Une approche basée sur les processus et dirigée par les compétences pour l'éducation en ingenierie des systèmes

Mohammed Bougaa

NNT : 2017SACLC050

Thèse de Doctorat
de l'Université Paris-Saclay
préparée à CentraleSupélec

Ecole Doctorale n°573(*Interfaces*)
Approches Interdisciplinaires: Fondements, Applications et Innovation
Spécialité du Doctorat: Informatique

M. Mohammed BOUGAA

Une Approche Basée sur les Processus et Dirigée par les Compétences pour l'Education en Ingénierie des Systèmes

Soutenue à " Cergy-Pontoise ", Le 29 Septembre 2017.
Composition du Jury :

M.	Claude LAPORTE	Rapporteur (Pr)	Ecole de Technologie Supérieure Montréal, Québec, Canada
M.	Eric LEVRAT	Rapporteur (Pr)	Université de Lorraine Nancy, France
Mme.	Marija JANKOVIC	Examinatrice (Pr) Présidente du Jury	CentraleSupélec Châtenay-Malabry, France
Mme.	Odile MORNAS	Examinatrice (Ing)	Thales university Jouy-en-Josas, France
M.	Rory V.O'CONNOR	Examinateur (Pr)	School of Computing Dublin City University, Ireland
M.	Alain RIVIERE	Directeur de Thèse (Pr)	Institut supérieur de mécanique de Paris, Saint-Ouen, France
M.	Hubert KADIMA	Co-Directeur de Thèse (Dr)	Ecole Int des Sciences et du Traitement de l'Information, Cergy, France
M.	Stefan BORNHOFEN	Encadrant de Thèse (Dr)	Ecole Int des Sciences et du Traitement de l'Information, Cergy, France

NNT : 2017SACLC050

PhD Thesis
of Paris-Saclay University
Prepared At CentraleSupélec

Doctoral school n°573($Interfaces$)
Interdisciplinary Approaches: Foundations, Applications and Innovation
PhD Specialty: Computer Science

M. Mohammed BOUGAA

A Process-Centered and Competency-Driven Approach for Systems Engineering Education

Defended at " Cergy-Pontoise ", On: September 29th, 2017.
Thesis jury :

M.	Claude LAPORTE	Reviewer (Pr)	Ecole de Technologie Supérieure Montréal, Québec, Canada
M.	Eric LEVRAT	Reviewer (Pr)	Université de Lorraine Nancy, France
Mme.	Marija JANKOVIC	Examiner (Pr) Jury President	CentraleSupélec Châtenay-Malabry, France
Mme.	Odile MORNAS	Examiner (Eng)	Thales university Jouy-en-Josas, France
M.	Rory V.O'CONNOR	Examiner (Pr)	School of Computing Dublin City University, Ireland
M.	Alain RIVIERE	Thesis Director (Pr)	Institut supérieur de mécanique de Paris, Saint-Ouen, France
M.	Hubert KADIMA	Thesis Co-Director (Dr)	Ecole Int des Sciences et du Traitement de l'Information, Cergy, France
M.	Stefan BORNHOFEN	Thesis Supervisor (Dr)	Ecole Int des Sciences et du Traitement de l'Information, Cergy, France

Acknowledgment

الْحَمْدُ لِلَّهِ الَّذِي بِنِعْمَتِهِ تَتِمُّ الصَّالِحَاتُ

Firstly, I would like to express my gratitude to my thesis director Prof. Alain RIVIERE for the supervision of this work, for his patience, and orientation, also for providing me with the necessary resources. His guidance helped me all the time of research and writing of this thesis. I thank him especially for trusting me and giving me all the necessary liberty and autonomy while being here to validate my research directions and decisions.

Besides my thesis director, I would like to thank my advisor Dr. Stefan BORNHOFEN, for being always here to support me and for the numbered discussions that led to the completion of this work. I thank him for pushing me to ask the right questions, to be sure I'm on the right paths, I also thank him for all his proofreading of my papers and this thesis. My sincere thanks also goes to Prof Samir. LAMOURI and Prof. Samir GARBAYA, who gave me opportunities to exchange with them around this work, their insightful comments and encouragement were very helpful. I also thank Mr. Claude LAPORTE and Mr. Eric LEVRAT for accepting to review this report, and Mrs Marija JANKOVIC, Mrs Odile MORNAS and Mr. Rory V.O'Connor for accepting to be part of the jury of my thesis defense.

My sincere thanks also goes to Faida MHENNI,Moncef HAMMADI, Jean Yves CHOLEY, Alexis FRANCOIS, Antoine LANTHONY, Antoine BRUNNER, and all Supmeca staff, for their help and assistance during this thesis, and especially for facilitating the platform experimentation, without forgetting to thank all my friends and colleagues at EISTI. I also thank all the staff of Interface doctoral school at CentraleSupelec and especially Mrs Suzanne Thuron, who had always been here to respond to my inquiries and questions, I thank her for her professionalism and sense of service.

Last but not the least, I would like to thank my family: my parents, my wife, and to my brothers and sister for supporting me spiritually throughout the three years of this thesis and my life in general. I would like to thank my parents for their love, encouragement, and support in all my pursuits, and for their faithful support during the final stages of this Ph.D. And My Sister and Brothers, for their support. Words cannot express how grateful I am to my supportive, encouraging, and patient wife Fatma, in addition to her thesis preparation, and spending her nights taking care of our baby, she always found the time to review my papers, to discuss about the direction i took, ...etc., sometimes, she even appeared understanding my work better than me :).

I dedicate this thesis to my adorable daughter, "Nour Filastine", my treasure, who came into the world one year ago to embellish us with happiness and challenges.

Abstract

Dealing with today's complex systems requires highly qualified systems engineers. However, in order to be effective as systems engineers, engineering students need practical and real world experiences with such complex systems, in addition to the necessary knowledge in their traditional engineering discipline. In fact, Systems Engineering (SE) education is as much about skills and processes, as it is about knowledge transfer. This makes teaching SE a very challenging task where the complexity of the engineered systems has to be considered together with SE complexity itself. Significant efforts have been made the last two decades in order to deal with this complexity. Attention has been paid to enhance both SE practices and SE education practices. These efforts are provided by organizations using SE, as well as by academic institutions dealing with engineering and SE education. They led to the production of standards, standards documentation, processes assessment models, competency models, competencies assessment resources and more, while improving SE practices within industry and organizations. By organizations we address all the different fields, other than industry, where systems engineering is applied, such as government departments and agencies...etc. In addition, many practical attempts addressed SE education at undergraduate, graduate, post-graduate, and professional levels.

This thesis surveys the current practices and advancements in SE education, and concludes that there is no common conventional learning path for this discipline (if we consider it as such). One of our most important finding is that there is no conventional education solution, among academia and industrials/organizations, capable of responding to different industry and organization's challenges, for seeking well-trained systems engineers, or specialized engineers with the minimum viable knowledge about the fundamental principles of SE. A solution that can produce systems engineers with different depth levels of expertise, and that should be easily transmissible from a university to another, and from a specific engineering discipline to another. We tried to consider some basic questions such as: what pedagogical model should be used ? what role technology and educators should play in a perfect SE educational environment? which tools should be used?

We ended up proposing a novel solution for SE education (an approach with its supporting web-based platform). The proposed approach is based on the recommendations of academic and industrial communities. It is centered around the use of SE standardized processes and at the same time very adaptive, with learning scenarios that can be driven by the acquired or to-be-taught SE competencies. The proposed solution is a web based platform that has been developed to support this novel approach within a distant Project Based Learning (PBL) environment. Finally, we put this solution to the test, firstly by questioning a sample of undergraduate students and their educators about this approach and its platform features, and then through another sample of doctoral students using it in the context of an introductory course to systems engineering, within a Project Based Learning (PBL) course. Students using this solution will be able to not only engineer the requested system in a distant and collaborative way, but also to engineer it the right way. The solution aims to ease the learning at the same time of fundamental principles and processes of systems engineering, along with communication, team management, collaboration, and related soft skills. On the other hand, educators will be able to better manage their learning scenarios, training resources, and the expected outcomes. Last, educators and students' organizations using this solution will be able to manage and normalize the competencies to be acquired by their future systems engineers at every level.

Keywords Systems Engineering, Systems Engineering Education, Systems Engineering Standards, Project-Based Learning, Lifecycle Model, Competency Models.

Résumé

Faire face aux systèmes complexes d'aujourd'hui nécessite des ingénieurs systèmes hautement qualifiés. Mais pour être efficace en tant qu'ingénieur système, les étudiants en différentes disciplines d'ingénieries traditionnelles ont besoin d'une expérience pratique et réelle dans ce type de systèmes complexes, en plus des connaissances théoriques nécessaires dans leur discipline.

L'éducation dans le domaine de l'ingénierie des systèmes (IS) concerne aussi bien les compétences et les processus, que les faits transférables et les connaissances. Cela rend l'education en IS un sujet très difficile, avec un besoin de considérer à la fois la complexité des systèmes à concevoir, ainsi que la complexité et l'abstraction de l'IS dans certains de ses principes fondamentaux. Des efforts considérables ont été accomplis afin d'améliorer les pratiques de cette discipline et pour faire face à la complexité des systèmes actuels et à l'éducation en IS. Ces efforts sont fournis par les organisations utilisant l'IS et les établissements universitaires et collèges formant des ingénieurs. Ils ont conduit à la production de normes, de documentations pour ces normes, des modèles d'évaluation des processus, ainsi que des modèles de compétences, tout en améliorant le déploiement et la pratique de cette discipline dans l'industrie.

Au cours de cette thèse, nous avons examiné les pratiques actuelles en matière d'éducation en IS, et nous avons conclu qu'il n'existe pas de solution d'apprentissage formelle, unique, et adoptée par la communauté pour cette discipline. Une solution capable de répondre aux défis de l'industrie, et de produire des ingénieurs systèmes bien formés ou des ingénieurs spécialisés ayant des connaissances minimales viables sur les principes fondamenteaux de l'IS. Une solution qui devrait être facilement transférable d'une université à une autre, et d'une discipline d'ingénierie à une autre, et qui permet de former des ingénieurs système de différents niveaux d'expertise en largeur et/ou en profondeur sur les principes de l'IS. Nous avons ensuite essayé de prendre en compte quelques aspects de base du domaine de l'éducation, comme le modèle pédagogique qui devrait ètre utilisé et le rôle que la technologie et les formateurs devraient jouer dans un environnement idéal d'éducation en IS. Nous avons fini par proposer une approche novatrice pour former les étudiants sur les principes fondamenteaux de l'IS, indépendamment de leur discipline technique. Nous avons proposé une approche centrée sur l'utilisation des processus normalisés en IS, tout en faisant en sorte que le scénario d'apprentissage soit très adaptable et qu'il puisse être piloté par les compétences d'IS acquises ou à acquérir. Ensuite, une plateforme Web a été développée pour soutenir cette nouvelle approche d'apprentissage. Enfin, un échantillon d'étudiants de premier cycle et de leurs formateurs ont été interrogé sur l'utilité et l'efficacité de cette solution, et un autre échantillon de doctorants l'avait expérimenté dans un cours d'initiation à l'IS, dans le cadre d'un apprentissage par projet (APP).

En utilisant cette approche, les étudiants ne seront pas seulement en mesure de bien concevoir le système demandé de manière distante et collaborative, mais ils seront aussi capables de l'élaborer de manière appropriée. Cela leur permettra d'apprendre les principes et processus fondamentaux de l'IS, à mieux communiquer dans un environement de travail, la gestion d'équipe, la collaboration et les compétences techniques connexes. Les formateurs d'un autre côté pourront mieux gérer leur parcours d'apprentissage, les ressources pédagogiques, et les résultats escomptés. En utilisant cette solution, les organisations de ces formateurs et étudiants, c'est-à-dire les universités et collèges, pourront gérer et normaliser les compétences acquises par leurs futurs ingénieurs systèmes à tous les niveaux.

Mots-clefs Ingénierie des Systèmes (IS), Education de l'IS, Normes de l'IS, Apprentissage Par Projet, Modèle de Cycle de Vie, Modèles de Compétences.

Contents

List of Figures

List of Tables

1
General introduction

1.1 Context

To ensure their survival, basic needs, and to shape a better future, humans have always been impacting, and were impacted by their surrounding systems at different levels and different degrees of involvement. Humans learned from the early ages how to observe the solar system and used it for counting, how to be a part of the plant cycle to ensure food and survival, and how to design and master other man-made systems, for hunting, preserving water, and protecting themselves. But when producing new systems, they also felt the need for good ways about how to transfer the acquired knowledge from a generation to another, both regarding the usability of their systems, in addition to best practices, moral and environmental considerations, and relationships with their fellow citizens, and with their interests. Some of these systems died out, others remained over time, while the most important and most useful ones evolved within their environment. A good example is pottery [1] industry which is still improving and developing, although the first potter's wheel was invented more than 6000 years BC, new pottery systems are continuously being conceived, or existing ones are continuously being improved and enhanced, depending on which region of the world and which environment we are talking about.

The evolution of man-made systems, as well as the permanent will of better understanding and using surrounding systems with their relationships and interactions, is not without consequences. The most important point resulting from this constant evolution is Complexity: complexity of systems in their nature and behavior, complexity of systems regarding their design, and complexity of systems regarding their maintenance and use in good conditions.

During the last few decades, the complexity of man-made systems has increased to an unprecedented level. This complexity is caused essentially by:

- The increasing number of stakeholders being part and/or having an interest in nowadays systems, throughout their entire lifecycle.
- The increasing number of stakeholders' disciplines involved in a system, along with the geo-distributed context of todays' engineering makes continuous collaboration difficult during the system lifecycle.
- The increasing number of elements forming one complex system in order to deliver the intended service satisfying the stakeholders needs and requirements.

[1] Pottery is the craft of making ceramic material into pots or potterywares using mud

- The complexity of existing interactions, more interestingly, the collaboration within the elements forming a complex system, and between different independent systems to deliver the expected service.

- The wide range of available technologies, materials, tools, and methodologies to conceive, produce and retire a system. In addition, the rapid growth of these resources means that systems need to be adapted and engineered/re-engineered very quickly, respecting more restrictive schedules, budgets and with more requirements to be satisfied.

- The difficulty to manage and benefit from the huge amount of knowledge being generated within nowadays complex systems engineering activities and operations.

"*Perhaps for the first time in history, humankind has the capacity to create far more information than anyone can absorb, to foster far greater interdependency than anyone can manage, and to accelerate change far faster than anyone's ability to keep pace. Certainly the scale of complexity is without precedent.*" [7]

Conceiving, designing, producing, using, and retiring these kind of complex systems, with better management of their requirements and their stakeholders' requirements, in other words "engineering" these systems, has led to the emergence of new approaches for managing the complexity generated through all the mentioned stages. One of these approaches is what we call Systems Engineering (SE). Systems Engineering is a structured approach focusing on the design and the management of complex engineering projects over their entire lifecycle. It is presented by the International Council on Systems Engineering (INCOSE), as *"an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functions early in the development cycle, documenting requirements, proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing and disposal."* [2].

Significant efforts have been made to address the adoption of systems engineering and its challenges as a new way of engineering systems. These efforts are primarily the development of multiple international standards, guides and methodologies, to help the industry and organizations adopting this approach. They led to the appearance of a harmonized view of all involved processes required throughout a system's lifecycle to transform customer needs into a system solution [8]. This harmonization took the form of several standards that have been defined, from the early US Military Standard MIL-STD-499 dating back to 1969, to the last revision of the ISO/IEC/IEEE 15288 published on May 15th, 2015 [4]. Thanks to the creation of such standards, and if well adopted, companies dealing with engineering complex systems may better cope with the problems that occur within the stages of a system lifecycle.

Because of this change in the engineering of systems and especially the fact that nowadays companies are dealing with very complex and multidisciplinary systems that need much more than technical excellence, the time where companies were seeking people based on a mere excellent background in their technical specialization or discipline is revoked. They actually need engineers who are not only excellent in their discipline, but also and more importantly, who are good team workers, adept communicators, and lifelong learners [9]. In addition, they seek people who are able to manage complex systems in order to produce clients expected outcomes, while satisfying different project stakeholders, and ensuring an optimization of time, cost, energy and different resources throughout the entire lifecycle of a project. In other words, these companies are seeking well-trained systems engineers, with high systems thinking capabilities, who master the fundamental and domain-specific Systems Engineering principles and their corresponding processes, tools, and methods.

This need for systems engineering and for well trained systems engineers is demonstrated by Wasson, in his paper [10], when he stated that: " *Unfortunately, the engineering of systems, performed in many organizations is often characterized as chaotic, ineffective, and inefficient. Objective evidence of these characteristics is exemplified by noncompliance to requirements, cost overruns, and late schedule deliveries in program metrics for a project's contract or task triple performance constraints – i.e., technical, cost, and schedule.*". Based on his experience, the author suggests that *"many engineers are estimated to spend on average from 50% to 75% of their total career hours collaborating with others concerning the engineering of systems – i.e., systems engineering - for which they have no formal education"*. The author also stated that, universities and colleges are proposing systems engineering courses or capstone projects based on systems engineering principles and practices, to meet program accreditation requirements and industrial needs. However, the outcomes of these programs are more likely to focus on domain-based design and engineering, with not significant amount of systems engineering concepts, principles, and practices required to effectively and efficiently perform systems development. In fact, engineering students need practical and real world experience in addition to the necessary knowledge in their traditional engineering discipline, in order to be effective as systems engineer, [11]. Also, engineering students, willing to be effective systems engineers, need to be taught on systems engineering in real-life like environments, they need to learn within a multidisciplinary and geo-distributed context, while using reality like processes and tools [12].

In addition to the efforts from the industries and organizations dealing with systems engineering and standardization institutions, to harmonize the practice of systems engineering and to help its deployment, another effort is made by universities and colleges, in order to train and produce more qualified systems engineers, capable of meeting the previous challenges. They are mostly interested in investigating new paths to teach systems engineering, by *defining which competencies* best characterize a systems engineer, which *pedagogical model* should be used in teaching this discipline, what characteristics the learning environment and context should have, the role of educators in this context, what can substitute the professional experience, etc. Interesting statistics presented in [13] suggest that the number of formal education courses in the U.S., related to systems engineering, have passed from 30 in 2000, 48 in 2005, 69 in 2010, to 282 in 2016, including both systems engineering and industrial engineering programs.

In this context of seeking good practices and means to efficiently teach systems engineering, the Placis[2] project, in which this thesis was performed, has been initiated by three French engineering schools, along with industrial partners, in order to create a collaborative educational platform for systems engineering.

1.2 Objectives and main issues

Our main goals in this thesis is the improvement of SE education by the design and development of an appropriate systems engineering educational environment, by investigating the different practices in the field of SE education. In this thesis we are mainly addressing the fact of educating systems engineering principles and practices, to undergraduate and

[2]Placis Project: Collaborative Platform for Systems Engineering, driven by Supméca. And funded by the French Ministry of Higher Education and Research under the future investments program with the reference ANR-11-IDFI-0029

graduate students. The obtained results might be transferable in the future, to industrials and organizations, who have careers plan to let specific domain engineers become systems engineers. However, we can't expect the same outcomes, as within systems engineering practitioners, systems engineering principles and practices are being acquired by directly working on real projects. In order to propose an adequate systems engineering educational environment, We start by considering the following three questions:

First, why is teaching systems engineering so important, and why shouldn't we treat it as any ordinary engineering discipline, if we consider it as a discipline?

Second, are the current practices to teach systems engineering good enough, considering the importance of this discipline to the different industries and organizations?

Third, from which perspective should we ideally analyze the problem of teaching systems engineering, in order to be able to provide innovative and efficient approaches and means for future systems engineers training?

