to describe failure modes through a simple negation of those items. Designing an effective monitoring system to be applied to the product is never trivial and the availability of operational, functional and architectural analyses provided by the SE might enhance this skill as well as a more detailed numerical simulation.

It is important to remark that the SE somehow fills the gap between the overall "Project management", dealing with the organization and the development of the project, being interpreted as the set of activities leading to the realization of products, and the "Design" which is strictly, and almost only, related to the definition of the product elements through some dedicated and focused actions. It defines the *needs* to be satisfied, but even performs the technical *trade-off* among some candidate layouts, dealing with the whole system as well as with the smallest components, providing their *integration*, first through the design stage, then thanks to the manufacturing process and even after.

2.1.2 Four Pillars

As soon as the Reader will briefly consider the wide literature already available about the SE, probably an immediate feeling of shipwreck could arise because a number of books, guidelines, handbooks and also some Standards could be helpful, but very often they are focused on extremely different subjects. As usual in mechatronics, it might be remarked that SE involves many disciplines and each author is prone to explore the contents more familiar, focusing on a portion of the whole topic. Roughly speaking, it might be realized that four are the main areas of interest usually explored in the literature: *methodology*, *language*, *tools* and *data management*.

The methodology. Describing the methodology proposed by the SE looks often rather difficult without a bright roadmap as it was proposed by Estefan (2008). The methodology applied by the SE is based on *digital models*, easily shared within a community of users and easily stored by a data management system. This main property leads to define the SE methodology as a "model-based" approach or *Model-Based Systems Engineering* (MBSE). Two main models are developed, the first one drives the system engineer through the product development and concerns the *product life cycle*, while the second one describes both qualitatively (only by logic) and quantitatively (by numbers) the *product* itself, being the complex *system*.

Several reference models were proposed in the literature to define the *product life cycle development*. They are basically an ordered and rational list of interlinked steps to be performed in engineering the system. It might look as a quite obvious task of the whole design activity, but defining both the goals and the sequence of actions to be performed, as well as how they are mutually linked or even connected to the manufacturing process is crucial. Facing the problem of designing the system by either considering first the whole product and its requirements, then by decomposing it in components through a sort of *top-down approach*, or proceeding from the single component to the whole system through a *bottom-up approach*, was a matter of discussion. How the ALM and PLM actions could be suitably interrelated step by step, thus creating the traceability above mentioned was even deeply investigated.

As a relevant result, the MBSE currently provides a sort of checklist for the product developer and helps in handling some practical difficulties associated to the different levels of the system architecture and to its complexity. Particularly, on this assessment of the methodology two main issues were defined as the *process* itself, being focused upon the action to be performed in a due sequence through a logical path, and the *method*. Practically speaking the process defines *what* the system engineer must do and the method *how* to proceed.

The implementation of the process through the method includes the real product development whose object is the *system*. Therefore the modeling activity directly applies to it. Basically, the system functionality is first analyzed, through the *functional modeling*, to allow the allocation of requirements, then the performance of the system is predicted by the *physical modeling*. If the physical modeling is based on some mathematical description of the system behavior and its language coincides with mathematics, the functional modeling for long time did not have an effective mean of expression, like equations are for numerical models. Nowadays some languages support the modeling of functions, behavior and structural layout of the system, which can be represented before than it could be detailed, for instance into a technical drawing. Moreover, to create a functional model, some standard tools are required, which might be understood by all the users involved in the product development.

The tools. Actually two kinds of tools allow implementing the MBSE approach, consisting in some *theoretical tools*, like several typical *diagrams* and some *engineering methods*, and in the *software* providing a digital and virtual environment where the MBSE methodology can be implemented through some standard language.

The language. Dealing with a complex system inherently means often involving many technical competencies. To establish a fruitful cooperation, a common terminology, or better a tool which might be easily understood and applied by all the developers of a same task, is required. To reach this goal, a fairly intuitive language based on a standard set of symbols and items could be found. In case of the SE, some examples are available. A prevalent tradition comes from the *Unified Modeling Language (UML)*, which was then adapted and enriched for this purpose as the *System Modeling Language (SysML)*, but even other ones are currently developed to overcome some limitations of those languages, particularly when applied to the industrial product manufacturing.