We had no difficulties to find answers to the two first questions. As to the first one, because of its multidisciplinary nature, SE cannot and should not be treated like any other regular engineering discipline. Teaching SE differs from traditional mono-disciplinary engineering courses, since the training needs to focus more on skills and less on transferable facts, especially in order to produce good systems engineers. A particular attention should be payed to soft skills inculcation [14]. Also, teaching SE is about processes, about learning how to think like an engineer; it is much more than a prescription of content [15], and one of the most important skills we need to inculcate to systems engineers is systems thinking. Are the current practices used to teach systems engineering good enough? For both systems engineering practitioners and academia, this is definitely not the case. As stated by some researchers such as Wasson [10], systems development projects still experience technical performance issues, in relation with the lack of well trained systems engineers. And this, despite the multiplication of education and training courses, the formulation and development of capability assessment and competency models, certifications, etc. Wasson formulated five issues, in relation with this question within the U.S. Department of Defense (DoD), but he observed them in other business domains, too. These issues are [10]:

- *Issue 1: Key systems engineering practices known to be effective are not consistently applied across all phases of the program lifecycle.*
- *Issue 2: Insufficient systems engineering is applied early in the program lifecycle, compromising the foundation for initial requirements and architecture development.*
- *Issue 3: Requirements are not always well-managed, including the effective translation from capabilities statements into executable requirements to achieve successful acquisition programs.*
- *Issue 4: The quantity and quality of systems engineering expertise is insufficient to meet the demands of the government and the defense industry.*
- *Issue 5: Collaborative environments, including SE tools, are inadequate to effectively execute SE at the joint capability, system of systems (SoS), and system levels.*

The third question however was not easy to answer. It directed us to a deeper investigation of the subject, in order to finally propose effective means to teach systems engineering. The issues we are considering as our research questions are:

Question 1: Systems engineering standardized processes, like those from the ISO/IEC/IEEE

15288 standard [4], encompass the fundamental principles of SE. According to the literature, there is no doubt about the interest of these SE standardized processes, within a SE educational environment. We think that all kind of processes may play a good role within a systems engineering educational environment, from technical processes to technical management processes, passing by agreement processes, and projects enabling processes. However, is there a good way to efficiently incorporate them within the learning scenarios, while providing/preserving:

- For students, the ability to acquire the necessary critical thinking, soft skills, the holistic view capabilities when engineering a system, and to master the different SE principles incorporated within these processes.
- For educators, the ability to manage the size of their learning projects, to adapt these processes to their projects' size and their learning goals, and to ensure that the processes are really executed as intended, and that the student really learned the fundamental and critical principles of systems engineering.

Question 2: There are several SE competency models, intended to be used for various purposes, including for managing the competencies of systems engineers, within systems engineering practitioners and academia.

The question is, which competency model should be applied to manage the acquired/to be acquired competencies in a SE educational environment, and how could this be done within the environment that incorporates the standardized SE processes of the first question? Even if students assessment is not the focus of this thesis, will the use of processes and competencies will allow the emergence of better ways to assess students acquired competencies and skills in SE?

Question 3: Numerous researchers and practitioners agree that SE is better taught through On-The-Job activities within real projects. But it is not always possible to teach SE this way because of the generated costs, the companies confidentiality and intellectual property protection policies, and the insufficient number of companies willing to incorporate interns in this discipline. As another reason, according to [16], we can no more count only on job experiences after graduation. The fact is that there are fewer and fewer opportunities to apply the processes and learn from experience. The authors assumed that a today's graduating engineer can expect only one military aircraft development program during his career, because of the increasingly longer program development times.

Most importantly, in most of cases companies that seek systems engineers do not have the time to train them everything about systems engineering. They hope to recruit engineers with at least a solid background about fundamental principles of SE, so that they can quickly adapt their knowledge to company methodologies, methods and tools. However, researchers and educators agree that the best way to teach SE, in the absence of On-The-Job experience, is the adoption of a Project Based Learning (PBL) approach, ideally in a geo-distributed, and culturally diverse educational environment.

The question is, how to incorporate a PBL approach while using a standardized processes based learning scenarios, driven by competencies management? How to decide and manage which supports and pedagogical resources to provide students with? How to avoid or broaden the well-known boundaries of a PBL approach, by exploiting technologies forces, such as the authenticity of PBL experiences compared to industry design experiences, the management of multidisciplinary engineering teams, and the evaluation of the performed work in regards to the quality of the produced results, along with the quality of the execution?

Question 4: Providing coaches, or better industrial coaches [9], is highly recommended for PBL based SE educational environments. If educators are the coaches, what should be the relation between educator and students within the SE educational environment? How to materialize this relation in a technology-based environment? What kind of interactions should or should not exist between them? In the case where educators lack industrial experience in SE, which is a very common situation, should industrial partners be involved in the education scenarios, and how?
An important thing not to be neglected is the role of technology, especially of educational technologies that allow the implementation of a distant PBL approach, helping students and educators to communicate without adding any further cognitive load.

1.3 Contributions

The main focus and contribution of this thesis is about improving SE education, by finding the best way to integrate SE standardized process in learning scenarios, within a PBL-based SE educational environment. During this thesis, several significant contributions have been made.

1. First, a survey about current practices on SE education, and their classification was realized [17]. We concluded our survey by a number of recommendations on how to improve SE education. This survey outcomes have been extended a lot in this thesis manuscript, by both telescoping the breadth of todays practice in SE education, and in-depth micro-scoping of these practices operations.

2. A new approach for SE education based on the use of standardized SE processes was designed. In this approach, each learning action is seen as a process, standardized or not, tailored or not. It allows the educators to adapt the processes, to the project size, learning goals, etc.

3. A web-based environment was developed to support this approach, allowing the use of processes to define the most adequate learning scenario for each project. An environment that allows geo-distributed students belonging to the same team, by executing the systems engineering processes as defined/tailored by the educator to engineer the requested system, while being assisted by the educator and/or an industrial expert.

4. Proposition of a new approach to match SE competencies from any SE competency model (like the INCOSE Competency framework, or Nasa Appel competencies) with the activities of SE standard processes (like the 15288 standard and 29110 series). This was the base for a new component: a "competencies management system", that was added to the proposed solution. This component assists the educator in better defining his learning scenarios for each new project, according to the competencies to be taught. We can then imagine that these competencies will be used, in the future, to better assess the students acquired knowledge, but also to select students participating in a specific project, based on their already acquired competencies, in addition to the competencies to be acquired.

5. Enabling new methods to evaluate students and improving at the same time PBL assessment challenges. This is done by giving the educators, in addition to the competencies management system, a vision of everything that is happening during

the system engineering lifecycle. The solution opens the door to implementations of novel methods of students assessment, regarding their final results, execution quality, along with their acquired knowledge and skills. Also, we are responding to other PBL challenges, such as controlling the complexity of the project, enabling geo-located teams management, distributing our solution in time and place, etc.

6. The first version of the proposed educational environment, -without the competencies management system-, has been evaluated by students from different engineering schools and universities, who have a minimum of knowledge in SE. Then, the second enhanced version, was experimented during a three-days doctoral training courses, introducing them to SE. We also propose at the end of this manuscript, a set of important directions to be considered in order to take the proposed approach and its supporting educational environment to another stage of efficiency and to made it fit a new kind of situations, for both educational and industrial purposes especially for small entities.

Throughout the lecture of this thesis, it is important to remember that systems engineering is a multidisciplinary approach. One of the disciplines which takes a major part of the systems engineering efforts, and which looks itself like systems engineering in nature, methods, and processes, is Software Engineering. As a matter of fact, most of nowadays complex systems include, or are even based on, important and complex software systems.

Therefore, even though we decided to focus on systems engineering in this work, we highlight the fact that the proposed solution addresses Software Engineering Education as much as it addresses Systems Engineering Education. It can be used, very naturally, to promote software engineering education within a systems engineering educational environment or within a dedicated software engineering educational environment. For this purpose, the ISO/IEC/IEEE 12207 standard (Systems and software engineering-software lifecycle processes) or the software engineering profile of the ISO/IEC 29110 series can be used. This projection can also be confirmed by the nature of the 12207 standard because, since 2015, it is harmonized in structure and content with the 15288 systems engineering standard [4].

1.4 Document organization

After explaining the context of our work, we present within chapters two to four our literature study and analysis, related to our topics of interest. These topics are about the importance of systems thinking and systems engineering in order to manage complexity, SE standards and processes, SE competency models and systems engineers competencies, and then the importance of SE education, characteristics, and current practices. In chapter five, a full description of our vision about a perfect SE educational environment is provided, followed by the presentation of our first contribution, i.e. a process centered adaptive approach for teaching SE, in a PBL context. In chapter six, the architecture of the competency management system will be presented, and how it can be used with any SE competency model, in addition to the relationship between the competency model and and the SE standardized processes within the learning scenario. We also highlight the added value of this component within the proposed solution. In chapter seven, the developed supporting web platform is presented in detail, along with its scenarios of use, before presenting our two experimentation plans and their results. We conclude this manuscript by the lessons learned from this thesis, and the perspectives for further work in the domain of educating systems engineers.

2 Systems Thinking and Systems Engineering

2.1 Introduction

In this chapter, we are interested on what systems engineering is and what relation it has with systems thinking, while providing different definitions of systems engineering. Then we will show the differences between systems engineering and domain engineering, before evoking the origins and evolution of systems engineering, while relating it to its international standards. we conclude this chapter by additional details about two standards that we are interested in.

2.2 Systems thinking

2.2.1 What is Systems Thinking

Systems Thinking (ST) is an undissociated part from systems engineering. Is is what provides engineers by a systemic view of their systems being engineered. ST allows engineers to perform their systems engineering tasks (technical, managerial, etc.), not forgetting that their system is a part of something bigger, regardless on which system level they are working. ST allows engineers to see the whole of a system, and to pay close attention to the interrelationships between systems element and the relationships with the environment. Systems thinking is both more challenging and more promising than our normal ways of dealing with problems. It helps us to see and understand complex situations. According to Senge [18], "*Systems thinking is a discipline for seeing wholes. It is a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static "snapshots.""*. He argues that because of the complexity overwhelming us today, systems thinking is needed more than ever, and it might be the antidote to the sens of helplessness that many engineers feel when they face the interdependence of todays systems.

Wasson in [10], said that systems thinking skills are an integral part of systems engineering attributes. Greene and Papalambros [19] believe that a wide adoption of systems thinking in engineering remains limited, and adding this to the fact that soft-skills are neglected in technical professions, led to a limited appeal to the holistic approach of systems thinking. They assume that *"systems thinking —like all thinking— is at its core an exercise in cognition, and relies upon high-order cognitive skills and a knowledge of why, when, and how to utilize them"*. To develop the skill of ST, Davidz and Nightingale [20] showed that ST development enabling mechanisms are: experiential learning, certain individual characteristics and a supporting environment.

According to Daniel Allegro et al. [21] "*systems thinking supports the reasoning that identifies, defines and analyzes a complex system. It responds to the need for understanding, which doesn't necessarily imply an action. Systems thinking alone doesn't enable the system to be realized*". The authors also stated that, systems thinking enables to ask the rights questions at the right moments. To fully understand a situation, this thinking mode is involved for decisions making, taking opportunities, or managing risks. Then, the authors proposed a mind map for systems thinking, which includes two aspects of systems thinking: Systems thinker attitudes, systems thinking concepts. Each of the proposed attitudes and concepts are then divided into several elements and levels.

Boardman et al. [22] , stated that *"A holistic approach is required as opposed to piecemeal efforts. This is why Systems Thinking is being seen as a much more valuable resource – because it gives analysts and problem solvers a vantage point to see the bigger picture."*. Boardman proposed, what he call "a Conceptagon", to make an important addition to the systems thinking toolkit. The major contribution of the Conceptagon, according to the authors, is to distill the essence of the different "systems languages" proposed by different systems thinking pioneers, into a set of concepts that are familiar to a wide variety of domain specialists.

2.2.2 Holistic systems thinking

It appears clearly that the Holistic view of a system is the core of systems thinking. According to Kasser et al. [23], there are nine viewpoints from which a system should be viewed in a systemic and systematic way. This is known as the "*holistic systems thinking*". The nine viewpoints are:

- Big picture perspective: in order to understand the whole system, including its interactions with adjacent or more distant systems, and with its environment.
- Operational perspective: it is about understanding how the system operates or will operate, without going deeper in the details. This is what we call the "*black box*[1] *perspective*". This perspective is often documented using, *use cases* and *concept of operations* [2].
- Functional perspective: allows the understanding of the functions of a system without linking them to the elements of the systems that perform them. The system is viewed here as a *white box* [3].
- Structural perspective: includes physical, technical, and architectural views of the system, and enables the understanding of the interconnections between elements and sub-systems.
- Generic perspective: places the system in the context compared to similar ones in other domains, in the present or in the past.

[1] A black box is a device, system or object which can be viewed in terms of its inputs and outputs (or transfer characteristics), without any knowledge of its internal workings [Wikipedia]

[2] A Concept of Operations (CONOPS) is a user-oriented document that "describes systems characteristics for a proposed system from a user's perspective. A CONOPS also describes the user organization, mission, and objectives from an integrated systems point of view and is used to communicate overall quantitative and qualitative system characteristics to stakeholders[24]

[3] A white box (or glass box, clear box, or open box) is a subsystem whose internals can be viewed but usually not be altered. [Wikipedia]

- Continuum perspective: considers that a problem may have more than one correct solution, and that a system can show different types of behaviour with different conditions. helps analyzing the failure modes of a system and its components.
- Temporal perspective: concerns the behaviour of the system over time, including using past patterns to predict future ones and to assess if enough data about the past patterns and their environment are understood.
- Quantitative perspective: intended to be used to develop the performance requirements. It is related to the three first perspectives, big picture, operational and functional perspectives.
- Scientific perspective: while the first eight perspectives are descriptive, this one is prescriptive. It covers the formulation and testing of hypothetical candidate representations of the solution system.

According to the same authors, the previous approach was developed from the only systematic and systemic approach discovered in the literature, for applying systems thinking, called the seven streams of system thinking mentioned in [25], as: dynamic thinking, closed-loop thinking, generic thinking, structural thinking, operational thinking, continuum thinking, and finally scientific thinking. In addition, some popular systems thinking methods and tools have been presented in the INCOSE systems engineering handbook [2]. This subject has also been treated by the Systems Engineering Body of Knowledge Sebok. [26]

2.3 Systems Engineering and Systems Thinking

According to Frank [27], systems thinking development can be accelerated within well designed and well taught systems engineering courses. This skill can be acquired through education, On-The-Job experience, and training [20], and systems thinking development is enabled by experience, individual characteristics, and supporting environment [20]. The experience element is highlighted by Davidz et al. [20] as: "*To develop systems thinking, systems engineering education and training programs should include experiential opportunities*" .

Frank [28] thinks that systems thinking skills enable individuals to successfully perform systems engineering tasks. Beasley and Partridge [1] state that the nature and some of the outcomes of systems thinking application are what differentiates systems engineering from other kinds of engineering. Therefore, systems thinking skills are required to do systems engineering. For Beasley and Partridge, "*Systems Thinking is the underpinning skill required to do Systems Engineering [...] It is radically different from the more common analytical or reductionist thinking found in most engineers*" [1].

2.4 Systems engineering

In short words, systems engineering exists to address the complexity of systems, in order to be able to transform users and stakeholders needs into operational systems via an interdisciplinary process. But how many different influential organizations define it?

2.4.1 System definition

One of the most used definitions of a system can be found in the general system theory book [29]. For Bertalanffy, *A system is a set of elements in interaction.* Even if this definition remains very general, it allows encompassing notions like the wholeness, holistic, organismic, etc., which all signify that, in the last resort, we must think in terms of systems of elements in mutual interaction.

This idea of wholeness is highlighted by Boardman and Sauser [30], when they state: "*We believe that the essence of a system is* "togetherness"*, the drawing together of various parts and the relationships they form in order to produce a new whole".*

NASA defines a system as follows: "*A system is a set of interrelated components which interact with one another in an organized fashion toward a common purpose*" [31].

A consensus has been found by the International Council on Systems Engineering (INCOSE). The adopted definition is too far away from the previous one, since INCOSE considers a system as: " *a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behavior and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected*" [32], [33].

Another recent definition, that considers the fact that a system can be considered as a product or as the services it provides, is the ISI/IEC/IEEE 15288:2015 standard definition: "*A system is a combination of interacting elements organized to achieve one or more stated purposes*". this definition also highlight the fact that: "*A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.*"

2.4.2 Systems engineering definitions

1. **First definition:** According to the first 2015 version of ISO/IEC/IEEE 15288 Standards, *"Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It 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. It Integrates all the disciplines an specialty 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 stakeholders with the goal of providing a quality product that meets the needs of users and applicable stakeholders"*[4]. This same standard also resume this definition as: "Interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholders needs, expectations, and constraints into a solution and to support that solution throughout its life".

2. **Second definition:** for the International Council on Systems Engineering (INCOSE), *"Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support,*

test, manufacturing and disposal. Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs." [32].

The INCOSE definition, is laterally the same as the ISO/IEC/IEEE 15288 standard definition, as they adopted this standard within the fourth version of their systems engineering handbook. They only added a definition of the complete problem as: "operations, cost and schedule, performance, training and support, test, manufacturing and disposal", and they replaced the term of stakeholders, with the customer, and the user.

These two definitions encompasses the main principles and characteristics of systems engineering, such as the need to define system requirements and functionalities early on in the development cycle, treating the system as a whole, etc. However, we assume that it lacks some important notions. First, this definition does not express the importance of traceability, e.g. for requirements, and the ability to adapt them over the system lifecycle. More importantly, it seems to ignore the importance of considering the system in its environment over time and enabling multiple views. Finally this definition only focuses on the systematic part and ignores the systemic one, passing from system requirements and functionalities definition to system retirement as a procedural process, which is not the case, especially for large complex systems. According to Kasser, "*The INCOSE definition is a compromise designed to satisfy everyone and fails to do so*" [34].

3. **Third definition:** Blanchard and Fabrycky define SE as, "*Basically Systems Engineering is good engineering with special areas of emphasis, which includes: a top-down approach, a lifecycle orientation, a better and more complete emphasis on definition of requirements, and interdisciplinary approach*" [35].

   This definition re-emphasizes the importance of requirements and their traceability, and the lifecycle orientation that suggest that (if we understand the definition of a lifecycle) the different natures of systems engineering processes imply a non-procedural execution of the lifecycle processes. However, it omits any consideration concerning both the wholeness and the system environment while engineering a system.

   According to [1], the two previous definitions failed in distinguishing the system engineering discipline from other regular engineering disciplines. However, we do not completely agree on this conclusion despite the lacks we noticed.

4. **Fourth definition:** In the definition used in Rolls-Royce currently, "*Systems Engineering is applying the concept of a system to a situation in order to gain insight and understanding (Systems Thinking), in a systematic and repeatable manner (Systems Approach), to the realization of a new system or the modification of an existing one. Where a system is an assembly of components (technologies, people, information, etc.), connected together in an organized way to form a whole, this whole showing properties of the whole, rather than properties of the components. A system has systemic properties and characteristics which we use to understand and make predictions about the problem or situation under investigation.*" [1]

5. **Fifth definition:** in the"NASA Systems Engineering Handbook", systems engineering is defined as follows: "*Systems engineering is a methodical, disciplined ap-*

proach for the design, realization, technical management, operations, and retirement of a system. A "system" is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce system-level results. The results include system-level qualities, properties, characteristics, functions, behavior, and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts. It is a way of looking at the "big picture" when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the systems. In other words, systems engineering is a logical way of thinking." [36]

These two last definitions, from Rolls Roys and NASA, are quite similar after careful review. They are the particularly explicit about the fundamental notions of systems engineering. Also, both of them mainly focus on the indispensable ideas of wholeness and interrelationships.

In this work, We are adopting the definition which has been defined by a consortium of experts, from different industries and different organizations. A definition that considers both the technical and managerial efforts, and which is adopted for ISO/IEC/IEEE 15288.

2.4.3 Domain Engineering vs. Systems Engineering. Is Systems Engineering a discipline?

Engineers can be classified into two categories: those with an in-depth expertise in a technology or a specific domain, and others with extended abilities and considerations to the boundaries of a problem. The former set are labeled "domain engineers" (mechanical engineer, computer science engineer, etc.), the latter are referred to as "systems engineers". More in depth details can be found in [37], including how the two kinds of engineers should be trained. Additional information about the relationships between domain engineering and systems engineering, with their convergent and divergent thinking is given.

A discipline is a particular field characterized by its own principles and methods [38]. According to Hyer [38], considering systems engineering as a separate field which can be taught independently from other fields is a common misconception. Hence, systems engineering should not be categorized with technical disciplines, neither should it be considered as a purely technical management responsibility. However, Hyer believes that it should be considered as a set of activities that can be used in a team environment to apply principles and technologies from different disciplines to solve problems.

Kasser and Hitchins [39] argue that systems engineering is a discipline. However, they also agree with Hyer that systems engineering is not a traditional engineering discipline. It can and should be differentiated from other disciplines. Kasser and Hitchins identified six overlapping camps for systems engineering: Lifecycle, Process, Problem, Discipline, Systems thinking and non-systems thinking, and Enablers.

2.4.4 Comparison of process (systematic) and skill (systemic) levels

We can see systems engineering through two different but valid lenses. The first one is the "the systems thinking concepts" lens, and the second one is "the SE Process concepts" lens

[1]. The full understanding of systems engineering should pass through the assimilation of both concepts, and these two views can be summarized as:

SE = Process*: individuals must know what to do, and if you do Systems Engineering, success will come.* [1]

SE = The skill of Systems Thinking*: the focus is on getting the right skills into organization roles, the right thinking will create better solutions.* [1]

Systems thinking skill for engineers is not enough alone, they also need a SE process to follow. Neither a good SE process without a skilled systems engineer is sufficient. Figure 2.1 below from [1] characterizes different balances between process and skill.

Being Systematic	**Good process**	**No invention / rework** No real thinking, Systems process methods "by the numbers" methods, ineffective Systems Engineering	**A very good chance** The right understanding of problem – skilled and appropriate use of process and method
	Poor Process	**No chance** Cannot handle complexity and unlikely to do well!	**No control** Individuals "uninterested" in process –no control, critical things missed, reinvention, some "mavericks"
		Poor Systems Skills	**Good Systems Skills**
		Being Systemic	

Figure 2.1: Comparison of process (systematic) and skill (systemic) levels [1]

2.5 Systems engineering origins and evolution

The term "Systems Engineering" dates back to Bell Telephone Laboratories in the early 1940s [40]. According to Hall, Mr. Gilman, Director of Systems Engineering at Bell, was the first who attempted to teach systems engineering as we know it today in 1950 at MIT.

In 1978, the concepts of systems engineering at Bell Labs have been traced back to the early 1900s. Before that, the notion of systems analysis, an important part of systems engineering, was created in 1946 by the United States Air Force. The Department of Defense entered the world of systems engineering in the late 1940s with the initial development of missiles and missile-defense systems [41].

A lot of research and development efforts have been made by the U.S. and its allies for developing principles, methods, processes, and tools for military defense systems, complemented by initiatives addressing industrial and governmental systems. These efforts and the huge investments have been done as a response to the challenges of the Second World War concerning the complexities of real-time command and control of extremely large systems, along with the the Cold War and Russian space achievements. Afterwards, the Software Engineering discipline has been defined as a response to the rise of software parts within systems, as a closely related discipline to systems engineering [26], [42]

Figure 2.2 bellow illustrates the important dates in the origins of systems engineering. Note that additional actions happened like the different INCOSE national chapters, established in several countries, and the ISE/IEC/IEEE 15288:2015 is not included at this figure.

1937	British multidisciplinary team to analyze the air defense system
1939–1945	Bell Labs supported NIKE missile project development
1951–1980	SAGE air defense system defined and managed by Massachusetts Institute of Technology (MIT)
1954	Recommendation by the RAND Corporation to adopt the term "systems engineering"
1956	Invention of systems analysis by RAND Corporation
1962	Publication of *A Methodology for Systems Engineering* by Hall
1969	Modeling of urban systems at MIT by Jay Forrester
1990	National Council on Systems Engineering (NCOSE) established
1995	INCOSE emerged from NCOSE to incorporate international view
2008	ISO, IEC, IEEE, INCOSE, PSM, and others fully harmonize SE concepts on ISO/IEC/IEEE 15288:2008

Figure 2.2: Important dates concerning the origins of systems engineering [2]

After 1990, because of the boom and the rapid growth of information systems and related technological advancements, systems engineering faced more challenges as systems became more interconnected, more complex, and seeking for more interoperability. The most important challenge is making future systems scalable, stable, adaptable, and human [26]. Many systems engineering practices have evolved in response to these challenges, such as **lean, agile, iterative and evolutionary systems engineering approaches**. **Enterprise systems engineering (ESE) approaches** have been developed, which consider the enterprise itself as a system to be engineered. And the increased need for up-to-date, authoritative and shared models to support lifecycle decisions has led to the development of **model-based systems engineering (MBSE) approaches**[26]. *"All forms of systems engineering will be needed to solve the engineering problems of the future, sometimes separately but increasingly in combination"* [43]

The issues discussed above are considered in the INCOSE Systems Engineering Vision 2025 [44]. This vision gives an overview of both the likely nature and challenges of future systems along with the question of how systems engineering needs to evolve in order to catch up with these challenges. Some of the expected evolutions, in relation with our subject, are [44]:

- *Systems engineering practices will adapt to and be transformed by new technology as efforts become more IT-centric and globally distributed among diverse collaborating enterprises.*
- *Systems engineering's theoretical foundations will advance to better deal with complexity and the global demands of the discipline, forming the basis for systems education as well as the methods and tools used by practicing systems engineers for system architecting, system design and system understanding.*
- *Methods and tools, based on solid theoretical foundations, will advance to address the market demands of innovation, productivity, and time-to-market as well as product quality and safety by harnessing the power of advancements in modeling, simulation and knowledge representation, such as domain-specific standard vocabularies, thereby meeting the needs of an increasingly diverse stakeholder community.*
- *Systems engineering will lead the effort to drive out unnecessary complexity through well-founded architecting.*
- ***Education and training of systems engineers and the infusion of systems thinking across a broad range of the engineering and management workforce will meet the demands for a growing number of systems engineers with the necessary technical and leadership competencies. and deeper system understanding.***

2.6 Systems engineering standards

2.6.1 SE Standards Overview

The early standard for Systems Engineering was the US Military Standard MIL-STD-499, Engineering management, published on July 17th, 1969 and produced by the US Department of Defense (DoD) for the defense industry [8]. It has been adapted twice after that: the MIL-STD-499A released on May 1st, 1974, and the MIL-STD-499B draft of 1992. In 1994, a group of organizations called Electronic Industries Alliance (EIA) collaborated to develop a commercial systems engineering standard meant to replace the military one. This group included representatives from the DoD, the Aircraft Industry Association (AIA), the National Security Industries Association (NDIA), the Institute of Electrical and Electronics Engineers (IEEE), and INCOSE [8]. In December 1994, they released the EIA Interim Standard 632 (EIA/IS 632) Systems Engineering. This Standard became later the ANSI/EIA 632-1998, Processes for Engineering a System [45], which has been approved on January 7th, 1999.

IEEE on one side, The International Organization for Standardization (ISO) along with the International Electro-technical Commission (IEC) on the other side, were also working on commercial systems engineering standards. In 1998, and after a trial-use version in 1995 (IEEE Std 1220-1994) [46], IEEE came up with the IEEE Std 1220-1998, Standard for Application and Management of the Systems Engineering Process [47], and in 2002, ISO and IEC released the ISO/IEC 15288 standard, Systems Engineering-System Lifecycle Processes [48], which has been created by the same group that created the ISO/IEC 12207 software life-cycle standard, along with Systems Engineering Experts [8].

Each of these three different commercial standards, EIA 632, IEEE 1220 and the ISO/IEC 15288 addressed different breadth and depth levels in the Systems Engineering processes, described in Figure 2.

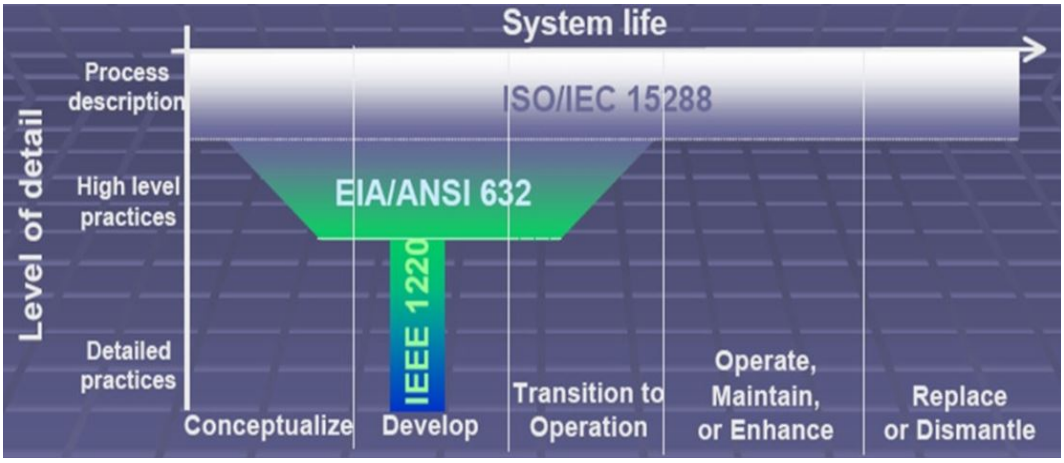

Figure 2.3: Breadth and Depth of Leading SE Process Standards [3]

The last active version of the EIA 632, Processes for Engineering a System, is still the one approved in 1999 and reaffirmed in 2003. The IEEE 1220 has been revised once in 2005 and is the currently active version of this standard [49]. The ISO/IEC 15288-2002, after its adoption by IEEE in 2004, has been revised and became the ISO/IEC 15288:2008 [50], before it became the ISO/IEC/IEEE 15288:2015 [4] which was prepared by Joint Technical Committee ISO/IEC JTC 1 (Information technology), Subcommittee SC 7 Software and systems Engineering) in cooperation with the IEEE Computer Society

Systems and Software Engineering Standards Committee [4]. In 2004, ISO/IEC 15288-2002 has been adopted by IEEE, in the IEEE Std 15288$^{\text{TM}}$-2004 [51], Also, INCOSE (International Council on Systems Engineering) adopted this standard and aligned upon it the process and lifecycle content in their 4th Version of the SE Handbook [2].

By gathering the most important normalization institutions and big industrial players, ISO/IEC/IEEE 15288 became the most revised and the most complete standard for systems engineering, even the starting point of many derivative products, different institutions and researchers producing content, guidance, reports, use cases, etc. Others are creating completely new products, such as the ISO/IEC 29110 series [52] which is a Systems and Software Lifecycle Profiles and Guidelines for Very Small Entities (VSEs). *"A very small entity (VSE) is an enterprise, an organization, a department, or a project having up to 25 people"* [53]. ISO/IEC 29110 is mainly based on ISO/IEC/IEEE 15288 for the systems engineering part, and on the ISO/IEC/IEEE 12207 for the software engineering part.

We surveyed a sample set of standards, to show their importance in the field of systems engineering. For more information about systems engineering standards, especially in specific disciplines, please take a look at the Incose website [54]. Figure 2.4 illustrates a non exhaustive list of todays systems engineering standards, one of the most important missing from the Figure, is the ISO/IEC 29110 series, evoked previously. Most importantly, no standardized systems engineering processes fit any kind of programs, projects, and systems. A framework for instantiating product-specific processes, for product quality achievement, was presented by Natarajan et al. in [55]

ISO/IEC/IEEE 15288	Systems and software engineering—System life cycle processes
ANSI/EIA-632	Processes for engineering a system
ISO/IEC/IEEE 26702	Systems engineering—Application and management of the systems engineering process (replaces IEEE 1220™)
SEBoK	Guide to the systems engineering body of knowledge
ISO/IEC TR 24748	Systems and software engineering—Life cycle management—Part 1, guide for life cycle management; Part 2, guide to the application of ISO/IEC 15288 (system life cycle processes)
ISO/IEC/IEEE 24765	Systems and software engineering—Vocabulary
ISO/IEC/IEEE 29148	Software and systems engineering—Life cycle processes—Requirements engineering
ISO/IEC/IEEE 42010	Systems and software engineering—Architecture description (replaces IEEE 1471)
ISO 10303-233	Industrial automation systems and integration—Product data representation and exchange—Part 233: Application protocol: Systems engineering
OMG SysML™	Object management group (OMG) systems modeling language (SysML™)
CMMI-DEV v1.3	Capability maturity model integration (CMMI®) for development
ISO/IEC 15504-6	Information technology—Process assessment—Part 6: An exemplar system life cycle process assessment model
ISO/IEC/IEEE 15289	Systems and software engineering—Content of systems and software life cycle process information products (documentation)
ISO/IEC/IEEE 15939	Systems and software engineering—Measurement process
ISO/IEC/IEEE 16085	Systems and software engineering—Life cycle processes—Risk management
ISO/IEC/IEEE 16326	Systems and software engineering—Life cycle processes—Project management
ISO/IEC/IEEE 24748-4	Life cycle management—Part 4: Systems engineering planning
ISO 31000	Risk management—Principles and guidelines
TechAmerica/ANSI EIA-649-B	National consensus standard for configuration management
ANSI/AIAA G-043A-2012e	ANSI/AIAA guide to the preparation of operational concept documents
ISO/IEC/IEEE 15026	Systems and software assurance—Part 1, concepts and vocabulary; Part 2, assurance case; Part 4, assurance in the life cycle

Figure 2.4: Current significant Systems Engineering standards and guides [2]

2.6.2 Considered Standards

Since we are aiming to use standardized processes for systems engineering education purposes, it seems natural that a decision needs to be made about which standards we want to adopt. Based on the previous overview of the different standards for Systems Engineering processes, the use of ISO/IEC/IEEE 15288 and ISO/IEC 29110 appears to be the appropriate choice. As a matter of fact, one of these standards covers the entire lifecycle of a system with a large number of different individual processes, while the other one is a smaller set of processes focusing on the need of VSEs. Some of their advantages are listed next. Note however that the previously cited standards, even the selected ones, are usually not sufficient to engineer a good system in real life, because none of them explains in detail how to apply them. For this reason, these standards are typically supported by Systems Engineering approaches, and applied in conjunction with many other standards such as the TR 24748, guide to 15288, the 42010 for architecture description, the 29148 for requirements engineering, 15289 documentation Standard and others.

The ISO/IEC/IEEE 15288 Standard

The most significant characteristics of the ISO/IEC/IEEE 15288 are:

- It is the systems engineering reference standard, and it is promoted by mainly all the standardization organizations including ISO and IEEE, and adopted by INCOSE. It is up-to-date and based on proven practices.

- Our first goal is to let students learn the fundamentals of systems engineering, but we do not want to bother them with much detail relative to deeper processes application. We rather want educators to be able to select most suited topics to learn for specific students and for specific kinds of systems. This standard provides them with the ability to do that.

- Its processes can be applied in different manners: concurrently, iteratively and recursively to a system, and incrementally to its elements. It can be applied to an element of a system, considered as a system itself, as it can be applied at any level in the hierarchy of a system across its lifecycle.

- It applies to man-made systems configured with one or more of the following, hardware, software, humans, or processes.

- When defining the lifecycle model and its different stages, educators can choose which of this standard processes to consider and to be in conformance with. However, its processes can also be tailored to fit for example a specific learning goal.

- It can be used alone or jointly with the ISO/IEC/IEEE 12207, for software engineering, which has the same terminology and concepts.

In its last revision of 2015, it includes 30 processes grouped into four categories, see Figure 2.5. It includes 2 **Agreement** processes, 6 Organizational **Project-Enabling** processes, 8 **Technical management** processes, 14 **Technical** processes. For each process, the standard provides us with: its **Purpose** as a paragraph that describes at a high level the overall goal of performing the process, its **Outcomes** that express the observable results expected from the successful execution of the process, its **Activities** providing the first level of structural decomposition of a process, they generally provide a set of the related lower-level elements called tasks, and its **activities tasks**: tasks are requirements, recommendations, or permissible actions intended to support the achievement of the outcomes. In addition and some Notes. The standard provides a common processes framework for describing the lifecycle of systems created by humans, adopting a systems engineering approach[4]. It does not describe a specific system lifecycle model, neither a development methodology, method, model or technique.

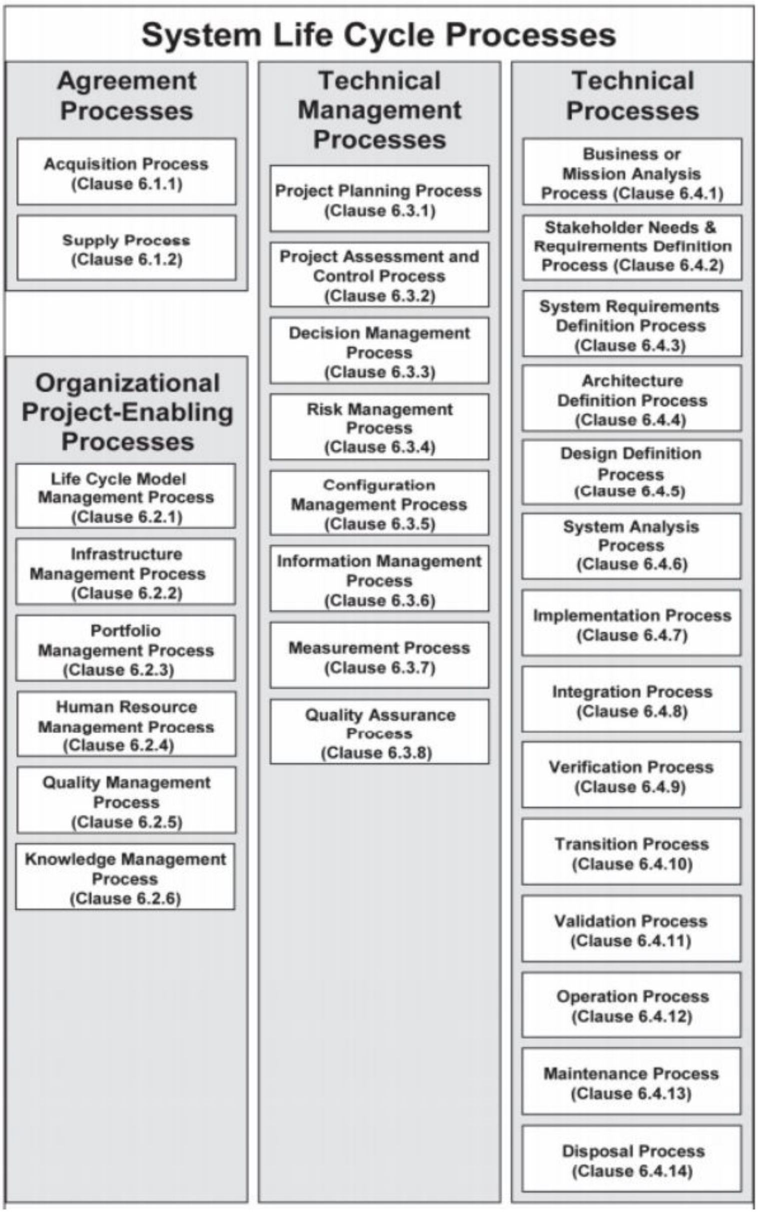

Figure 2.5: ISO/IEC/IEEE 15288:2015 System Lifecycle Processes [4]

The ISO/IEC 29110 series

The most significant characteristics and advantages of this series and its differentiation compared to the first one are:

- The recently published set of ISO/IEC 29110 international standards (IS) and Technical Reports (TR) [56] are specifically aimed at addressing the specific needs of VSEs (Very Small Entities), i.e. enterprises, organizations, departments or projects with up to 25 people.

- The engineering standards and guides developed by an ISO working group, Working Group 24 (WG24), are targeting VSEs which do not have experience or expertise in selecting, for a specific project, the appropriate processes from lifecycle standards such as ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288, and tailor them to the needs of a specific project [57].

- Building upon the success of ISO/IEC 29110 for software, in 2009, an INCOSE working group was established to evaluate the possibility of developing a standard using the generic profile group scheme of the ISO/IEC 29110 series and the systems engineering lifecycle processes standard ISO/IEC/IEEE 15288 (2008), for organizations developing systems instead of just softwares. This new ISO/IEC 29110 series is targeted for VSEs that do not have experience or expertise in tailoring ISO/IEC/IEEE 15288 to their needs. The result is the publication of a Systems Management and Engineering Guide Entry profile (ISO/IEC TR 29110-6-5-1:2015), i.e. for VSEs working on small projects (e.g. at most six person-months effort) and for start-up VSEs and Basic Profile (ISO/IEC TR 29110-6-5-2:2014) [58].

- The systems engineering basic profile is composed of two processes: Project Management (PM) and System Definition and Realization (SR). An acquirer provides a Statement of Work (SoW) as an input to the PM process and receives a product as a result of SR process execution, see Figure 2.6.

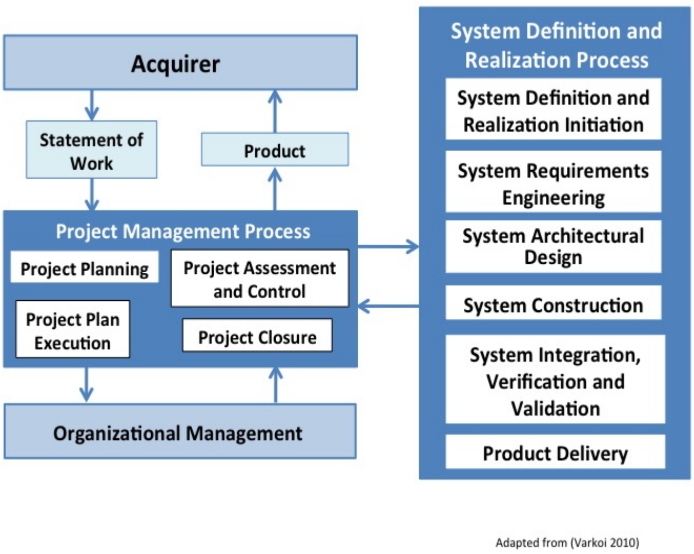

Figure 2.6: ISO/IEC TR 29110-6-5-2:2014 architecture

2.7 Conclusion

In this chapter we highlighted that each of the systems engineering and the systems thinking concepts are necessary to each other. we also presented different definitions of systems engineering, while presenting the differences between them. This made us understand that systems engineering, regardless which definition we consider, is mainly here to manage the complexity of nowadays systems. We also demonstrate in this chapter the impact of systems engineering standards on its evolution.