The data management. The *information* and the *data storage* are both critical issues of the SE. They require a careful design of a common environment, including hardware and software, to which all the authorized users can access, allowing the tools being interoperated, i.e. data can be automatically transferred from one tool to another one, without a direct action of the user. This common environment, to be referred as *platform*, should allow producing, storing, sharing and elaborating all the required data. The platform must be compatible with the needs of different users and operators and with the software products used by all of them, to perform the modeling activity and to develop some related services. The platform needs a connection through the network, being based on a data bus, a cloud or other web services. The so-called *interoperability of software* tools and the *cyber security* are crucial issues of the platform building activity.

Those four pillars, as they were here above briefly defined, actually allow interpreting the contents of several contributions proposed in the literature and even some research activities currently performed within the MBSE. A clear rupture with the past approaches, based on documents to support the system development, consists in the model based approach, even fully digitalized nowadays. This is a distinctive characteristic of the MBSE and motivates its current wide application.

By converse the standardization of *methods*, *languages* and *tools* is still under assessment, although it looks a relevant goal for a complete diffusion and application of this approach. The Reader shall surely appreciate in this handbook how much *tools* and *methods* of the MBSE depend on the software currently available on the market, sold by several vendors. That's the actual challenging issue for a complete assessment of the SE. It is known that the different features exploited within the software products lead to a specific implementation of the methods and of the theoretical tools of the SE, according to some vendors' interpretation. A good standardization was already found in the system procurement, rather than in the product development, being the core in transportation and defense domains. In those cases, the product development is actually based on a complete definition of the *system capabilities*, described through a set of standards *views*, which are built up by resorting to the so-called *architecture frameworks*. These are another important component of the MBSE.

Next chapters will analyze more in details the above mentioned concepts to present the whole MBSE approach. Moreover, those key issues look useful for an easier interpretation of the existing literature. Some main references will be herein considered to start, although many others are available and published, whilst this short handbook is composed.

2.2 Some Historical Notes

The Systems Engineering, as is currently known by professionals and scientists, represents a convergence of several direct experiences performed by a number of system developers in several technical domains. Therefore the roots of the SE should be found definitely before than several theories and public references appeared since the latest 90s. For sure, the technical challenge of designing innovative and complex systems was typical of the fascinating story of the conquest of space. This was related to a fast growth of the space systems as well as of electronics, computer science and communications. Therefore, it does not surprise that a popular reference to learn the methodology of SE is the dedicated *NASA handbook* (NASA, 2007), but the real birth of the SE dates since the World War II. At that time clearly arose the need of managing some rather complex systems like aircrafts, tanks, but even the army logistics. The SE was evenly relevant for the operations research and decision analysis. During the 50s some preliminary references appeared and were applied to space programs and intercontinental ballistic weapons, thus promoting some methodologies to develop the systems (*Systems Engineering*) and to assure the full accomplishment of the goals of technical projects (*Project Management*), together with an effective prediction of risk (*Risk Analysis and Management*). Nevertheless, over the years, a bright necessity of relating those three main competences was satisfied by creating a suitable correlation through the SE.

After a first appearance of the *concept of system* in engineering within electric power distribution and telephones, the SE began to be conceived as a multidisciplinary approach in the early 50s of twentieth century. Basically, three main topics were related to the SE as the *general theory of systems*, the *cybernetics* and the *operations research* (Arrichiello, 2014).

In 1950 Ludwig von Bertalanffy identified in the "General System Theory" (Bertalanffy, 1972) the need of investigating biological and technical systems not only in terms of assembly of parts, but even for their mutual interactions as well as those with the environment outside their neighboroughs, introducing the new concept of *open system*. This immediately linked the *theory of systems* to the growing up fields of the *information technology* and of the *automatic control*.