3

SE Competencies and Competency Models

3.1 Introduction to SE Competencies and Competency Models

System Engineering (SE) competencies are defined as *"the knowledge, skills, and abilities (KSAs) necessary for a systems engineer to perform tasks related to the discipline"* [59]. According to the Systems Engineering Body of Knowledge book (SEBoK) [26], systems engineering competencies reflect the individual's Knowledge, Skills, Abilities, and Attitudes (KSAAs), which are developed through education, training, and on-the-job experience. According to the same source *"For an individual, a set of KSAAs enables the fulfillment of the competencies needed to perform the tasks associated with the assigned systems engineering role"*.

Systems engineering competencies which, in a specific way, form A systems engineering competency model, reflect the individual's KSAAs. The KSAAs in their turn are related to the different roles in the company/project, which are then associated with a set of tasks. So, a competency model is a framework for organizing a collection of observable KSAs/KSAAs. According to [26], *"SE competency models generally agree that systems thinking, taking a holistic view of the system that includes the full lifecycle, and specific knowledge of both technical and managerial systems engineering methods are required to be a fully capable systems engineer"*.

KSAAs can be used as learning objectives for systems engineering competencies development, especially when defining them in terms of a standard taxonomy, as authors did in [42]. They developed a systems engineering competency career development model, as an analytical approach using Bloom's taxonomy. More generally, in addition to the ability to use them for education, training, and career development, competency models can also be used for recruitment and selection, and human resources planning and placements [26].

"In practice, KSAs tend to be lists of statements written by, or on behalf of, candidates. These statements are targeted to specific positions and describe as applicable a number of (lists) situational challenges and outcomes in a prior position that are to be used by evaluators in a pass-fail mode when looking for qualified candidates for the specific position"[60].

Competency models, can also be used by universities in order to manage the systems engineering competencies and skills which students should develop before and during industrial placement [61]. In this project preparing students for industrial placement, Hubbard explained how they were able to create a competency map identifying the relevant competencies for systems engineers, from different sources, with a focus on the INCOSE competency framework, while adding a set of soft skills. Then, they create a link to an online personal development planning tool (called RAPID), to record academic, professional, and individual development, for monitoring progress of transferable skills. Most importantly, they did a mapping with which modules enable the acquisition of necessary skills and competencies. and finally, develop a "preparing for industry" guide for systems engineering undergraduates.

Unger highlighted the relevance of systems engineering competency models in [62], by developing a systems engineering capability development plan. The author developed a systems engineering competency model (starting from the NASA competency model), before suggesting a set of development strategies for each competency area. The proposed competency model includes 9 areas of technical excellence and 6 areas in the field of interpersonal skills.

3.2 Main Competency Models

The competency model describes the competencies, key actions and behaviors that are needed for systems engineers to be efficient in their jobs.[63] There are two kinds of systems engineering competency models. The first one is the shorter "success model" that describes a small set of competencies for successful systems engineers. The longer "comprehensive model" has the objective to identify all the competencies required to fulfill a particular systems engineering role [63].

Different competency models exists for systems engineering. Most of them have been developed for specific contexts or specific organizations. Since the required competencies vary between different organizations and projects, these models can and should be tailored. The most well-known competency models in the field of systems engineering are:

- Defense Acquisition University (DAU) ENG Competency Model, to identify competencies required for the Department of Defense (DoD) acquisition engineering professionals [64].
- NASA Academy of Program/Project and Engineering Leadership (APPEL), a project management and systems engineering competency model, to improve project management and systems engineering at NASA [65].
- The MITRE Institute Systems Engineering Competency Model, to define new curricula systems engineering and to assess personnel and organizational capabilities [63]
- INCOSE UK Working Group Competency Framework, to identify the competencies required to conduct good systems engineering projects [66].

Most organizations tailor such models by including specific domain KSAAs (competencies) and other particularities of their organization. Also, different competency models can be used together in order to develop a new competency model for a specific need, or for a specific organization, as already done by White [42].

3.2.1 Defense Acquisition University (DAU), ENG Competency Model (Formerly SPRDE)

This is the U.S. Department of Defense (DoD) acquisition engineering professionals competency model. It includes 41 competencies, classified into four areas, analytical, technical management, professional, and business acumen, see Table 3.1. The lifecycle view used in the INCOSE model is evident in the ENG analytical grouping, but is not cited explicitly [26]. However additional professional skills and other components have been added to meet the needs for strong leadership [26].

Table 3.1: ENG Competency Model competencies

Analytical – **1. Mission-Level Assessment** – **2. Stakeholders Requirements Definition** – **3. Requirements Analysis** – **4. Architecture Design** – **5. Implementation** – **6. Integration** – **7. Verification** – **8. Validation** – **9. Transition** – **10. Design Considerations** – **11. Tools and Techniques** **Technical Management** – **12. Decision Analysis** – **13. Technical Planning** – **14. Technical Assessment** – **15. Configuration Management** – **16. Requirements Management** – **17. Risk Management** – **18. Data Management** – **19. Interface Management** – **20. Software Engineering** – **21. Acquisition**	**Professional** – **22. Problem Solving** – **23. Strategic Thinking** – **24. Professional Ethics** – **25. Leading Performance Teams** – **26. Communication** – **27. Coaching and Mentoring** – **28. Managing Stakeholders** – **29. Mission and Results Focus** – **30. Personal Effectiveness/Peer Interaction** – **31. Sound Judgment** **Business Acumen** – **32. Industry Landscape** – **33. Organization** – **34. Cost, Pricing, and Rates** – **35. Cost Estimating** – **36. Financial Reporting, Metrics** – **37. Business Strategy** – **38. Capture Planning, Proposal Process** – **39. Supplier Management** – **40. Industry Motivation, Incentives, Rewards** – **41. Negotiations**

3.2.2 NASA Academy of Program/Project and Engineering Leadership (APPEL)

The U.S. National Aeronautics and Space Administration (NASA) APPEL project provides a competency model to support the professional development of NASA's technical workforce (Table 3.2). This competency model integrates Project Management (PM) and SE. It includes three parts, 17 PM competencies, 17 SE competencies, and 14 shared competencies common to both disciplines [26].

NASA has defined 4 proficiency levels: **SE Proficiency Level I**-Technical Engineer/ Project Team Member, participates as a team member to gain an overall understanding of the stakeholder expectation definition and management process and to gain initial experience in the competency, and participates as a team member to gain an overall understanding of the product verification process and to gain initial experience in the competency. **SE Proficiency Level II**-Subsystem Lead, performs SE activities for a subsystem or simple project (e.g. no more than two simple internal/external interfaces, simpler contracting processes, smaller team/budget, shorter duration). **SE Proficiency Level III**-Project Systems Engineer and **SE Proficiency Level IV**-Chief Engineer. Note that we are mostly targeting the first two levels with this work.

Table 3.2: NASA Academy of Program/Project and Engineering Leadership (APPEL)

- **Competency Area: 1.0 Concepts and Architecture**
  - **1.1 Mission Needs Statement**
  - **1.2 System Environments**
  - **1.3 Trade Studies**
  - **1.4 System Architecture**
- **Competency Area: 2.0 System Design**
  - **2.1 Stakeholder Expectation Definition and Management**
  - **2.2 Technical Requirements Definition**
  - **2.3 Logical Decomposition**
  - **2.4 Design Solution Definition**
- **Competency Area: 3.0 Production, Product Transition, Operations**
  - **3.1 Product Implementation**
  - **3.2 Product Integration**
  - **3.3 Product Verification**
  - **3.4 Product Validation**
  - **3.5 Product Transition**
  - **3.6 Operations**
- **Competency Area: 4.0 Technical Management**
  - **4.1 Technical Planning**
  - **4.2 Requirements Management**
  - **4.3 Interface Management**
  - **4.4 Technical Risk Management**
  - **4.5 Configuration Management**
  - **4.6 Technical Data Management**
  - **4.7 Technical Assessment**
  - **4.8 Technical Decision Analysis**
- **Competency Area: 5.0 Project Management and Control**
  - **5.1 Acquisition Strategies and Procurement**
  - **5.2 Resource Management**
  - **5.3 Contract Management**
  - **5.4 Systems Engineering Management**
- **Competency Area: 6.0 NASA Internal and External Environments**
  - **6.1 Agency Structure, Mission, and Internal Goals**
  - **6.2 NASA PM/SE Procedures and Guidelines**
  - **6.3 External Relationships**
- **Competency Area: 7.0 Human Capital Management**
  - **7.1 Technical Staffing and Performance**
  - **7.2 Team Dynamics and Management**
- **Competency Area: 8.0 Security, Safety and Mission Assurance**
  - **8.1 Security**
  - **8.2 Safety and Mission Assurance**
- **Competency Area: 9.0 Professional and Leadership Development**
  - **9.1 Mentoring and Coaching**
  - **9.2 Communication**
  - **9.3 Leadership**
- **Competency Area: 10.0 Knowledge Management**
  - **10.1 Knowledge Capture and Transfer**

3.2.3 MITRE Institute Systems Engineering Competency Model

The MITRE systems engineering competency model has been created as a comprehensive one, in order to address a wide range of systems engineering jobs, by developing and assessing individuals and teams for that. This competency model made it very clear that is has been developed in a way that its details are unique to its developer's needs (i.e. MITRE's). However, it can be applied to other professionals doing systems engineering with the necessary tailoring effort.

It encompasses 36 competencies, grouped into five sections (competency areas): enterprise perspective, systems engineering lifecycle, systems engineering planning and management, systems engineering technical specialties, collaboration and individual characteristics, see Table 3.3. It has three levels of proficiency: foundational, intermediate, and expert, and therefore the key actions and behaviors are shown following three columns.

Table 3.3: MITRE Institute Systems Engineering Competency Model

- **1.0 Enterprise Perspectives**
  - **1.1 Comprehensive Viewpoints**
  - **1.2 Innovative Approaches**
  - **1.3 Foster Stakeholder Relationships**
- **2.0 Systems Engineering Lifecycle**
  - **2.1 Concept Definition**
  - **2.2 Requirements Engineering**
  - **2.3 Architecture**
  - **2.4 Systems Design and Development**
  - **2.5 Systems Integration**
  - **2.6 Test and Evaluation**
  - **2.7 Systems Implementation, O and M, and Transition**
- **3.0 Systems Engineering Planning and Management**
  - **3.1 Transformational Planning**
  - **3.2 Government Acquisition Support**
  - **3.3 Contractor Evaluation**
  - **3.4 Risk Management**
  - **3.5 Configuration Management**
  - **3.6 Integrated Logistics Support**
  - **3.7 QA and Measurement**
  - **3.8 Continuous Process Improvement**
- **4.0 Systems Engineering Technical Specialties**
  - **4.1 Cost/Benefit Analysis.**
  - **4.2 Human Centered Engineering (HCE).**
  - **4.3 Modeling and Simulation (M and S).**
  - **4.4 Security Engineering.**
  - **4.5 Reliability, Maintainability, and Availability (RMA).**
  - **4.6 Safety Engineering.**
  - **4.7 Software and Information Engineering.**
  - **4.8 Communications/Networking Engineering.**
  - **4.9 Collaborating with Technical Specialties.**
- **5.0 Collaboration and Individual Characteristics**
  - **5.1 Building Trust.**
  - **5.2 Building a Successful Team.**
  - **5.3 Communicating with Impact.**
  - **5.4 Persuasiveness and Influence.**
  - **5.5 Facilitating, Managing, and Championing Change.**
  - **5.6 High Quality Standards.**
  - **5.7 Results Orientation.**
  - **5.8 Adaptability.**
  - **5.9 Integrity.**

3.2.4 INCOSE UK competency Framework

The goal of the INCOSE UK Advisory Board (UKAB) for developing the SE Competency Framework (SECF) was *"to have a measurable set of competencies for systems engineering* *which will achieve national recognition and will be useful to the enterprises represented by the UKAB" [66]*

The INCOSE systems engineering competency framework is a document describing systems engineering competencies, to help people understand what is needed for individuals and teams doing systems engineering. These competencies should be tailored to meet the individuals and enterprises needs. They are based on different resources such as : The 15288 standard [4], the SE Body of Knowledge (SeBoK) [26], the INCOSE SE Handbook [2], and the NASA Handbook[36].

Systems engineering **competencies**, **basic skills and behaviors**, and **supporting techniques** are the main components of the framework, in addition to the **domain knowledge**, which is not detailed in the framework document, because it is difficult to produce a specific set of competencies for domain knowledge, and it is specific to each industrial field.

The SECF is generic in purpose [66] and intended to be used by:

- **Organizations:** combined with the terminology, processes and job roles that exist when the framework is applied, to:
    1. Identify organizations roles SE competencies, redefine roles, and create new roles.
    2. Adapt the processes and organization capabilities, and exploit Systems Engineering competencies;
    3. Measure and benchmark the state of Systems Engineering in the organization, and prioritize organization improvements and / or training programs, while identifying gaps in skill base.
    4. Provide a better understanding of Systems Engineering by explaining what people do when undertaking Systems Engineering.
- **Project or application:** to determine, on which part of the framework to focus, depending on what section of the lifecycle is addressed.
- **Individuals:** to assess their level of Systems Engineering competencies, and to plan ongoing professional development.
- **Academic institutions/Training Providers:** to develop educational programs in Systems Engineering, and to assess education or training programs, in both dimensions, its depth represented by the level of systems engineering involved competencies assessment, and its breadth by considering the number of competencies covered in any course.

However, as this framework cannot be specific at the same time for each of the previous cases, a degree of tailoring is necessary [66].

Characteristics of this framework [66]:

- Its competencies can be combined with other domain specific competencies and relevant skills and behaviors.
- It focuses on the competencies and skills that give better results when **linked with processes**, organizations and infrastructure capabilities (keep that in mind for next chapters).

- Subsets of this framework may be selected and used independently.
- To reach a higher competencies level, all the indicators for the levels below apply.
- The most important aspect of this framework is the fact that it has been supported by the guide to competency evaluation, published as a separate document [67]. This guide is designed to be a companion to the competency framework document, and gives guidance on how to evaluate people against the competencies included in the competency framework.
- Another very important fact is its independence regarding engineering domains and disciplines.

The framework competencies are grouped into three themes (the equivalent of Competency Areas in the two previous models): ***Systems Thinking:*** *contains the underpinning systems concepts and the system/super-system skills including the enterprise and technology environment.* ***Holistic Lifecycle View****: contains all the skills associated with the systems lifecycle from need identification, requirements through to operation and ultimately disposal.* ***Systems Engineering Management****: deals with the skills of choosing the appropriate lifecycle and the planning, monitoring and control of the systems engineering process* [66]. The competency framework defines only 21 separate competencies, within three different groups, [66]. See Table 3.4. Each competency is represented by a "competency table", and each competency table provides: a description of the competency, why it matters, and the effective indicators of knowledge and experience as (Awareness, Supervised Practitioner, Practitioner, and Expert) [66]. The systems engineering basic skills and behavior, along with the supporting techniques are represented each in a single table. It is almost the same structure for all SE competency models.

Table 3.4: INCOSE UK Systems Engineering Competency Framework

• **1.0 Systems Thinking**
  – **1.1 Systems Concepts**
  – **1.2 Super System Capability Issues**
  – **1.3 Enterprise and Technology Environment**
• **3.0 Systems Engineering Management**
  – **3.1 Concurrent Engineering**
  – **3.2 Enterprise Integration**
  – **3.3 Integration of Specialisms**
  – **3.4 Lifecycle Process Definition**
  – **3.5 Planning, Monitoring and Controlling**
	• **2.0 Holistic Lifecycle View**
	  – **2.1 Determine and Manage Stakeholder Requirements**
	  – **System Design:**
	    * **2.2 Architectural Design**
	    * **2.3 Concept Generation**
	    * **2.4 Design for . . .**
	    * **2.5 Functional Analysis**
	    * **2.6 Interface Management**
	    * **2.7 Maintaining Design Integrity**
	    * **2.8Modeling and Simulation**
	    * **2.9 Select Preferred Solution**
	    * **2.10 System Robustness**
	  – **2.11 Systems Integration and Verification**
	  – **2.12 Validation**
	  – **2.13 Transition to Operation**

3.3 Additional competency models

Since no consensus exists on a specific competency model, many additional SE competency models have been developed for specific needs or for specific organizations. The SECCM Model is one case of additional SE competency model and merges a number of competencies from eight different competency models including the previous ones.

The Systems Engineering Career Competency Model (SECCM) was developed by the systems engineering department at the Naval Postgraduate School. The categorization of the competencies within this model is based on the ENG competency model (it includes the same competencies and their categories). It is an extension of the ENG Model with additional KSAs from other models including NASA, MITRE, SPRDE, etc. Its current version (2013) has 3052 KSAs organized into the 41 ENG specific competencies [59].

Twelve systems engineering competency models have been surveyed in [68]. These models were used by the author to extract the systems engineers cognitive competencies that form the first dimension of his systems engineering cognitive competency model. Three other dimensions were used: role, proficiency level, and competency level.

3.4 Chapter Summary

The table illustrated by Figure 3.1 provides a side-by-side comparison of three SE competency frameworks/models and how they each address five primary competency vectors [5].

	Competency Model		
Competency Vector	**INCOSE (2010)**	**DAU SPRDE-SE/PSE (2010)**	**NASA (2009)**
Technical Processes	Core Competencies (Sys Thinking, Holistic Lifecycle View)	Analytical Competencies (Technical Processes)	System Design & Product Realization
Technical Mgt Processes	Core Competencies (SE Mgt)	Technical Mgt Competencies	Technical Management
Supporting Techniques	Supporting Techniques	Analytical Competencies (System Assurance, Software Engineering, Safety, RAM)	Security, Safety & Mission Assurance
Domain	Domain Knowledge	N/A	Internal & External Environments
Leadership/Personal Attributes	Basic Skills and Behaviors	Professional Competencies	Professional & Leadership Development
Note: *NASA also has Human Capital Mgt & Knowledge Mgt as competency areas*			

Figure 3.1: Comparison of INCOSE, DAU and NASA SE Competency[5]

While the INCOSE Framework was designed to be used by a large number of entities working on systems engineering, the three first competency models (ENG, MITRE, NASA) were developed respectively for the U.S. Department of Defense, the U.S. government institutions following the government view of systems engineering, and for NASA projects and systems engineering. That does not mean that they cannot be used in other entities, industries or domains, but they have to be tailored to the context of application considering the institution specific competencies removal or adaptation. This is the way how the INCOSE framework should be applied too, even if it does not include domain specific competencies. It should be augmented by some of them depending on the context of use. So at this level Incose, neither any other model is taking the advantage.

An advantage of the INCOSE framework over the three first competency models is that, the three were developed by a single entity with a focus on the needs of one organization, which make it very efficient and meets the challenges of the specific organization. However the INCOSE framework was developed by a working group with members from different entities/industries, and based on documents and resources from several institutions including NASA SE hand book and systems engineering standards, what makes it very adaptable to other industries and fields of application.

Also, unlike other models, the INCOSE Framework can be distinguished from the other models, due to the fact that it provides individuals, organizations, and education/training entities with additional information. It shows in fact, how the framework should be tailored and applied, and provides a framework for competency evaluation, that can be used to assess students in systems engineering courses and systems engineers competencies. It also provides additional data about skills to be found in systems engineers, focusing on soft skill, along with tips and supporting techniques that can be used to acquire these different competencies and skills of the framework.

At the first glance, INCOSE competency framework seems to be the most suited model for systems engineering education in regard of its form and content. However, despite his presentation form, including much content than other models (suggested techniques, the competencies evaluation guides...etc), after studying INCOSE Model and its 21st competencies in depth, we consider that it still needs some improvements and is not the best suited model for now. We could state that INCOSE framework is less understandable than other models, it lacks depth and breadth coverage and explanation, in term of its different competencies. Its competencies does not cover all systems engineering needed competencies, and some of the proposed ones are not explicit enough. And most importantly, The framework should have consider elucidating and dividing competencies into more trivial sub-competencies. This is important due to the fact that sub competencies (such as the competency elements in NASA model), might be more simple to understand and link to SE processes activities. According to [60], this framework is created based on the knowledge systems engineers have in UK, rather than on the knowledge systems engineering need.

We consider that INCOSE would have done better considering the use of NASA competency model structure and tailored content, while adding the competencies evaluation guide and the suggested techniques and methods to acquire each competency and competency elements. Finally, note that even if we are trying to find which one might be the best choice to manage students competencies and help elaborating efficient learning scenarios, we will not advocate a specific model for all learning situations. Our proposed approach will be a competency model independent, so each organization using it will have the entire liberty to select what to use, and maybe, it will define its own model.

4
Systems Engineering Education

4.1 Introduction

The need for effective systems engineering education is evident. Systems Engineering Education is often regarded as an extension to regular Engineering Education, typically taught to graduate students along with interdisciplinary studies, and sometimes included in undergraduate university programs [11]. INCOSE has formulated a policy statement that emphasizes the importance of Systems Engineering and its impact on regular engineering disciplines: "*INCOSE believes strongly that a systems perspective and the fundamental principles of systems engineering have an important role in the education of all engineers regardless of their specialty. This will strengthen the general recognition that most of today's engineering tasks are performed in multidisciplinary teams, and degree granting programs in systems engineering must be encouraged and supported*" [69]. Burke et al. from the U.S. Federal Aviation Administration [70] stated that systems engineering curriculum play a major role in developing better systems engineering processes, as they provide the needed intellectual source material.

Engineering schools do not always offer independent Systems Engineering courses or programs, in other terms, they don't always offer a theoretical or practical teaching about the systems engineering discipline itself, even if they may use its principles within other domain specific design and engineering courses. Systems Engineering may be taught in the scope of a single module, under the label of product development, product design, design engineering or design thinking. Design Thinking *"reflects the complex processes of inquiry and learning that designers perform in a systems context, making decisions as they proceed, often working collaboratively on teams in a social process, and "speaking" several languages with each other (and to themselves)"* Dym et al. [9]. Other schools have created dedicated programs and curriculum for Systems Design, Systems Engineering and closely related areas [71]. We can therefore distinguish two approaches of Systems Engineering Education: Systems-centric and Domain-centric. *"Systems-centric programs treat systems engineering as a separate discipline and most of the courses are taught focusing on systems engineering principles and practice. While, Domain-centric programs offer systems engineering as an option that can be exercised with another major field in engineering*" [14].

The American Society for Engineering Education (ASEE) suggests including the following points, in any engineering discipline curriculum [72]. When we analyze these points, it appears that they recommend the integration of systems engineering principles and knowledge within any engineering discipline:

- *Team skills, and collaborative, active learning.*
- *Communication skills.*
- *A systems perspective.*
- *An understanding and appreciation of diversity.*
- *Appreciation of different cultures and business practices, and understanding that engineering practice is now global.*
- *Integration of knowledge throughout the curriculum a multidisciplinary perspective.*
- *Commitment to quality, timeliness, continuous improvement.*
- *Undergraduate research and engineering work experience.*

- *Understanding of social, economic, and environmental impact of engineering decisions.*

- *Ethics.*

For Sage [73], a major goal of Systems Engineering Education should be to acquire the abilities relative to each of the 19 focus areas for Systems Engineering identified by the Systems Engineering Capability Model (SECM) [74]. The author presents Systems Engineering Knowledge as a composition of three aspects:

- **Knowledge Perspective** which allows forecasting the need for innovation, including innovation principles to identify the appropriate systems planning and marketing directions.

- **Knowledge Principles** as formal problem solving approaches, which are usually linked to fundamental knowledge needed for research and development.

- **Knowledge Practices** representing the accumulated experience that has led to standard operating policies for well-structured problem solving.

What does it take to produce good systems engineers?, which are the competencies of these systems engineers? What are the characteristics of an efficient SE educational environment? What has been done so far on this subject?, These are the questions we will address through this chapter.

4.2 Criteria for successful systems engineers

We first consider what a systems engineer needs to know, and what he needs to be able of. We start by the following criteria for successful systems engineers within Mitre, as detailed by the Mitre Systems Engineering Guide [43], and excerpted from the MITRE Systems Engineering Competency Model[63]. According to Mitre, a systems engineer should be able to:

- *Define the sponsor's and customer's problem or opportunity from a comprehensive, integrated perspective.*

- *Apply systems thinking to create strategies, anticipate problems, and provide short-term and long-term solutions.*

- *Adapt to change and uncertainty in the project and program environment, and assist the sponsor, customer, and other stakeholders in adapting to these.*

- *Propose a comprehensive, integrated solution or approach that:*
  - *Contribute to achieving the sponsor's, customer's and other stakeholders' strategic mission objectives in a changing environment.*
  - *Can be feasibly implemented within the sponsor's and customer's political, organizational, operational, economic, and technical context.*
  - *Address interoperability and integration challenges across organizations.*
  - *Shape enterprise evolution through innovation.*

- *Cultivate partnerships with our sponsors and customers to work in the public interest.*

- *Bring their own and others' expertise to provide sound, objective evidence and advice that influences the decisions of our sponsors, customers, and other stakeholders.*

Frank [68] on the other hand, proposed 20 cognitive competencies of successful systems engineers, related to systems thinking. They have been extracted from twelve different competency models in systems engineering:

1. Understand the whole system and see the big picture; think broadly; have grand visions; have a generalist's perspective; have holistic view; think strategically;
2. Be able to work consistently at an abstract level;
3. Understand interconnections; closed-loop thinking; recognize patterns;
4. Understand system synergy (emergent properties);
5. Understand the system from multiple perspectives;
6. Think creatively; think out of the box; be able to make good associations of ideas, to seek multiple solutions; think laterally; think divergently;
7. Understand systems without getting stuck on details;
8. Tolerate ambiguity and uncertainty; adapt to change;
9. Understand the implications of proposed change;
10. Understand a new system/concept immediately upon presentation;
11. Understand analogies and parallelism between systems;
12. Understand limits to growth;
13. Ask good (the right) questions; know when to ask; maintain healthy skepticism
14. Be innovators, originators, promoters, initiators, curious;
15. Be able to define boundaries;
16. Be open minded; open to new ideas
17. Be able to take into consideration non-engineering factors;
18. "See" the technical/engineering future (vision); have a sense of faith or vision; anticipate problems; see future trends;
19. Think objectively
20. Think critically

Prior to that [75], he proposed a framework of ten cognitive characteristics, eleven abilities, and ten behavioral competencies for engineers with high capacity for engineering systems thinking (capacity of becoming systems good engineers).

More recently, Walden [76] proposed three key critical principles, in order to be a successful systems engineer. He started by the main characteristics of systems thinking, which are the holistic approach and understanding of relationships, and suggested three elements that help developing these skills:

1. **Look Up:** to fully understand the higher levels we are part of.
2. **Look Out:** to understand the critical peer-level interfaces and relationships.
3. **Look Down:** to understand the lower levels that are part of your responsibility .

4.3 Relevant characteristics for effective SE educational environments

In addition to the following statements and recommendations, the previous criteria for successful systems engineers are the starting point for defining characteristics of effective educational environments for systems engineering.

For Jackson [77], one of the main reasons slowing down the dissemination of the systems engineering process to non-engineer and non-systems engineers, is the focus on complexity: the complexity of the problems, but sometimes the complexity of the process, too. For the author, systems engineering should be described as a process to understand needs, and to design elegant and harmonious solutions for these needs. He advises to stop describing it as as the process used to design complex systems, because *"we do not set out to make our systems complex. It is the needs that are complex"*[77]. The author outlined ten requirements to consider for designing a curriculum for educating the non-engineer in the systems engineering process. Then he used these requirements to design his curriculum, named "Getting Design Right".

Asbjornsen and Hamann [78] provided an overview of systems theory and systems engineering methodology in order to design a pedagogical concept for both engineering education in general and systems engineering education in particular. They argue that the initiative to take up Systems Engineering at a university level has come from industry and not from academia. By examining the industrial motivations, the authors identified a list of learning targets for high-quality systems engineering education, such as:

- Qualitative and Quantitative Knowledge
- Systems engineering ability and insight, and learning ability.
- Human factors and, loyalty and individual responsibility
- Global and environmental concerns

According to Muller [14], systems engineering education differs from traditional mono-disciplinary engineering courses, since the training needs to focus more on skills and less on transferable facts. The author gives a set of recommendations to consider for a good Systems Engineering Education program, including interaction with students, soft skill development, media use and student feedback. Dym [15] believes that "*a good Engineering Education is about process, about learning how to think like an engineer; it is much more than a prescription of content*".

Dym et al. [9] recommend the following three activities for a powerful learning environment for Systems Engineering and similar disciplines:

- Instrumenting the learning process to obtain quantitative and qualitative data that support metrics consistent with quality control.
- Teaching Design Engineering and other disciplines such as Systems Engineering across geographically dispersed, culturally diverse, international networks

- Engage design coaches to help manage the contextualization of engineering design theory and practice.

In a broader context, Herrington and Kervin [79] specified nine main characteristics that any learning environment, technology-based or not, should feature:

- Provide authentic context that reflect the way the knowledge will be used in real life
- Provide authentic activities
- Provide access to expert performances and the modeling of processes
- Provide multiple roles and perspectives
- Support collaborative construction of knowledge
- Promote reflection to enable abstractions to be formed
- Promote articulation to enable tacit knowledge to be made explicit
- Provide coaching by the teacher at critical times, and scaffolding and fading of teacher support
- Provide for authentic, integrated assessment of learning within the tasks

Hyer [38] recommended the following elements for an effective approach to systems engineering training:

- They should be built on a solid and pertinent undergraduate education and a certain amount of professional experience.
- They should include a combination of classroom education and professional experience.
- They should develop and improve a systems engineer's instinct for system level thinking and communication.
- They should be continuous throughout an individual's career and be monitored and adjusted according to opportunities.

Pfar et al. proposed four areas for enhancing existing education courses in systems engineering [80]:

- Deep thinking about about systems in their context or environment, and understanding the "big picture" of a system.
- Situational leadership abilities development, in regard to multiple levels decision making.
- More education and experiential learning opportunities about how to think about systems and system interactions within and across life cycle phases and the ability to anticipate future scenarios.
- Take advantage of the benefits of collaborative systems thinking, and enable training for entire teams.

Team coordination for systems engineering has also been highlighted by Avnet [81]. He concluded for his research that team members learn most from each other when they work on difficult or unfamiliar problems, i.e. when they applying systems engineering, as this approach is mainly applied for complex or unfamiliar problem resolution.

Hyer [38] considers that, for a most effective impact, the training in systems engineering must be planned, monitored, and occasionally adjusted. For this purpose, he suggests to engages supervisors, teachers, or professional mentors in the training process. According to [82], one of the reasons systems engineering is poorly taught, is the lack of good teaching, as well as the lack of good delivery methods that bear little relationship to the environment where modern engineers collaborate and make decisions. Verma [83] stated that, given the iterative, multi-discipline, and complex nature of a systems engineering process, computer-based course-ware can be an invaluable aid to instruction.

In Table 4.1, we illustrate a synthesis of the main characteristics for a successful systems engineering educational environment. These characteristics have been extracted from the previous statements, along with the characteristics of successful systems engineers, and the requirements for efficient Project Based learning implementation as we will see in the next section.

Table 4.1: Classification of main characteristics of an effective systems engineering educational environment

SE Educational Environment Characteristics
1 - Students, Educators, and their Organization
Educators should point out that systems engineering exists not to make complex systems, but to find solution for complex needs.
Students should be aware of the fact that systems engineering is characterized by a high level of uncertainty about the solution and the process by which it will be achieved.
For introductory systems engineering education experiences, no industrial experience neither a degree are required. However, to advance students knowledge, abilities and skills in systems engineering, they should at least have an undergraduate degree, or a first industrial experience in engineering systems.
Educators and/or industrial experts are wanted as coaches, to assist and to help students contextualizing theory and practice, within the systems engineering education experience.
When students are engineering a system during a learning project, they should understand that they are part of a team as systems engineering resources, and they might be considered as a part of the system being engineered itself.
The organization should ensure that its educators skills and abilities are up-to-date and meet the practitioners needs. Educators should be in regular contact with industry and organizations dealing with systems engineering.
Students should be able to understand and manage their roles, and be aware that their knowledge should be constructed collaboratively. They need to understand that team members will learn most from each other when working on difficult or unfamiliar problems and applying systems engineering.
2 - Learning Process and Pedagogical Approaches
Continued on next page

Table 4.1 – continued from previous page

SE Educational Environment Characteristics
The learning process should be adaptive and take into consideration the learning goals (competencies and depth level), student competencies, industry/other organizations requirements, and educators/experts availabilities.
The learning process should guide students through the context of their engineering actions (higher, peer and lower levels of the system-of-interest), in addition to the consideration of global and environmental concerns.
The learning process should make students understand and practice the ideas of wholeness and big picture of a system, as well as collaborative systems thinking, mainly by proposing suitable systems to be engineered, within suitable contexts.
The learning process should make students understand and practice the idea of interrelationships between subsystems and between the system and external systems and the environment.
The learning process should be continuous, monitorable and adjustable, through an individual's career or curriculum.
The learning process should allow for self-learning, to improve students learning abilities.
The education experience should contain both classroom courses and use-case projects (replacing on-the-hand jobs). However, it should focus more on skills acquisition and development. Pedagogical resources and technical knowledge acquisition should be considered as means to better engineer the system and to better perform systems lifecycle processes.
3 - Supporting Environment, Tools, and Resources
The educational environment should provide educators and students by means that allow them to tailor Systems Engineering processes and to understand the importance of processes tailoring.
It should provide an advanced implementation of the Project Based Learning approach, in a geo-distributed context that facilitates the interaction between distant students in the same team, and improves the implication of coaches during the learning process (educators or industrial experts).
It needs to make students engineering the requested system in a reality-like context, emphasizing communication, collaboration and soft skills acquisition, within geographically dispersed teams, with the introduction of cultural and international dimensions.
It should allow students to work on real-life projects, proposed by industrial partners, to substitute the need of on-the-job experiences.
It should be adaptable to the needs of advanced education in systems engineering.
It should ensure students competencies management, by updating students profiles after each SE training experience, and making plans (proposing learning sessions), that address the missed competencies.
The learning environment should be able to be instrumented, in order to get metrics, that will serve for students evaluation and learning processes improvement.
Students assessment should address different points: evaluation of the final results (the quality of the design of the engineered system), the systems engineering processes execution quality, and the acquired skills and knowledge. It should address both individual and collective assessment.

Continued on next page

Table 4.1 – continued from previous page

SE Educational Environment Characteristics
Educators should be aware of available modern tools and teaching materials and should be supported by their organization to acquire and use them within the teaching experience.
Student should be provided by tools within the educational environment, in order to sub-divide the requested system (the structural architecture), and manage sub-systems and system elements as systems to apply on them the systems engineering required lifecycle processes.
Educators and coaches or industrials providing the system to be engineered, should be able to see what students are doing at any time, in order to correct them if they did something wrong, or to adapt their requirements and validate them with students.

4.4 Project Based Learning and SE educational environments

4.4.1 Inductive Teaching

Traditional *deductive teaching* of Engineering and Science starts with theory and progresses to their applications. The educator typically introduces a topic and explains its general principles, derives mathematical models from these principles, presents illustrative applications, makes students practice on similar applications, and finally tests their acquired knowledge in an exam [84].

In contrast to the deductive teaching approach, *inductive teaching* is about letting and helping students discover and learn theories only after the need to know them. This process is usually started by the educator presenting specific observations, case studies or problems. Some examples of inductive methods are: problem and project-based learning as two different methods, discovery learning, just-in-time teaching, etc. In the context of this thesis we are interested in Project-Based Learning, as defined by Prince et al.: *"Project-based learning begins with an assignment to carry out one or more tasks that lead to the production of a final product-a design, a model, a device or a computer simulation. The culmination of the project is normally a written and/or oral report summarizing the procedure used to produce the product and presenting the outcome"* [84].

4.4.2 Project Based Learning

In addition to the its characteristics, it is a convention that SE educational environments should be built on top of a Project Based Learning (PBL) model. This is stated for example by Turnquist et al. in [85]. The authors affirmed that an integrated part of systems engineering educational programs resides in the use of case studies and project experiences, which allows learning the set of systems engineering skills by practice. Frank et al. emphasized in [86] the benefits of PBL for systems engineering and systems thinking capabilities development, because of their exposure to the processes, design procedures, and some underlying systems engineering principles. They also stated that PBL within an educational environment allows student to stimulate the professional work of a systems engineer.

Teamwork is one of the main components of PBL, and *"improving a student engineer's ability to work effectively and collaboratively within a team is an objective shared in all engineering programs, but it is a critical objective in systems engineering programs"* [87]

According to Dym et al. [9], "*the currently most-favored pedagogical model for teaching Design is Project Based Learning*". Despite the differences between design and systems thinking (a core aspect of the Systems Engineering discipline) [88], both Engineering Design and Systems Engineering mostly deal with processes and skills, and not with transferable and fundamental knowledge. Engineering Design is defined as *"a systematic, intelligent process in which designers generate, evaluate, and specify concepts for devices, systems, or processes whose form and function achieve clients' objectives or users' needs while satisfying a specified set of constraints."* [9]. Therefore, we can assume that Project-Based Learning (PBL) is actually the most-favored pedagogical model for both Engineering Design and Systems Engineering Education. According to Khalaf et al. [89], the nature of these disciplines is in inherent alignment with the PBL pedagogy, so PBL is recommended especially for developing analytical and problem-solving skills needed to address multidisciplinary and complex engineering problems.

Most of Systems Engineering Education programs that will be presented later, make use of PBL as their proposed curriculum.

Current PBL Challenges

PBL seems to be the most adequate model to teach general engineering, and especially for Systems Engineering. However, this does not mean that PBL is a perfect model that does not suffer from some limits, or that does not need adjustments to be more efficient. There still are a number of open research questions regarding PBL. Some of them have been identified by Dym et al. [9], such as:

- *What are the best proportions of problems, projects, teamwork, technology, and reality for a given state of student development? In other words, how authentic should PBL experiences be compared to industry design experiences? Some work has begun to emerge in this area, but the answers are not yet definitive.*
- *How do the proportions change with regard to the context of different engineering disciplines and institutional missions?*
- *How should multidisciplinary design-learning teams be managed?*
- *Can a pedagogic framework developed for co-located learning teams be distributed in time and place? If so, how?*
- *How can students be authentically evaluated and graded in design courses with regard to, for example,*
    - *the quality of the design produced vs. the quality of the process demonstrated; and*
    - *individual cognitive development vs. collective team development?*

4.5 Systems Engineering Education - Current Practice

This section illustrates some important currently proposed approaches, curriculum and programs, and proposals for Systems Engineering Education around the world. Interestingly, no common pedagogical model can be identified. Each institution has its own teaching format varying from one-week crash courses to multiple-year programs. Some of these approaches, curriculum and programs, and proposals are developed for/by academic institutions, industrials, or by both of them. Depending on their nature and goals, they aim to develop systems capabilities within industrial systems engineers practitioners, to provide the fundamentals of systems engineering for non systems engineers, or to produce fresh systems engineers using different means: theoretical background, capstone and cornerstone projects, and on-the-job experiences. We classify the different approaches depending on their organization in time, into long, short, and mid-term programs.

4.5.1 Long-term programs

Masters and undergraduate Programs

To meet the needs of modern engineers, and to address the innovation crisis in the U.S, Craig and Voglewede from Marquette University, Milwaukee, presented in 2010 a Master program of Engineering in Mechatronics [90]. Their approach focuses on finding a balance between academic rigor and best practice. The program includes 12 one-credit key modules covering fundamental engineering knowledge such as mathematics, physics, mechanics and electronics, along with Systems Engineering related knowledge such as control, analysis tools, systems modeling and design. The lessons learned in these modules are applied to four three-credit case study courses: *transportation system, home and office system, energy system, and automation system.* During these courses students get comfortable with the most important practices in Systems Engineering, including user and problem understanding, design, implementation, integration, trade-offs and optimization. Finally, a six-credit on-site experience allows the students to put it all together in a genuine industrial context.

Another significant example is the Master program created by the Johns Hopkins University (JHU) Whiting School of Engineering. In 2011, it was considered as the largest systems engineering program of the United States in terms of enrollment [91]. The program balances theory and practice, and offers different learning methods combining in-person classes, online classes, and industry partnerships. It also includes a challenging capstone Systems Engineering Project. According to the authors, many students who followed the JHU program, pursued doctoral studies in Systems Engineering at other universities such as George Washington University, George Mason University, Stevens Institute of Technology, University of Virginia, and Old Dominion University.

Also, the department of systems engineering within the United States Military Academy offers an undergraduate degree in systems engineering [92]. It includes [92] a one year long design experience, where teams of students from the program work on a real world problem. The goal of this capstone project is to give systems engineering Majors the opportunity to work on real-world project for real-world clients, which are relevant to the Army and to the Academy.