In the same time, Norbert Wiener introduced the concept of *feedback*, seen as a possibility of adjusting the performance of systems by knowing their past experience, applying this idea to a new science that he called *cybernetics*, i.e. the control and communication in animals and machines (Wiener, 1948). An additional interesting issue of his theory was the introduction of the *system behavior* as a fundamental goal of the investigation, in all the fields of science and technology.

Operations research added a substantial contribution to this new discipline, during the World War II, as a mean to support *decisions*, then as a branch of mathematics and statistics, although it involved physicians, engineers and other scientists. That's why Weaver (1948) observed that when several members of diverse groups work together and create a unit, products are definitely greater than those obtained by a sum of parts and focused the attention upon the *system integration* and to the *holistic view* of product development.

The above mentioned origins found a favorable context in the Cold World War to grow up in projects like the *Intercontinental Ballistic Missile* (ICBM) and the *Semi-Automatic Ground Environment* (SAGE). The scale of those systems, the number of competences involved and the very challenging performances required made this context a perfect test rig for the SE tools and methods. Few years later the conquest of space and some programs like the *Apollo* missions finally brought to a formal definition of the SE process.

This growth of the tools of SE motivates some typical characteristics, which need today to be somehow updated and adapted to some other contexts like transports, health care, smart manufacturing and mechatronics. To catch immediately the substantial and powerful contents of the SE, it can be said that it smartly puts in evidence the *three basic ingredients* for an effective system design, described by the Royal Academy of Engineering in 1999 as follows: "every *design process* should begin from a *clear defined need*, should be performed through a *suitable vision* aimed to give an *effective response* to the need and the delivered product should meet the *expectations of the customer*, expressed by that need". As it will be herein explained, this simple process holds when the designer is aware about the *customer needs*, plays a deep attention to *requirements* and to a precise *validation of product* at the end of *process*. The SE helps the designer to consider the whole system, its neighboroughs, the *stakeholders* and all the *interactions* among them as well as among the system components *and interfaces*. In this purpose some additional benefits are provided.

The SE approach teaches to follow a structured rationale, to identify a hierarchy of items, to realize how the system behaves, to perceive the need of defining some suitable metrics, to *measure the performance* and to *model and simulate* that behavior, even through the merge of different approaches and tools might make the *simulation heterogeneous*. The literature, the tools and the software basically give this kind of information and help the user to follow the approach.

2.3 A Survey on the Literature About the Systems Engineering

A key starting point to understand the SE is a brief survey upon some popular textbooks, tools and standards, which nowadays constitute an appreciated reference for the different topics above described.

A first issue concerns the *approach* and the *processes* applied. As a matter of facts the most recent guideline about the SE including a fairly deep information is:

- David Walden, Garry Roedler, Kevin Forsberg, Douglas Hamelin, Thomas Shortell—*Systems Engineering Handbook of INCOSE*, 4th Ed., John Wiley and Sons, 2015

where a complete description of the *approach*, of its straight implementation through some standard *views* and *tools*, respectively, is provided, after a deep assessment performed by authors and contributors, over the years. It is significant even for a professional certification, through the INCOSE, being a worldwide recognized community for the systems engineers (see for instance the website www.incose.org and the activities promoted by the sections).

This reference obviously covers a wide range of topics related to the SE, although a relevant support for the *application to the aerospace engineering* is given by:

- The *NASA Systems Engineering Handbook*, NASA/SP-2007-6105 Rev.1, National Aeronautics and Space Administration, NASA, Headquarters Washington, D.C. 20546, December 2007

which focuses on the technical domain of aerospace and is witness of the relevant contribution given to the SE by this domain, since the edition:

- Robert Shishko—NASA-SP-6105 *The NASA Systems Engineering Handbook*, NASA, June 1995

where the fundamentals of the SE were defined and detailed. Some concepts as the *flow of activities* within the product development, the *elicitation of requirements* based on the system *goals, mission and operation scenarios* are clearly stated. It could be compared to:

- ESA-ESTEC (Requirements and Standards Division), *Space Engineering Technical Requirements Specification*, European Space Agency (ESA) Requirements and Standards Division Technical Report ECSS-E-ST-10-06C, Noordwijk, The Netherlands

A deeper information about the *process* and the *views* of the MBSE could be found in the popular reference:

- *Systems Engineering fundamentals*, the Defense Acquisition University press, Fort Belvoir, VA, USA, 22060-5565, January 2001

being focused on the system acquisition process of the Department of Defense (DoD) of the USA, thus representing the implementation of the SE within the military domain, under the constraints of dedicated technical and military standards. Very often the literature refer to the *DoD approach*, defined by the above cited document, free and available on the web. It is crucial resorting to this kind of support when the system development is strictly subordinated to a precise *commitment document for the acquirement*, as it happens in case of the Army, Navy and Air Force.

Those three references surely give a complete overview about the whole holistic approach, but several authors proposed some deepened descriptions of the related *methods* for an easier decomposition of the system and to face its inherent complexity. Among the others:

- David Oliver, Timothy Kelliher, James Keegan—*Engineering complex systems (with models and objects)*, McGraw Hill, New York, 1997
- Alexander Kossiakoff, William Sweet, Samuel Seymour, Steven Biemer—*Systems engineering: principles and practices*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003
- Charles Wasson—*System analysis, design and development: concepts, principles and practices*, Wiley, 2nd ed., July 2015

could be appreciated for introducing a clear classification of *goals, missions* and *scenarios* to deal with the *requirements* definition, the *operational and functional and analyses*, then to the *design synthesis*, through a bright path of activities and a driven decomposition of the system elements. This approach applies even to other technical domains like the health care, some industrial and social applications, including automotive, railways systems, smart cities and manufacturing.

It is worthy noticing that implementation in some cases is never so straight because of a lack of experience or an imperfect management of this approach. This difficulty is mainly found in the concept design and in managing the *integration* of system in production, therefore it motivates an extended analysis of the SE *management* as it was performed in:

- Andrew Sage, William Rouse—*Handbook of Systems Engineering and Management*, John Wiley and Sons Inc., 1996 and following editions
- Benjamin Blanchard—*System Engineering management*, Wiley, New York, 2004

A more integrated overview on the SE is provided by some other sources, either by groups deeply involved in the SE education, application or service supplying or even vendors of products for the SE. Among the most known there are:

- Richard Adcock (ed)—*Guide to the Systems Engineering Body of Knowledge* (SEBoK), The Trustees of the Stevens Institute of Technology, Hoboken, NJ, USA, vs.1.3, 2014 (available at www.sebokwiki.org)
- Some references shared for free through the web by the *Vitech Corporation* (www.vitechcorp.com)

A growing interest about the SE was shown by some important Society, like the *American Society of Mechanical Engineering* (the *ASME International*), since the appearance in the catalogue of publications of:

- Thomas Van Hardeveld, David Kiang—*Practical Application of Dependability Engineering: An Effective Approach to Managing Dependability in Technological and Evolving Systems*, The ASME Int., New York, 2015

where a detailed analysis of *dependability* and *traceability* of requirements within the system design is proposed, by introducing several concepts about the product lifecycle development, applied to the mechanical engineering.

Since a longer time the *Institute of Electrical and Electronics Engineers* (*IEEE*) supports the information and the education within the SE through several contributions, mainly based on some journals like:

- the *IEEE Systems Journal* of the IEEE System Council

However a specific focus on the SE is kept by the *International Council on Systems Engineering* (*INCOSE*), through several means, even in connection with other institutions or some firms, basically through the web, several publications and conferences.