The Systems Engineering and Evaluation center (SEEC), within the university of South Australia proposed an approach for systems engineering training based on both education and work experience [93]. The proposed approach for developing systems engineers includes four phases:

1. Undergraduate phase: Through an introductory systems engineering subject offered to the final year undergraduates. **It should be built upon one of the systems engineering standards**.
2. Induction phase: Introducing newly recruited engineers to the organizations **engineering processes**.
3. Training phase: To gain experience in the organization, its modules should follow the professional body, or at least follow the main **systems engineering process steps**.
4. Professional working phase: The normal working life of a design engineer.

Turnquist et. al presented in [85] a systems engineering master program developed in collaboration between Cornell University and Lockheed Martin Federal Systems (LMFS). This collaboration emphasizes the importance of building teamwork and systems engineering skills through project experience. The resulted curriculum from this partnership focus on eight main elements, closely related to the elements identified by INCOSE as core material for systems engineering programs [85] :

1. Requirements analysis.
2. Functional, behavioral, and structural analysis.
3. Simulation, optimization and multi-criteria decision making.
4. Risk assessment and management.
5. Lifecycle analysis.
6. Teamwork and project management.
7. Information flow in systems.
8. Feedback control.

The underlying education philosophy for this master's degree education is the main contribution of the academic partner, while the industrial one brought the application part of this philosophy through use cases, especially when this collaboration gives high consideration to the project aspect.

The Fachhochschule München has successfully introduced a one-year graduate course for Systems Engineering [94]. Candidates can enroll at this one year course only if they already have an academic degree before starting the course. The course is divided into two parts, the first one concerns the principles of systems engineering and systems engineering knowledge, methods and tools of system development, and methods and tools of system management. The second part is related to: engineering project within an industrial company, courses explaining successfully completed systems engineering projects, and a course in business English. Additional details about these subjects and their organization can be found at [94].

Facing the diversity of systems engineering education initiatives, and considering the absence of a community-accepted recommendation or guidance on what to teach in systems engineering graduate program, another very significant effort was performed by

Stevens Institute of Technology in collaboration with other systems engineering institutions like INCOSE. This effort is represented by the BKCASE international project "Body of Knowledge and Curriculum to Advance Systems Engineering". It resulted in the elaboration of a "Graduate Reference Curriculum for Systems Engineering (GRCSE)". Version 1.1 is now available for public [95]. GRCSE provides a reference curriculum for professional master's degree in systems engineering.

The goals of writing GRCSE is to assist in designing new systems engineering graduate programs and reviewing existing ones. In other terms, it is a tool to support development, maintenance, update, and selection of master's programs in SE to meet particular needs.

GRCSE intends to [95]:

- Improve existing SE graduate programs
- Assist in the development of new SE master's programs.
- Guide the deliberations of strategic advisory boards, to assist universities in appropriate program design.
- Support increased enrollment in SE programs
- Support prospective students and employers in gauging the suitability of a particular program for their individual purposes.
- Assist engineering educators in general to appreciate the distinctive knowledge and perspectives of systems engineering.

The second valuable product of the BKCASE project is the "Systems Engineering Body of Knowledge" [26], available as a Wiki, and providing a widely accepted, community based, and regularly updated baseline of SE knowledge.

Professional curriculum development programs

The U.S. Federal Aviation Administration (FAA) designed a systems engineering curriculum for both human capital and process improvement [70]. The curriculum focused on knowledge base and behavioral skills and competencies, which have been defined before proposing the curriculum. The proposed systems engineering curriculum has the following five features:

1. It considers both behavioral competencies and technical/organizational knowledge and skills.
2. It covers educating people on generic systems engineering processes and techniques, as well as on FAA-specific ones.
3. It make use of professional resources as ANSI/EIA-632 standard systems engineering processes and INCOSE resources and recommendations.
4. It ties directly to the normal expected progression of systems engineers at the FAA.
5. It does not focus on the domain specific technical knowledge, as it will be acquired separately (through previous professional education or other FAA curricula).

   In term of content, the curriculum covers 15 major knowledge areas which are detailed within the paper [70].

Another example of graduate systems engineering education for professional development is the Air Force Institute of Technology program [96]. It is mostly dedicated to Air Force candidates (military officers), coming generally to the course with four to seven years after their undergraduate degree with significant professional experience. The curriculum is built upon four principles: *the core, the breadth, the depth, and the design.* The *Core* consists of a course in system design and additional supporting courses, it provides the framework for systems engineering processes and a number of basic necessary tools. Concerning the *breadth*, students take specialization sequences in a systems engineering related area (operations research, optimization, etc.). The *Depth* consists of making students take an advanced sequence in a traditional engineering discipline. Finally, the program includes a group design projects, the *Design*, so that students can demonstrate entry-level mastery of systems engineering, while working on real-world complex system and applying systems engineering. An additional capstone design course within the Air Force Academy, regarding the integration of systems engineering design principles in a capstone design course is presented by Newcamp in [97].

In addition to this previously cited Master program, built with Cornell University, Lockheed Martin created a program to develop a senior level systems engineers within the company [98]. The proposed program was built upon five elements:

- Documented systems engineering methodology: in the form of a cookbook style, to make the reader quickly understand how systems development are approached within the company. This methodology is tailored at the beginning of a job by the project manager, and the document is reviewed. In other terms, the methodology provides an excellent road map of *"what"* to do to develop a system.

- Process documents: starting with the most common processes, a set of process documents are developed to describe *"wow"* to perform common systems engineering functions. The process is presented with a flow chart outlining the process at a high level with associated an text that explains how to perform each step of the process.

- Personal development process: The program is supported by a 17-step process, developed as a guide for producing an "Employee Development Plan". The development plan documents the employee's skills and competencies, career goals, and plans to obtain the desired skills and training.

- Training plan: By determining that training plans should be built and planned on an individual basis, in order to integrate all employees in a common core course, and individual's specialized training. The training plan is documented as a part of the employee's development plan.

- Mentoring: Implementing team mentoring by the section supervisor on a weekly lunch time basis, addressing both social and technical topics. A senior employee mentoring junior employee plan is recommended.

The French engineering school, ENSTA ParisTech, at the request of the DGA (General Direction of Armament), built an advanced systems engineering course, entitled -FAIS- [99]. It comprises 256 hours delivered on 2 days per week format and is addressed to technical experienced engineers aiming to learn about systems engineering. The course is open to other companies than the DGA to support the collaborative work in systems engineering. The course validation is done by making the participants passing the Incose ASEP certification exam, and preparing a dissertation to be presented in front of a jury. A multi sensors observation system was used as case study to support the theoretical principles. The course used a digital environment to enhance communication and collaboration between students and with teachers, based on Alfresco (electronic document management system). The used computers were configured with the needed tools (Doors, Mega, etc.).

Sheard and Swayhoover [100] proposed a plan to eight steps that structure a professional systems engineering development program within an organization. This plan considers both selection of candidates with a high potential to become systems engineers and training them. The eight elements are:

1. Define characteristics for good systems engineers within the organization.
2. Identify candidates.
3. Rank candidates against the the previous characteristics.
4. Select participants for the development program
5. Categorize the tasks that the students currently perform in term of roles.
6. Provide immediate development opportunities in these the roles currently being performed.
7. Design structured development plans for each student for potential growth roles.
8. Ensure that the development program is following the plan by monitoring the progress of the student.

A similar five-step plan was introduced by Davidz and Maier in [101]. It focuses on developing group capabilities instead of individual's capabilities. These steps are: Framing the problem and desired capability, developing strategy, aligning all levels, improving continuously, and executing the plan.

Armstrong and Wade emphasize the impact of learning systems engineering by teaching it, for developing professional systems engineers expertise. This is mainly done by the necessity to understand the teaching material and further study the subject, to be able to teach it and to answer the students questions [102].

California Institute of Technology launched in 2004 a Systems Engineering Advancement plan (SEA), in order to advance the practice of systems engineering withing the Jet Propulsion Laboratory (JPL) [103]. The plan focuses on three parts: People, Process, and Technology. The people element supports the recruiting, selection, and development of systems engineering via different strategies, including On-The-Job training (OJT), which represents the heart of this elements. According to the authors, if you want good systems engineers, sometimes you have to grow your own. The goal of the SEA OJT program is to increase the number of highly trained systems engineers at the Lab, by selecting and developing the capabilities of ten systems engineers each years for five years. The program is based on a specific competency model, including Technical Knowledge, Process Knowledge, and Personal Behavior components. For more details about how the SEA OJT works, please check Jansma and Derro's work in [103].

In order to shorten the time it takes a systems engineering to attain senior level, the Systems Engineering Experience Accelerator research project, has been initiated by Wade et al. [104] within Stevens Institute of Technology. The project goal is to create a learning and content development environment based on an open architecture. The Experience Accelerator allows the learner to deepen his/her knowledge and skills according to the competency model developed for that reason [105]. For detailed information about the current state of the Experience Accelerator, your can read the most recent Technical Report of the project [**?**].

Academy-Industry collaboration

Paris Higher Institute of Mechanics (Supméca), leader of the Placis Project, has created a Systems Engineering Education program under the same name "PLACIS" [106]. It aims to train engineers in a new format, asking students from different engineering schools, different countries and different disciplines, to work collaboratively on an international and multidisciplinary project. The students use the most recent engineering tools and technologies, including Catia V6, SysML, Abaqus, WebEx and Sharepoint.

Depending on the configuration of the program, it addresses both Master and Bachelor students. In the case of Masters students, the projects are generally carried out during one semester with the following course of events:

- A multidisciplinary project is proposed or re-conducted by a company.
- The project is approved by the Industrial and Academic partners.
- Multidisciplinary student teams from different universities are formed (6-10 students per group).
- A kick-off meeting with all involved persons is organized (in-person or video conference).
- The students work on the project by distant collaboration. They are coached by teachers and industrial tutors.
- Teachers and tutors assess the students' results and performance (reports, models, behavior, final presentation, etc.)

Delft University of Technology has an international post-graduate program for experienced people seeking expertise in space systems end business engineering, called "the SpaceTech program" [107]. The program combines space systems engineering with business engineering and marketing. This program is taught during five two-weeks intense sessions of actual classroom instructions (about 50 hours/week), spread over ten months. It aims to let student staying at their jobs, while acquiring a Master of Space Systems Engineering degree. The five session topics are:

- Space Mission Analysis and Design
- Systems Engineering Principles and Tools
- Space Applications (communications, earth observation, navigation)
- Business Engineering
- Interpersonal Skills including team building and team management.

The agreement and technical processes of the program are related to the ISO/IEC 15288:2008 Standard [50]. In addition to the lectures and exercises of the five previous sessions, the most important component is the Central Case Project (CCP). During the CCP, participants work intensively together between the sessions to exercise space systems engineering fundamentals with marketing and business engineering tools, to create a commercially viable virtual business.

These are not the first courses that collaborate with industrials and work with PBL. Before 1992, the University of Virginia had a similar senior capstone design course in systems engineering. It differs from the previous one in the fact that this course involved students in real design problems by working with real clients. [108]. The goals of this course were:

- To combine the elements of systems engineering theory and practice.
- To conduct design and experimental work such as that expected by practicing systems engineers.
- To give the students a hands-on experience on a real systems design problem.

It focused on several critical domains of knowledge: systems engineering, systems and decision science, systems design methodologies, and systems management.

4.5.2 Mid-Term Programs

Student challenges

AFIS, the French chapter of INCOSE, organize since 2006 a yearly student competition for robot design, called RobAFIS [109]. Around 10 student teams from French universities and engineering schools, inexperienced in Systems Engineering, participate each year in this competition.

Each team can consult a Systems Engineering teacher, and they can also question other AFIS experts. The roadmap starts about 8 months before the final stage competition, when AFIS communicates the general schedule, the regulations, specifications, and a reference development document. Three months before the final stage, the teams register and receive a LEGO Mindstorms Robotics kit in order to physically implement their solution. Fifteen days before the final stage, the teams send their development document to Systems Engineering experts for evaluation. The competition concludes by a final stage where all teams meet and operationally validate their works, along with project and configuration audits. Few weeks after the competition, the students receive a detailed debrief regarding their work.

Few-weeks Projects

In 2004, Bonnema et al. presented a solution to introduce Systems Engineering to third-year students in Industrial Design Engineering at the University of Twente [110]. The SAS project (Sensors, Actuators and Systems) applies Systems Engineering tools and techniques in a concrete situation. More specifically, the project allowed students to learn the basics and goals of Systems Engineering, and to keep an overview in a complex design project. The students worked in large groups of 12-14 persons in a project-based learning approach, without a tutor except for the possibility to discuss with some specific staff members. They were provided with lectures on a selected set of subjects on Systems Engineering, which represents 34 credit hours of the entire 140 credit hours of the SAS project, together with 53 hours for Sensors and Actuators, and 53 hours for the Assignment. The main study material was the 2.0 edition of the INCOSE handbook, the "Systems Engineering and Analysis" book [35] and the Introduction to Systems Engineering book [7].

There was no planning given to the students, except for two milestones: *Customer requirements, systems requirements, system concept, and sub systems*, that needed to be done during the first three weeks of the project, and *Sub-system design, plan for system integration and test, final system design*, scheduled for the following fours weeks. In 2004, the students were asked to design an intelligent climate-control system for houses, and in 2005 an intelligent car. They were evaluated based on poster sessions, a Sensors and Actuators exam, and a short essay on the application of Systems Engineering methods in their project.

4.5.3 Short-Term Programs

Short Introductory Courses

To introduce non-engineers and non-systems engineers to the process of systems engineering, Jackson [77] outlined the problem of considering systems engineering as complex systems definition solution instead of a solution to define simpler solution to complex problems. This was also highlighted by Grange [111]: "*Systems Engineering is concerned with providing a balanced, well defined, achievable solution to a complex, often abstract problem. The starting point for Systems Engineering is a high level of uncertainty about the solution and the process by which it will be achieved*". Jackson proposed the "Getting Design Right" curriculum to introduce non-engineers to systems engineering process based on his own ten requirements for systems engineering education. To avoid a misunderstanding of systems engineering for non-engineers, he even decided on the curriculum title "Getting Design Right" instead of systems engineering. This curriculum is built as a cyclic eight-step process for design. These steps are:

- Define the Problem
- Measure the Need and Set Targets
- Explore the Design Space
- Optimize Design Choices
- Develop the Architecture
- Validate the Design
- Execute the Design
- Iterate the Design Process

This approach is built on multiple sources: systems engineering, software engineering, Six Sigma, product design and development, and project management. This curriculum has been implemented in several forms (distant learning course for college freshmen, distant learning certificate program for working professionals, and text for use in other universities and corporate training facilities).

The *Technische Universität München, Germany* proposes since 2012 a laboratory on Systems Engineering in the context of product development [112]. This laboratory is held by three researchers in form of a five-day event, targeting Master and Bachelor students in Mechanical Engineering without previous knowledge in Systems Engineering. An industrial case study is used as a teaching framework, along with a subset of aspects and processes from a typical Systems Lifecycle in Systems Engineering. The selected processes are explicitly highlighted in the paper as "*planning of activities and responsibilities for various tasks; assessment, control, and decision-making concerning organizational processes, time management, tasks, design concepts, and chosen methods; stakeholder requirements definition, requirement analysis, and architectural design of the given system; implementation, verification, and validation of the system design*". Methods related to these processes are taught by examples, after which the students (as individuals or teams) make their choice in terms of which method may be best suited for each task and apply them to the use case of the laboratory. By this means, the students do not only acquire the Systems Engineering methods, but also several important soft skills, such as moderation, presentation, and discussion.

A similar approach has been adopted at Cranfield University, UK, as a part of a whole Masters degree program in Systems Engineering. It takes form of a one-week full-time laboratory based on a LEGO Robotics Kit [113].

Theoretical Courses

Already in 2000, Yurtseven from Dogus University in Turkey presented two courses that dealt with Systems Engineering and Design for students in Industrial Engineering [114]. These courses were mainly theoretical. The first one addressed senior level students and provided a background on the fundamentals of Systems Engineering. After introducing the main concepts of design and engineering, the course included several topics such as: Design Options, Engineering Systems Modeling, Analysis of System Reliability, System Dynamics and State Transition Matrix Models, Modeling the Research and Development Process, Systems Life-Cycle and Optimization, and the Management of Engineering Systems Design and Operations.

The second course addressed graduate level students and introduced some unconventional methodologies in Systems Engineering. It included the following topics: Introduction to the Management of Advanced manufacturing Technology, Introduction to Socio-technical Systems Theory, Cognitive Systems Engineering, and Soft Systems Methodology.

4.5.4 Programs Promoting Systems Engineering Processes

Considering our interest in system engineering standards and their processes, we have reserved for this last section of the chapter some systems engineering education practices demonstrating the usefulness of these processes.

Process Centered SE Course within Traditional Engineering Degree

A good example for the use of processes is the South Australian university bachelor of engineering program [115] that contains four courses in systems engineering representing 12.5% of the program. Each course is placed in each year of the program and represents 25% of a semester workload. The four SE related courses are: Engineering Communication and Innovation, Systems Engineering Management (SEM), Systems Engineering 1 (SE1), and Systems Engineering 2 (SE2). After raising the students awareness about the broad vision of systems engineering to achieve large things and about seeing the "big picture", SE1 introduces to the range of systems engineering standards, and teaches the major elements of requirements engineering. Afterwards, SE2 offers a year undergraduate course in two components: a lecture and a tutorial. The lecture component introduces students to different aspects of systems engineering processes, such as conceptual design, system approach, problem solving, design analysis, functional analysis, etc. The tutorial component asks students in groups of 15 to 20 students to develop a project up to a preliminary design review stage. The projects are provided by the teacher. Note that the SE1 and SE2 courses, which represent 50% of the whole program, both deal with SE standards and processes, their understanding and application. Even if the projects are not performed for a company, and no industry.organization partners are involved in the program, the authors of [115] stated that "*using contextual information and local realism is a means for enabling students to gain a industry relevance perspective in their development of engineering skills.*". This is done in the SE2 course by letting students work in large hierarchically structured teams, and by the focus on learning about SE processes that students acquire at the same time.

Note: This course emphasizes the interest of SE standardized processes for education purposes.

Process Centered Collaborative Engineering Environment

In order to facilitate the work of distributed design teams within the satellite industry, Hartmann and Knirsch [116] proposed an approach for Collaborative Engineering Environment (CEE). It is built upon an Internet portal which represents its first module, while the other four modules are: Process Guidance, Tools, Project Database, and Company Documents Management System. We are mainly interested in the structure and the use of the Process Guidance module which is responsible for providing the user with a customized view of the processes to be used along with its data and properties. This element contains the standard satellite design phases, each phase consists of approximately ten to twenty steps, and each step includes a definition of the task and its inputs and outputs. The main advantage of this solution is the ability to get a direct access to tools that have to be used for engineering a system.

However this solution was not developed for systems engineering, neither for satellite systems engineering education. With some modifications and additional features, this "*Process Centered Collaborative Engineering Environment*" might become a good educational environment for systems engineering. The changes may include:

- Allowing students to work collaboratively within the platform and to see the other results, by adding a "*project shared space component*".
- Allowing real time interaction and communication between students and their teacher, through the platform.
- Creating different users roles, associated with different environments of the platform, education entities, educators and students.
- Allowing for an easy use of domain-independent standards, with the ability to tailor the processes to meet the project size and goal, by adding a " *processes management component*". It is also important to allow the introduction of non standardized processes and domain specific processes.
- Adding the ability to adapt the platform environment by the students users, to meet the project structural architecture, and to allow them to apply the systems engineering processes to both the system, the sub-systems, and the system elements.
- Adding a competency management system.
- Adding students assessment features.

The proposed collaborative engineering environment (CEE) can be considered as derived from a more general paradigm, the Platform-Based Engineering (PBE). "*PBE is a cost effective, risk-mitigated system development approach that employs a common structure from which high-quality derivative products can be developed rapidly* [117]". The *PBE* Paradigm, and especially the *Adaptable PBE* paradigm, were widely studied by Madni [117]. The author highlighted their pros and cons, together with the impact of the adaptability on future resilient systems engineering. He stated that *"adaptable PBE is a key enabler of resilience and especially well suited to engineering long-lived resilient systems"*.

Development of systems engineering people at Thales

To improve performance and quality in systems engineering, Thales started CHORUS 2.0 [6], an Enterprise Reference System, accessible through an internet web portal. The main characteristics of this system are:

- COHORUS 2.0 identifies applicable homogeneous processes and roles, together with their redefinition strategies, in order to facilitate efficient and collaborative work of Thales employees.
- It is structured by processes, defined with respect to external standards and references, and includes processes, organization rules, methods and tools to be used.
- It provides the description of the key roles and responsibilities of key players.
- It ensures that fundamental processes are applied throughout the enterprise, but it allows these processes to be tailored, in order to meet specific constraints.
- It is a way to help employees to perform their day-to-day work.