This activity already led to the introduction of a newer interpretation of the SE as it was proposed more recently in terms of *Lean Systems Engineering* in:

- Bohdan Oppenheim—*Lean for Systems Engineering with Lean enablers for Systems Engineering*, Wiley, 2011
- Josef Oehmen (Ed.)—*The Guide to Lean Enablers for Managing Engineering Programs*, Joint MIT—PMI—INCOSE Community of Practice on Lean in Program Management, Cambridge, MA, USA, 2012 (http://hdl.handle.net/1721.1/70495)

Those references basically show a link between the strategic approach of the *Lean manufacturing* (King, 2009), enabling a careful reduction of waste and cost in production, and the systematic approach to the product development of the SE, with the aim of combining the effectiveness of the SE approach to the sustainability of the lean process.

All the above cited sources do not complete the wide list of references currently available, but it happens that in daily practice of SE some of those are mentioned for their well known contribution.

A straight implementation cannot be uncoupled from the tools and the test cases. Aside the above literature another set of references specifically deals with the *languages* of the Model Based Systems Engineering.

A unified approach among several technical domains is currently under development by resorting to the *System Modeling Language* (*SysML*), evolution and adaption to the SE needs of the former *Unified Modeling Language* (*UML*). Many authors provide a complete overview about the language and its rationale use within SE, although some are more frequently cited as:

- Sanford Friedenthal, Alan Moore, Rick Steiner—*A practical guide to SysML, the System Modeling Language*, The MK/OMG Press, 1999 (and following)
- The *OMG SysML*, (version 1.1 November 2008 and following)
- Tim Weilkiens—*Systems Engineering with SysML/UML—Modeling, Analysis, Design*, The MK/OMG Press, 2008
- Lenny Delligatti—*SysML Distilled: A Brief Guide to the Systems Modeling Language*, Addison Wesley, 2014
- Pascal Roques—*Modélisation de systèmes complexes avec SysML*, Eyrolles, 2013 (*in French*)

It might be considered that new evolutions are already foreseen, as the Reader could realize in:

- Lifecycle Modeling Language (LML) Specification, (http://www.lifecyclemodeling.org/spec/)

although a first convergence of methods and standard processes is still currently looked for, by a direct implementation of the SysML or at least some enriched version.

More than handbooks about the SysML, it seems crucial remarking the existence of some other sources oriented to its practical use through the methods of the MBSE. Surely very well known is:

- Hans–Peter Hoffmann—*Systems Engineering best practices with the Rational solution for systems and software engineering, Deskbook* (Model Based Systems Engineering with Rational Rhapsody and Rational Harmony for Systems Engineering), the IBM Software Group, © IBM Corporation, 2011

which enunciates the proprietary *Harmony© approach* proposed and implemented within the software tool *IBM Rational Rhapsody®* by IBM. That approach is still matter of testing for several technical domains, where contents are applied to introduce the SE itself. Because of this need, some complete examples of implementation are currently widely appreciated as the:

- Bruce Powel Douglass—*AGILE Systems Engineering*, MK Morgan Kaufmann, Waltham, MA, USA, 2016

where a bright development of a real test case is shown with all the details of modeling activity

Many other sources are already available in this field and listing all precisely should be unpractical, or even impossible. Those were proposed actually to outline some typical contributions, the related subjects covered by the literature and even because some specific content has been matter of a refinement of the SE methodology, or it helped in assessing some critical issue, as soon as the SE was applied to the series and material products of industry.

Mentioning those references was aimed at showing to the Reader some different topics of education about the SE and even to point out a number of views, which enriched the approach, by making sometimes difficult a fast and complete convergence towards a generalized and comprehensive *standardization*, as is still currently looked for.

To complete this overview it might be mentioned that a specific need of references is still perceived within the definition of *metrics* for the evaluation of artifacts and products during the *verification* and *validation* activities, as it shall be soon explained in this handbook. A preliminary answer to this necessity was given by a dedicated series of textbooks, published by some universities, to support the classes, as, for instance at MIT (Massachusetts Institute of Technology):

- Steven Eppinger, Tyson Browning—*Design structure matrix methods and application*, Engineering System, MIT Press, 2013

which highlights, as a current trend of the literature, to move from some main references dealing with the concept of system and the general approach of the SE towards some detailed topic of its implementation, to assess and refine the *tools* previously proposed.