In addition to the Enterprise Reference System, Thales own systems engineering methodology "Sys-EM" was developed, in order to perform CHORUS 2.0 activities in a methodological way.

To support the change generated by these transformation plans, close attention should be payed to Thales employees competencies development, in order ensure a good understanding and a good application on projects.

To do so, a systems engineering training path presented in Figure 4.1. has been created by Thales University, i.e. Thales own training center. The training path addresses all the systems engineering and architecting roles. Each module lasts less than 5 days. The authors stated that their work has to be challenged with external competency models such as INCOSE to guarantee that the learning path covers all the recognized skills. The specialty activities should be involved within the collaborative process to improve systems engineering.

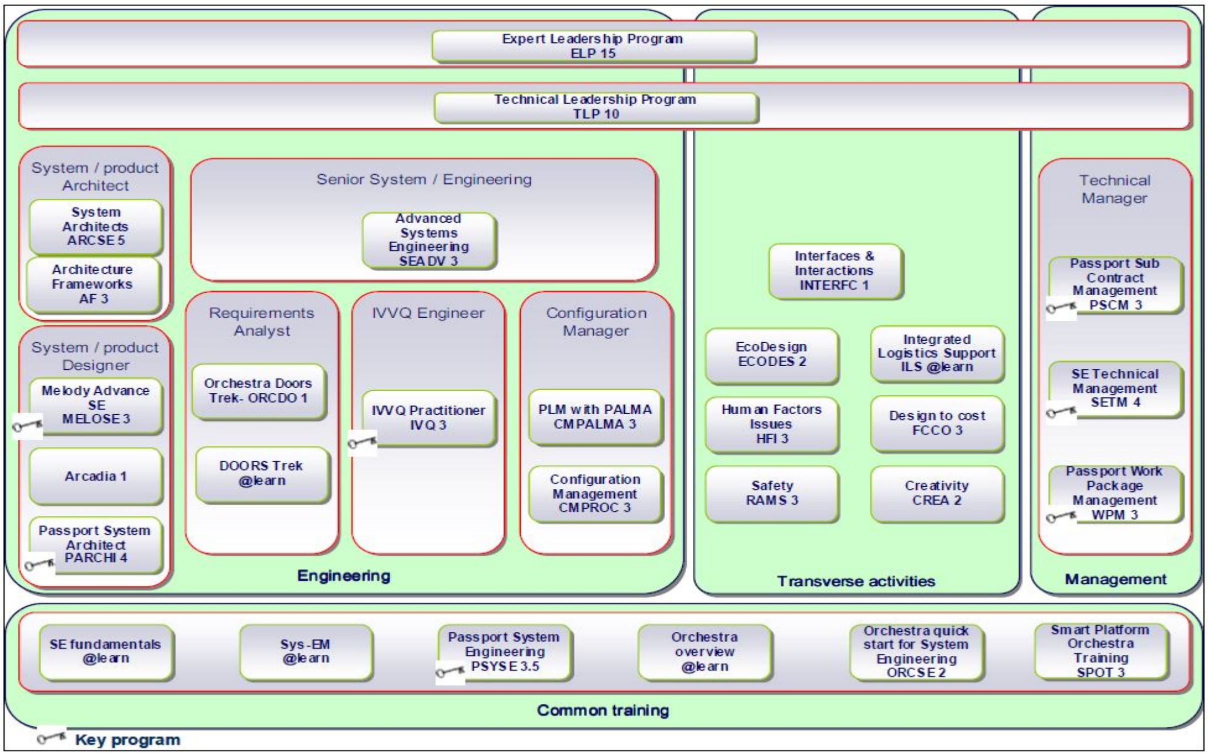

Figure 4.1: Thales University systems engineering training path [6]

CHORUS 2.0 itself is not the part which contributes most to development plans, but what can be done with it. It has been demonstrated that the integrated tooled up processes, the Sys-EM methodology and training had a significant impact on reducing the non quality costs, addressing more and more complex systems, and improving competitiveness by improving the product line approach.

A systems perspective for undergraduate students from any engineering discipline

Simoni et al. authors of [118] presented a new approach to provide a systems perspective to undergraduate students of any engineering discipline. The presented approach consist of creating a framework of system models by adapting and simplifying traditional systems engineering concepts, specifically using fundamental principles of Model Based Systems Engineering (MBSE). The resulting system models can the be presented to students, using a familiar use case (which can be a remote control for consumer electronics), in order to let them understand the system perspective by understanding the process of creating these models. Most importantly, students have to apply these concepts of system models of the proposed framework to an open-ended unfamiliar project of increasing complexity. The presented five system models are:

1. Stakeholder and feature model
2. Interactions and functional architecture
3. Creating the functional architecture
4. Writing technical requirements
5. Function decomposition and synthesis of a physical architecture

The authors affirmed that this system modeling process has improved both the quality of the resulted design projects and the students' understanding of their projects. They also think that the most useful features of this process are the limitations imposed on the concepts and models to be used that help students to organize their thoughts and efforts. A similar MBSE and SysML based approach was introduced by Singla et al. [119], as a self-learning systems engineering approach. And another one was introduced by Fernandez and Moreno [120], where MBSE principles were coupled with the organization systems engineering subprocess and applied to engineering a system in groups of five persons. They concentrated on a preliminary design rather than a detailed one, the main focus being on requirements engineering, functional architecture, and physical architecture. For that, the used systems engineering subprocess has five main steps related to these main components. As a conclusion, the use of systems engineering processes should help students organizing their engineering work.

Formalization of the specification process to logically guide trainers and trainees SE practices.

The interest of systems engineering standardized processes has been highlighted by Gouyon et al. stating that: "*Our eight years of experience in teaching SE within a master's degree in Complex Systems Engineering convinced us that, indeed, it is important for students to understand first how SE basic precepts are structured*" [121]. They also assumed that *"ensuring compliance with SE standards"* is one among other elements that can be followed for systems engineering teaching. This helps realize processes selection and application, and trainees can verify if they correctly applied the processes to solve the problem right. The authors insisted on the fact that students applying model based systems engineering (MBSE) need to be guided in order to correctly model the system and not just draw it. More theoretical details regarding the proposed four inter-operation subjects (Problem-Solution spaces, Source-Sink objects, Optative and Indicative moods, and Verification and Validation processes) can be found in [121].

Systems engineering and distance learning

A significant effort has been made by Badiru and Jones [122] within the Air Force Institute of Technology (AFIT) by developing an innovative approach for using systems engineering process for distance learning. Their main goal was not to train systems engineering by using standard process, but to use systems engineering processes, in their case the SIMILAR process, combined with additional Triple C and D.E.J.I models, in order to conceive an efficient distance learning environment. They targeted engineering graduate programs, without a focus on any specific traditional engineering discipline, neither systems engineering. Even if systems engineering and its standardized processes was a tool for them to conceive a better distance learning environment, they demonstrated that the resulting environment can be used for systems engineering distance learning by making the MS systems engineering degree program within AFIT fully available via distance learning.

Capstone K-12 and Cornerstone undergraduate project

In order to teach Systems Engineering fundamentals and to raise interest in STEM education in the United States, Patel et al. [123] proposed a complete and innovative pedagogical model that takes the form of a challenge, through an engineering-based product development Capstone project for US K-12 students, and also for Cornerstone undergraduate students. Unlike the previous solutions, it is technology-centered and incorporates some key principles of Systems Engineering in the provided teaching model.

The authors implemented their teaching model through an Integrated Design and Manufacturing Infrastructure (IDMI). It is essentially based on CATIA V6, a commercial tool from Dassault Systèmes, and employs both virtual resources such as CAD systems, and physical resources such as 3D printers. The pedagogical model includes five modules:

- *Introduction to Product Lifecycle Management: using provided video tutorials.*
- *Introduction to Systems Engineering Principles using moderately complex Electro-Mechanical Systems: LEGO Mindstorms.*
- *Computer Aided Design: using Dassault Systemes CATIA V6.*
- *Additive Manufacturing: using 3D printers and STL files.*
- *Collaborative Tools: using SwYm (See What You Mean), Dassault Systèmes online social network.*

The pedagogical model has been experimented in the Prize Challenge Summer Camp at Georgia, where students went through different stages of a product lifecycle: Co-create, Design, Build and Operate, to build a LEGO Mindstorms-based product for the challenge. Compared to other project-based courses and programs, the proposed pedagogical model appears to be the most complete approach, especially by making students collaborate together in a PBL Model, by introducing the notion of virtual design of the product, and by enabling distant collaboration using Catia V6 and SwYm.

However, even if it introduces students on Systems Engineering during the two first modules, it did not provide them with adequate infrastructure to ensure that they understand and pass through the appropriate systems engineering processes in designing their project. That means that it is more a solution for getting students interested in STEM disciplines than for educating them in Systems Engineering.

4.6 Chapter summary

We investigated the main goals of Systems Engineering Education and presented the current education practice of systems engineering within both academia and systems engineering practitioners. The presented list of current practices and approaches is not exhaustive. Many other approaches are used to teach Systems Engineering especially in the United States. Szajnfarber et al. [124] studied some of them and classified them into *Quizzes, Lab Reports, Design Projects, Arduino Projects, Exams, Homework, Labs, Lecture and class discussion, Predominately Exams and a Design Project, Design Challenges, Research Papers, Research Projects, Case Studies.*

Current situation

A number of lessons have been learned from the previous literature study. The most important one is: even if there exists a multitude of different teaching approaches, we should not ignore SE standardized processes within systems engineering educational environments, as they form a thread of the systems engineering principles. Some additional lessons learned are:

- The number of proposed SE programs is rapidly increasing, and enrollment in SE programs is increasing even faster.

- Defining or adopting and tailoring a competency model seems to be common practice within organizations before launching any career development plan.
- When an industrial launches a career development plan, it typically considers both standardized/generic systems engineering processes, and domain/enterprise specific processes.
- On-the-job experience learning is at least as important as classroom training, and when it is absent, it is at least replaced by a design project (sometimes by real project).
- Teamwork and systems engineering skills are an important part of systems engineering and should be considered as such in educational situations.
- Requirements engineering is a main topic in systems engineering, and education should consider its importance as such.
- A minimum of an undergraduate academic degree is appreciated before enrolling in systems engineering long-term programs. Sometimes it is a condition.
- Behavioral competencies consideration is as important as technical and managerial competencies.
- Implementing team mentoring is useful within industry and organizations, and this needs to be the case within team based learning approaches in academia.
- In the absence of systems engineering formal metrics to assess the learning outcomes, all means are good, even making participants pass the INCOSE certification exam.
- Digital learning environments are appreciated to enhance communication and collaboration between students and teachers or coaches, and to manage training documents and supports. We think that they should be useful to manage the learning process, too.
- Sometimes, even participants in a systems engineering education program are selected depending on their willing and their abilities to become good systems engineers. Not all engineers, even experienced ones, can become good systems engineers.
- Education programs in systems engineering should be continuously improved, just like in any other discipline.

SE Standard Processes for SE education

We cannot conclude this chapter without presenting the literature background of the interest of systems engineering standards and processes for educational environments. According to Rochet et .al [125], "*Systems Engineering processes are an aggregation of "good practices", elaborated through several years, which constitutes an accumulation of acquired experiences in project management*". Lamb and Rhodes [16] stated that systems engineering standardized processes may be a tool to enforce and develop systems thinking, and that they are a way to enhance effective coordination between individuals and teams working on complex problems. In the same perspective, Avnet [126] demonstrated that members learning outcomes within design teams are maximized when expected and reported interactions are aligned. This happens when the team follows a well-defined process in the technical design, based on information flow.

For Pfar et al., SE standards processes are ways for executing engineering tasks to support engineering excellence and overall effectiveness. They help reducing the variability, facilitate scheduling and cost estimation, and addressing problems across the system lifecycle in a systematic way [80]. Systems engineering is widely taught through On-The-Job experience, it can be the reason for misunderstanding of fundamental principles of systems engineering. This fundamental and missed knowledge can be acquired by a good understanding of systems engineering processes [127]. Harris [93] highlighted the need of an introductory course to systems engineering for final year undergraduate students, as one phase of an effective approach to develop systems engineers. He also assumed that this course should be based upon one of the available systems engineering standards, because according this will prepare for the foundation of top-down processes and multidisciplinary thinking.

We were able to trace the origin of systems engineering processes within systems engineering educational environments back to the 1992. In fact, Verma described in [83] the approach used by Virginia Tech for a SE teaching course, as a combination of three courses: systems engineering process, logistics engineering, and economic evaluation of projects. He also described the development and utilization of computer based courseware.

Systems engineering processes listed in standards, such as the ISO 15288, are described in such a way that a wide variety of organizations (e.g. enterprises, government departments) can use the same standard and develop usable processes, suitable for their domain, organizational culture, selected lifecycle model (e.g. waterfall, iterative) and project size and risks. SE processes described in standards are not usable 'as is' since they describe the requirements (i.e. the 'shall statements' or 'what to do') of the processes. Any organization, whether it is an industry a government agency or an academic institution, needs to document the SE processes to describe in detail 'how to do' (e.g. describing inputs and outputs, procedures, detailed content of documents).

Systems engineering activities are not described as a procedural set of steps to follow; rather, the systems engineer must imagine what needs to be done before doing it [100]. This is not contradictory to the use of standardized processes within the learning scenarios or within systems engineering practices in the organization. On the contrary, systems engineering processes will assist the practitioner to use his systems thinking and critical thinking skills to find unprecedented solutions for unprecedented problems, while being assisted and guided by the processes.

5

A Process Centered Adaptive Approach For Systems Engineering Education (ProCASEE)

5.1 Introduction

Our vision regarding what might be a good environment for teaching SE focuses on new disruptive technologies such as 3D and Virtual Reality (VR), Internet of Things (Iot), 3D Printing and Machine Learning, coupled with a Project Based Learning (PBL) model, and a process based learning path. Of course, this vision is not to be deployed today. However, in this chapter we present our current work results, that might lead to achieving this vision in the near future.

We aim to make the teaching of SE fundamentals true to reality. Therefore, we promote the learning-by-doing-paradigm, where multidisciplinary students from different locations collaborate in engineering a system requested by an educator. This is what we call a "system of interest" [4]. Students can adopt different roles such as designer, production operator, requirements engineer, architect or tester. They are guided to apply SE standard processes and therefore meet situations similar to real-life SE challenges.

Another accordance with real-life SE is the fact that our vision emphasizes two main components, a virtual and a physical environment, operating through an Internet-of-Thing (IoT) infrastructure. A high level of connectivity between these two environments is needed, not just at the engineering level, but also with respect to the teaching process. The educator should be able to track the learning activities inside both the physical and the virtual environments, in order to assist and evaluate students knowledge acquisition.

The global approach will therefore be a domain independent solution used by both educators and students within SE education organizations, and possibly with industrial collaborators, enabling a high level of collaboration and interaction. The main components of this approach are illustrated in Figure 5.1.

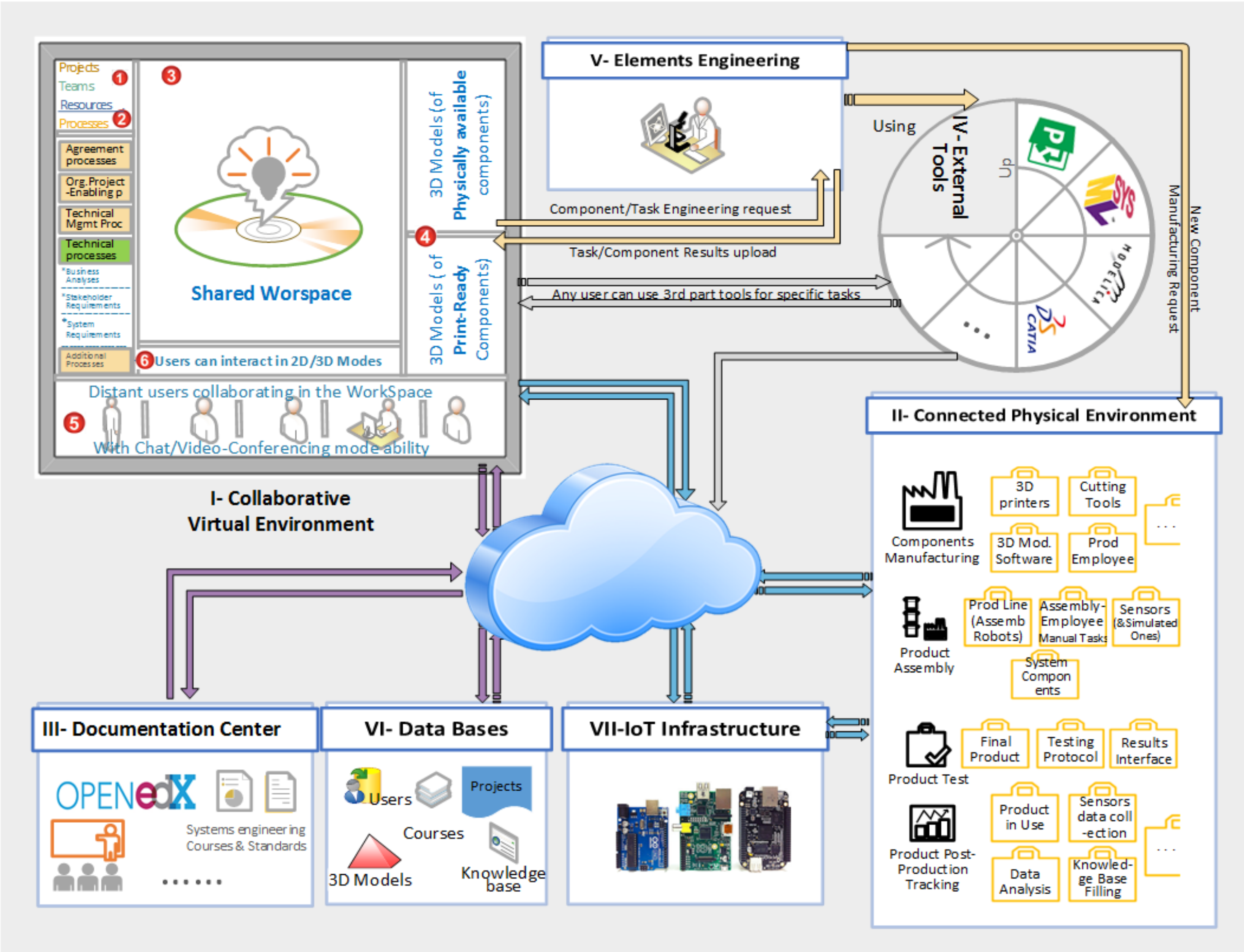

Figure 5.1: Our vision for a systems engineering educational environment

5.1.1 Primitive Resources

Primitive resources are the atomic components used to create a new system or to modify an existing one. For each tangible primitive resource, students have at their disposal a 3D model inside the *Collaborative Virtual Environment* (CVE), and a corresponding assembly part inside the physical environment. However, in order to satisfy the particular requirements or their mission, they may need additional components. In this case, students design a 3D Model of the missing piece, using the *Elements Engineering* component of this approach, and they produce the physical unit inside the *Physical product assembly environment*, e.g. using a 3D printer.