From the above mentioned citations it looks clear that:

- several technical domains are currently involved within the development of methods and tools related to the SE and express different approaches and needs;
- technicalities are related to some issues like the guidelines for the implementation, the languages and their assessment, the software tools and the creation of suitable data management systems;
- many other activities like the *safety engineering*, the *risk management*, the *maintenance of systems* are connected to the SE and their tools and methods must somehow meet those more typical of the MBSE.

2.4 Technical Standards on the Systems Engineering

More details about the SE processes are today already provided by several technical standards, being aimed at driving the user in implementing the SE and its tools. According to the brief sketch above offered about the history of the SE, it could be considered that a preliminary description of systems was provided by some *military standards* as the *DoD-Mil-Std 499* (1969), the *Mil-Std 499A* (1974) and the *Army Field Manual 770-78* (1979). They led to a preliminary draft of the *Mil-Std 499B* (1994), which actually was never released as itself, but it was enabler of the development of the *ANSI/EIA 632* “*Processes for Engineering a System*”, which collects the recommendations of the *American National Standards Institute* (*ANSI*) and the *Electronic Industries Association* (EIA).

The management of the SE was then goal of the *IEEE 1220-1998* "*Standard for Application and Management of the Systems Engineering Process*", which was presented in 1998 and refined in 1999, respectively.

During the early 2000s the real references for the SE were released as the *ISO/IEC 15288:2002* "*Systems Engineering—System Life Cycle Processes*" (2002), as a result of the activity of the *International Electrotechnical Commission* (*IEC*) together with the *International Organization for Standardization* (*ISO*), whilst the guidelines of the INCOSE Handbook (recent version 2015) and of the NASA handbook (recent version 2007) were updated.

Other standards were even published as the *ISO/IEC 19760* "*Guide for ISO/IEC 15288—System Life Cycle Processes*", which drives to a straight application of the ISO/IEC 15288. The *Institute of Electrical and Electronics Engineers* (*IEEE*) basically accepted and applied the relevant contents of the ISO/IEC 15288 in the *IEEE Std 15288 2004*. Moreover, the NASA analyzed several issues of those standards within the *NASA NPR 7123.1A* "*Systems Engineering Processes and Requirements*".

As the Mil-Std 499, 499A and 499B basically describe the life cycle approach, while some details on the process, the goals, the outcomes and the activities were described by the ANSI/EIA 632 as well as by the ISO/IEC 15288 and the IEEE 1220, some differences could be detected among the three main standards. The ANSI/EIA 632 defines the *processes* required to engineering or re-engineering the system, the ISO/ICE/IEE 15288 provides a *framework* to define the system life-cycle, while the IEEE 1220 focuses on the system *management*.

A key contribution to this topic was even provided by the:

- ISO/IEC/IEEE 16085 (2006), "*Systems and Software Engineering Risk Management*"
- ISO/IEC/IEEE 15939 (2007), "*Systems and Software Engineering Measurement Process*"
- ISO/IEC/IEEE 16326 (2009), "*Systems and Software Engineering Project Management*"
- ISO/IEC/IEEE 24765:2009 (2009), "*Systems and Software Engineering Vocabulary*"
- ISO/IEC/IEEE 150261 (2009), "*Systems and Software Engineering System and Software Assurance, Part 1: Concepts and definitions*"
- ISO/IEC/IEEE 150262 (2010), "*Systems and Software Engineering System and Software Assurance, Part 2: Assurance case*"
- ISO/IEC/IEEE 42010 (2011), "*Systems and software engineering Architecture description*"
- ISO/IEC/IEEE 15289 (2011), "*Systems and Software Engineering Content of LifeCycle Information Products (documentation)*"
- ISO/IEC/IEEE 29148 (2011), "*Systems and software engineering Requirements Engineering*"
- ISO/IEC/IEEE 150263 (2011), "*Systems and Software Engineering System and Software Assurance, Part 3: Integrity Levels*"

Those standards clearly demonstrate how much the SE could benefit of the experience within the *software engineering*, although it applies to several other systems, like to aerospace and mechatronics. This link to the software engineering can be better appreciated if some technical standards of that field are analyzed.

In addition to the above mentioned ISO/IEC/15288, recently updated (2015) about the Systems Engineering, the:

- ISO/IEC/12207:2008 (2008) deals with the "*Systems and software engineering—software life cycle processes*", and applies to the software development, seen as a product, and to the processes proposed for a more general implementation to the industrial systems
- ISO/IEC/26262 (2011–2012) defines the *functional safety for automotive equipment throughout the life cycle of automotive electronic and electrical safety related systems* and looks like an adaption of the following one:
- IEC 61508 (2012) even applied to the automotive electric and electronic devices, being entitled "*Functional safety of electrical/electronic and programmable electronic safety-related systems (E/E/PES)*".

Some additional references strictly related to the SE were released directly by the ISO as the:

- ISO 17666:2003 (2003), "*Space Systems Risk Management*"
- ISO 31000:2009 (2009), "*Risk Management Principles and Guidelines*"
- ISO/IEC 31010:2009 (2009), "*Risk Management and Assessment Techniques*"

It is worthy noticing that in all of the above mentioned standards, quite independently on the specific goal of each one, some clear attributes of the SE approach are stated. The approach is *solution-oriented*, operates on the base of well defined *needs* and proceeds through a *holistic view* of the system, including operations, environments, stakeholders, and the whole *life cycle*, as it was brightly confirmed by the *SeBok Handbook* (2014). Among all, the ISO/IEC/IEEE 15288 represents the result of a gradual evolution of those standards. The process of SE there identified follows a rationale which starts from the exploration of needs, investigates functions and behaviors, proposes and analyses some suitable architectures to reach a final synthesis of design.

2.5 Software Tools for the Systems Engineering

The implementation of the SE requires some *software tools* to collect the requirements and to proceed with the modeling activity proposed by this approach. Among the existing software some are quite well known and to make aware the Reader will be herein briefly cited.

The IBM Rational DOORS® is currently widely used for the requirements elicitation. It is usually integrated with the IBM Rational Rhapsody® which allows implementing the typical tools of the SysML language. However, other vendors already developed similar packages like Agilian® (Visual Paradigm), Artisan Studio® (Atego) nowadays embedded into the PTC Integrity Modeler®, Enterprise Architect® (Sparx Systems), Cameo Systems Modeler® (No Magic) and UModel® (Altova). They are available on market in different versions and usually sold by the vendors above cited within brackets. In addition, are distributed for free the Modelio® (Modeliosoft) and Papyrus® (Atos Origin).

Selection of software tools is quite a critical issue of the SE platform definition. The above mentioned tools offer different contents, in terms of options, modules and features, but a main issue is the format for exchanging the data and their compatibility for a full *interoperability* with several other tools currently used within engineering. This topic will be herein deeply described, since it looks the main bottleneck for a very fast and easy implementation of the SE in all the technical domains and for every application, although the problems occurring in some case might be overcome by resorting to some standard connectors like the *Functional* Mock-up Interface (FMI) or by basing the creation of the *tool chain* on some interoperability standard like the *Open Services for Lifecycle Collaboration* (OSLC).

References

Adcock, R. (2014). *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (Bd. 1.3). Hoboken, NJ, USA: The Trustees of the Stevens Institute of Technology.

ANSI/EIA 632 "*Processes for Engineering a System*".

Army Field Manual 770-78 (1979).

Arrichiello, V. (2014). Introduction to the systems engineering. In *INCOSE Italian Chapter Conference on Systems Engineering (CIISE 2014)*. Rome, Italy.

Bertalanffy, Lv. (1972). General systems theory. *The Academy of Management Journal, 15*(4), 407–426.

Blanchard, B. (2004). *System engineering management*. New York: Wiley.

Defense, A. U. (2001). *systems engineering fundamentals*. Fort Belvoir, VA, USA: Defense Acquisition University.

Delligatti, L. (2014). *SysML distilled: A brief guide to the systems modeling language.* Addison Wesley.

Division, E.-E. R. (2017). *Space engineering technical requirements specification, technical report ECSS-E-ST-10-06C.* Noordwijk, The Netherlands: European Space Agency (ESA) Requirements & Standards Division.

DoD-Mil-Std 499 (1969).

Douglass, B. P. (2016). *AGILE systems engineering*. Waltham, MA, USA: MK Morgan Kauf-mann.

Eppinger, S., & Browning, T. (2013). *Design structure matrix methods and application, engineering system.* MIT Press.

Estefan, F. A. (2008). *Survey of the model-based systems engineering (MBSE) candidate methodologies.* The INCOSE MBSE Initiative.

Friedenthal, S., Moore, A., & Steiner, R. (1999). *A practical guide to SysML, the system modeling language.* The MK/OMG Press.

Hoffmann, H.-P. (2011). *Systems engineering best practices with the rational solution for systems and software engineering, deskbook (model based systems engineering with rational rhapsody and rational harmony for systems engineering).* The IBM Software Group.
IEEE 1220-1998 "*Standard for Application and Management of the Systems Engineering Process*".
ISO/IEC 15288:2002 "*Systems engineering—System life cycle processes*".
ISO/IEC 19760. (2002). "*Guide for ISO/IEC 15288—System life cycle processes*".
King, P. L. (2009). *Lean for the process industries.* Taylor and Francis.
Kossiakoff, A., Sweet, W., Seymour, S., & Biemer, S. (2003). *Systems engineering: Principles and practices*. Hoboken, NJ: Wiley.
Lifecycle Modeling Language (LML) Specification. (2017). http://www.lifecyclemodeling.org/spec/.
Mil-Std 499A (1974).
Mil-Std 499B (1994).
NASA. (2007). *Systems engineering handbook* (Bd. rev.1 and following versions). NASA/SP-2007-6105.
Oehmen, J. (2012). *The guide to lean enablers for managing engineering practice on lean in program management.* Cambridge, MA, USA: MIT—PMI—INCOSE Community of Programs.
Oliver, D., Kelliher, T., & Keegan, J. (1997). *Engineering complex systems (with models and objects)*. New York: McGraw Hill.
Oppenheim, B. (2011). *Lean for systems engineering with lean enablers for systems engineering.* Wiley.
Roques, P. (2013). *Modélisation de systèmes complexes avec SysML.* Ey-rolles.
Sage, A., & Rouse, W. (1996). *Handbook of systems engineering and management.* Wiley.
SeBok Handbook. (2014). www.sebokwiki.org.
Shishko, R. (1995). *The NASA systems engineering hand-book.* NASA-SP-6105.
Van Hardeveld, T., & Kiang, D. (2015). *Practical application of depend-ability engineering: An effective approach to managing dependability in technological and evolving systems*. New York: The ASME Int.
Vitech Corporation. (2017). von www.vitechcorp.com.
Walden, G. K. (2015). *Systems engineering handbook of INCOSE.* Wiley.
Wasson, C. (2015). *System analysis, design and development: Concepts, principles and practices* (2nd ed.). Wiley.
Weaver, W. (1948). *Science and complexity, American scientist, 36*, 536. New York City: Rockefeller Foundation.
Weilkiens, T. (2008). *Systems engineering with SysML/UML—Modeling, analysis, design.* The MK/OMG Press.
Wiener, N. (1948). *Cybernetics: Or control and communication in the animal and the machine*. Boston, MA: Technology Press.

http://www.springer.com/978-3-319-71836-1

Systems Engineering and Its Application to Industrial Product Development
Brusa, E.; Calà, A.; Ferretto, D.
2018, XXII, 353 p. 217 illus., 197 illus. in color.,
Hardcover
ISBN: 978-3-319-71836-1