5.1.2 Collaborative Virtual Environment CVE

This is the main component of our vision, and our key topic for this thesis. It represents the engineering workspace that manages all the efforts of systems engineering in order to produce the right system for the right stakeholder. It is intended to be a web-based application where students can collaborate and interact with educators and coaches throughout the whole project, including the entire system lifecycle. Figure 5.2 shows that this environment incorporates several elements:

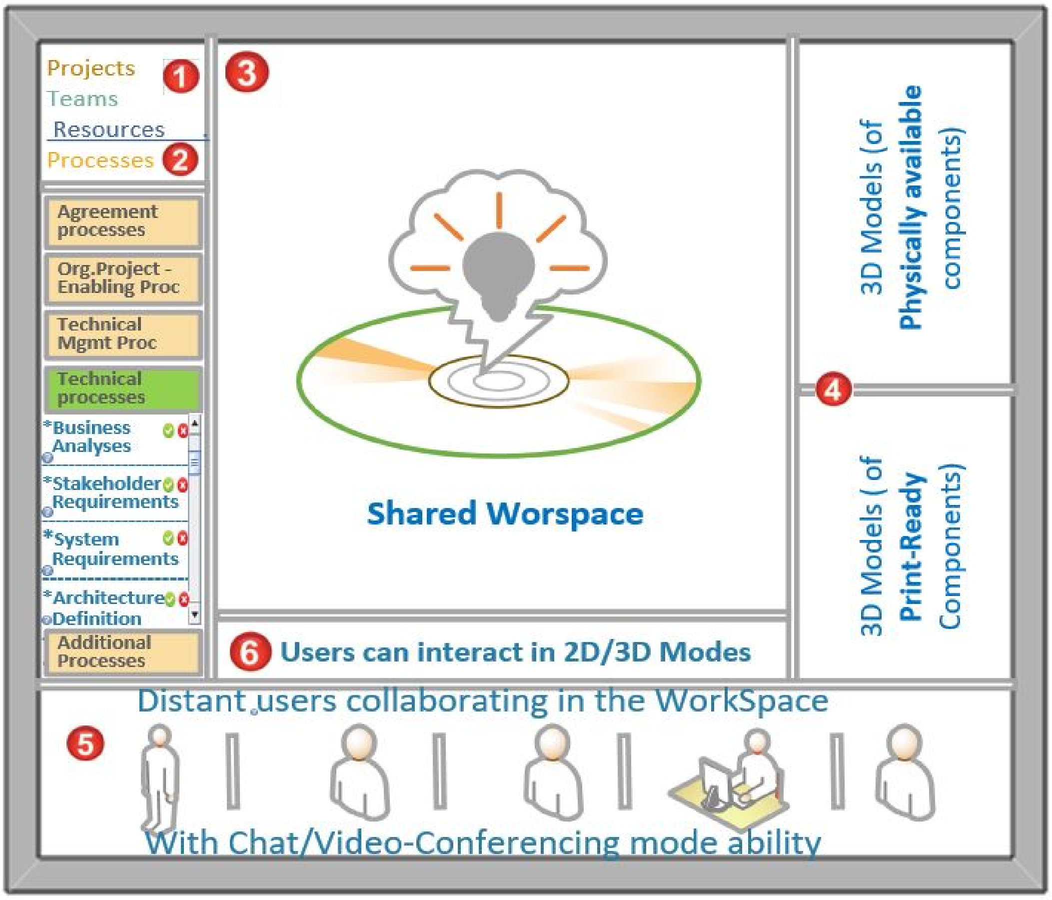

Figure 5.2: The main components of the Collaborative Virtual Environment

- Projects, teams, and resources **(1)**: The top left part of the screen shows a three-component menu. Its first element, *Projects*, includes a description of the project mission, given out by the educator. By clicking on the *Teams* element, students find information about the other members of their team co-working on a specific project. They can also manage the different roles assigned to each student during the product. Under the third element of this menu, *Resources*, students find a collection of suggested resources provided by the educator, to guide them through the engineering processes.

  The educator plays two different roles. First, he can act as the acquirer of the system-of-interest, but another entity can replace this role and define the project mission. Second, the educator plays the teacher role, assisting students throughout the entire process of engineering the system, and evaluating them from an individual and a collective points of view.

  Prior to that, students must be registered in the students data base, with their biography, curriculum, competencies and skills. This information helps the educator to efficiently perform student-team-project assignments. A team of students can be responsible for engineering of an entire system, or their system can be an element of a bigger one. In this case, the educator assigns teams to specific parts of one higher-level project.

- Lifecycle model processes **(2)**: This is one of the most important parts of the collaborative virtual environment, where students consider the lifecycle model processes, in order to engineer the requested system. The lifecycle model is defined by the acquirer/educator based on the learning goals and the nature of the system to be engineered, and on students competencies and skills. In addition, for more information about the currently used processes, or for further training regarding the used resources and methods, students can always access the *Documentation Center* which may be a LMS or a MOOC platform.

- The shared workspace **(3)**: Represents a virtual place where students can report the results of their performed tasks. All team members, including the educator, have a complete overview of their progress at any time, and they are able to annotate and exchange work results and ideas.
- 3D virtual models **(4)**: The shared workspace gives access to the 3D models that can be used as primitive elements in the design process. These models may already exist in the physical world, or they can be 3D-printed to assemble the final system-of-interest. As illustrated by **(6)**, students are able to interact with the shared workspace, both in 2D and 3D modes, depending on the nature of their task.
- Collaboration **(5)**: Using the collaborative virtual environment, students and educators can communicate and exchange through a chat or video-conferencing system.
- An additional component, not shown in the figure, is the competencies management system, that can be used by the organization to create his own, or tailor an existing systems engineering competency model. The adopted competency model can then be used by educators to link competencies to activities of the adopted standardized processes. By doing so, they will be able to better manage the learning scenario and control the competencies to be acquired.
- Another additional component is the ability for the teacher to review and annotate the different tasks results during the system engineering lifecycle, and to evaluate the results at the end of the project.

5.1.3 Physical Environment

The Physical Environment represents the traditional manufacturing factories and production lines. We distinguish the *Manufacturing Environment* with activities relative to new components production, using tools and machines such as 3D printers, and the *The Assembly Environment*, which may include a robot based production line for components assembly, with the help of an *assembly operator*. Moreover, we assume that all components contain sensors that allow a real-time tracking of the assembly operations, reporting relevant data to the CVE through an adequate IoT architecture. The physical environment includes a *testing environment* where the operator can perform a series of test procedures on the assembled system-of-interest, and report the results to the CVE. This approach is particularly interesting if the IoT infrastructure allows a post-production tracking of the system-of-interest, allowing for additional tests to be directly performed from within the virtual environment. The physical environment within this learning approach can be an workshop within a university or an engineering school, as it may be a a co-working space (what is called a Fab lab [1])

[1] A Fab lab (fabrication laboratory) is a small-scale workshop offering (personal) digital fabrication. A Fab lab is typically equipped with an array of flexible computer-controlled tools that cover several different length scales and various materials, with the aim to make "almost anything". This includes technology-enabled products generally perceived as limited to mass production. [Wikipedia]

5.1.4 Learning and Documentation Center

The documentation center is a virtual space where students have easy access to educational resources, such as documents and videos, on-line libraries, LMS and MOOC platforms, etc. By this means, students can find the appropriate time to learn more about different aspects of the SE discipline and domain-specific engineering, including its standards, processes, methodologies, and consult useful information and tutorials about SE related tools. It is also possible that students get direct access to scientific publishers resources and learning platforms. Students may also need to access to specific templates document relatives to specific engineering tasks. As they requested during our experimentation plan, a sample project, which may be an example of a past project, would also be highly helpful, especially on improving their understanding of the "how-to-do".

5.1.5 System elements engineering

This part of the global vision is related to a scenario where students are asked to design a required component for the system-of-interest, which turns out to be so complex in itself that it needs collaboration between different students in the team or between different teams. In such case, the component can be considered as a new system-of-interest in itself, and students have to engineer it by applying the same SE lifecycle processes as for the higher-level system-of-interest. After performing each task, the outcomes are uploaded to the collaborative virtual environment.

5.1.6 Third-party tools and resources

A good SE learning solution should be highly tool-independent, so that every organization, educator and student can choose the most adequate tools for their SE activities. Ideally, students should be able to access their tools directly inside the collaborative virtual environment. Otherwise, they may resort to external tools and upload the results to the collaborative virtual environment.

5.1.7 IoT Infrastructure

The role of this module in our global approach is to enable a real-time connection between the two main components of the solution, the virtual environment (mainly the CVE) and the physical environments including all its sub-environments. This will allow students to:

- Verify the state of physical resources, like the availability, especially when the solution is deployed in an environment with a large number of teams with limited resources. This represents a real life problem because factories need to carefully manage their resources and their availability.
- Directly access the physical resources from a dedicated interface inside the CVE, for example: the manufacturing operator launching the 3D printing of a specific component directly from the CVE.
- Automatically manage their manufacturing and production resources, by booking time-intervals through the CVE interface, accordingly to their production and manufacturing plan. The students will not be able to use the 3D printer or another tool, if they do not have an allocated session. This will improve their resources management skills.

- Make and test their production plan virtually before sending it to the physical environment, so they will have the opportunity to learn about production optimization processes.
- Maintain a real time distant view of the production, testing and post production plans execution.
- Track the engineered connected-products, in post production use, in order to improve them in a support stage.

5.2 The focus of this thesis

This vision illustrates the ideal environment for systems engineering education. However, this thesis is focusing on one single element of this vision. We have been working on conceiving, developing and testing the most important component of this vision, *"The Collaborative Virtual Environment"*. As we will see in the next chapters, our contribution is mainly the proposition of a novel approach for SE education: a processes centered approach that promotes the use of SE international standards and allows students to learn through a distant project-based learning approach.

5.3 Contribution: A Process Centered Adaptive Approach For Systems Engineering Education (ProCASEE)

At this stage, a new approach for SE education is proposed, an approach based on the use of standardized SE processes, where each learning action is seen as a process, standardized or not, tailored or not. The approach allows adapting processes by educators, to fit the project size, learning goals, students competencies, etc. It will be supported by a web-based environment, that allows the use of systems engineering processes to define the most adequate learning scenario for each project. The environment connects geo-distributed students working in the same team, to execute systems engineering processes as defined/tailored by the educator to engineer the requested system, while being assisted by the educator and/or an industrial expert. In short terms, we are proposing an approach that possesses all the features of the collaborative virtual environment identified before.

To begin, in the next sections and in our experimentation and use case, we are considering only the technical processes to demonstrate how the proposed approach can be used. Note however that it is possible to use other kind of processes, for example to standardize the system or subsystem agreement procedures between the acquire (educator) and the supplier (students). As a matter of fact, the solution may also implement the acquisition and supplier processes, the first processes of the ISO/IEC/IEEE 15288 agreement processes, where educators and students can negotiate and agree on the work to be done. Technical management processes, and project enabling processes may also be used. It's up to the educator to select the processes to use, as we will see later. The proposed approach and its supporting framework, allow even the educator to organize his processes in different stages to formalize the lifecycle model, which can then be augmented with necessary resources and methods.

5.4 Learning scenarios

In this section we will present the main scenario of use of the most important features of this approach. We start by the global learning scenario, that include two sub scenarios, the new project creation scenario and the system engineering scenario. The former includes in his turn the processes management scenario, while the later encompass the systems engineering processes to be executed by students during their system-of-interest engineering. Some missing scenarios concern the organization roles, such as for managing students and educators accounts.

5.4.1 Global learning scenario

This represents the high-level learning scenario. As illustrated in Figure 5.3, it encompasses other sub-scenarios. The proposed solution has two main players. On the one hand, educators are responsible for creating projects, by defining their goals and life-cycle models. They also assign student teams and resources, while ensuring assistance and coaching, and results and processes execution quality assessment. Students, on the other hand, are responsible for collaboratively engineering the system, with respect to the processes defined by the educator.

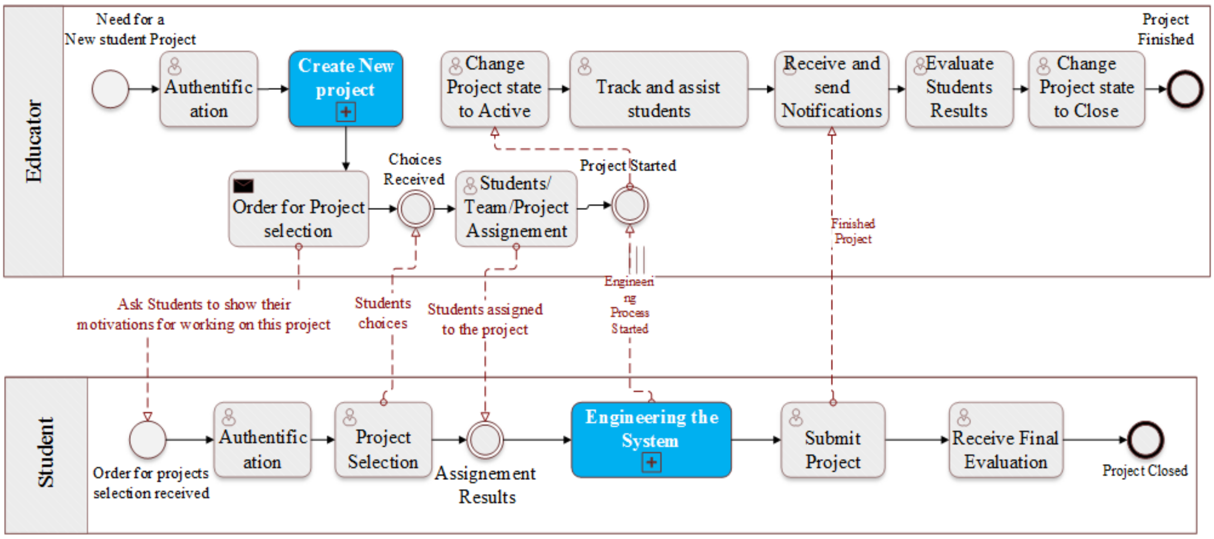

Figure 5.3: Global Learning Scenario

5.4.2 Educator: New project Creation Scenario

As illustrated in Figure 5.4, creating a new project goes through several stages. The educator defines the project title and description, as well as the life-cycle model, based on standardized systems engineering processes which will be followed by students. For this purpose, the educator selects and tailors, if necessary, a number of processes from the processes database, while documenting them. If a specific process does not exist in the database, it can be added using the processes management system, as illustrated by Figure 5.5. Finally, the educator specifies the resources and tools to be used by students.

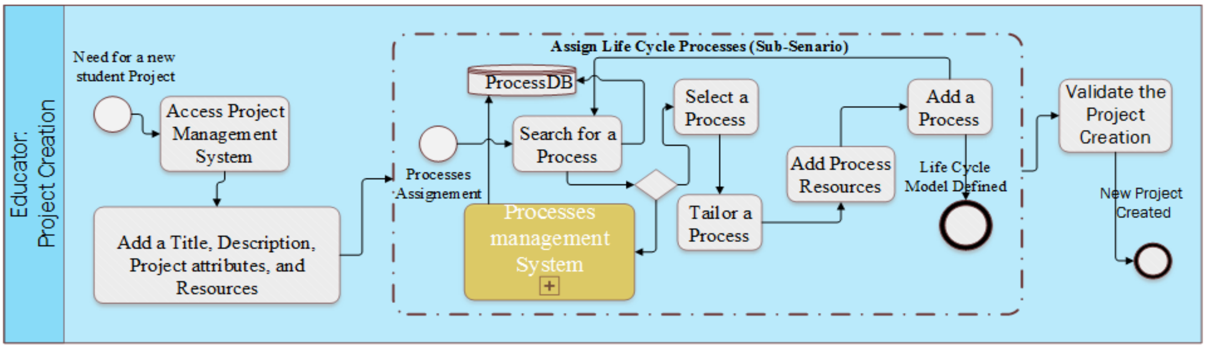

Figure 5.4: Project Creation Scenario

For the time being, the life-cycle model is defined by the educator. In the future, if the solution will be used to advance students knowledge in systems engineering, it may be possible that students with a background in SE are asked to define the life-cycle model for themselves. This will help them put in practice the tailoring process, and better understand the wholeness idea of systems engineering and systems thinking.

5.4.3 Educator: Processes management scenario

This scenario allows educators to create, adapt or remove SE processes. Educators can also add domain-specific processes or create a set of additional activities to be performed by students as a process. The adopted architecture for a process is compliant to the 15288 standard. In addition to its purpose and its outcomes, a process is defined as a set of activities, and each activity is defined as a set of tasks to be performed. Two components were added to this structure, the process inputs and the process outputs. In the next chapter, additional informations will be added to processes: the competencies related to their activities.

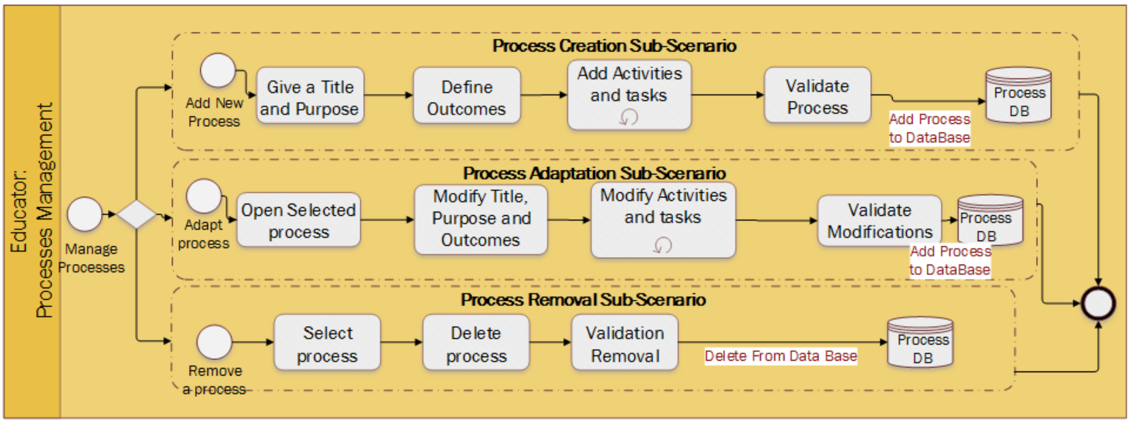

Figure 5.5: Processes Management Scenario

5.4.4 Student: Project engineering scenario

This scenario, described in Figure 5.6, represents the high-level stages that students will follow to engineer the requested system. After selecting an active project, and after managing their roles, students engineer the system by performing the tasks of each activity of the life-cycle model processes. The tasks are done using the adequate external tools and methods, and their results are uploaded to the team-project workspace.

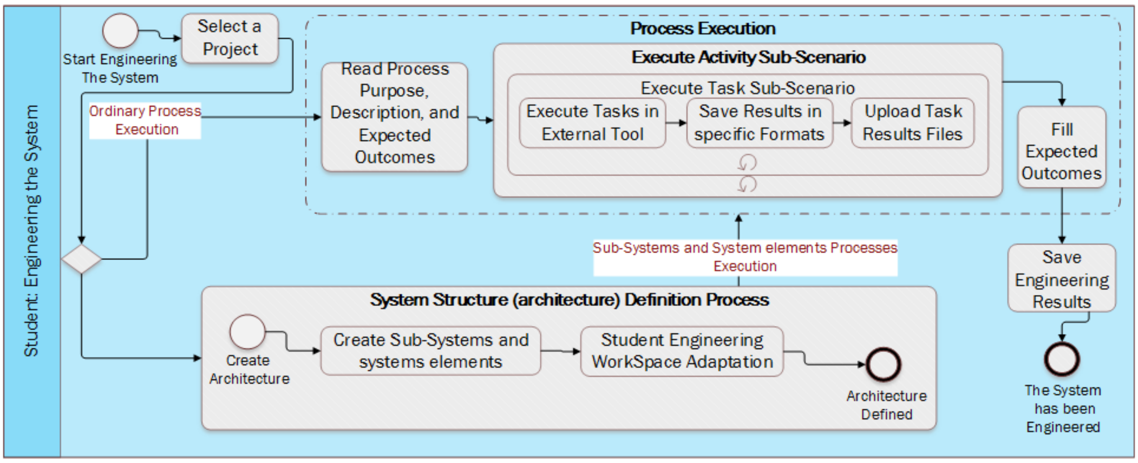

Figure 5.6: Project Engineering scenario

5.4.5 Student: Environment adaptation according to system structure

Students will need to define the structural representation of their system-of-interest as a set of its subsystems and system elements. The students workspace will then be automatically adapted to provide sub-workspaces for each subsystem and for each system element, so that they can execute the lifecycle model not only for the whole system but also for its subsystems and system elements. This feature is illustrated at Figure A.11, in Appendix A.

5.4.6 Virtual 3D representation and the design of a tangible system

Virtual 3D representation and the design of a tangible system are becoming indispensable in any system design process. In the first version of our solution, we proposed virtual system assembly in a collaborative mode, where students in different locations are able to visualize and interact with the system being assembled in real time. Currently, the virtual design is possible for a limited number of pieces. If users want to use the 3D features, they can only engineer systems based on the LEGO® MINDSTORMS® Education EV3 Bricks and their additional elements (Sensors, Motors, etc). However, if 3D virtual design is irrelevant for students and educators, the solution can be used to engineer any kind of systems. Due to the complexity of design assembly for any kind of systems, we decided to drop this feature for the time being, and we recommended its adaptation for future versions of the solution, to be used for collaborative 3D design review of the system instead of system assembly. The implementation of this feature within the first version of our solution "ProCASEE", is illustrated in Figure A.16., in Appendix A.

5.5 Collaboration

Collaboration is enabled and encouraged in this solution, allowing students to work together on different tasks of the same project, while maintaining a global vision of the work all along the engineering scenarios. More features are planned to enhance collaboration, such as annotation of results, or chat and video-conferencing.

5.6 Assessment

Students assessment is out of the scope of this paper. However, we profoundly believe that this solution will help implementing new assessment methods, allowing educators to objectively evaluate students with respect to various teaching objectives. We already identified a number of aspects that need to be considered for assessing students in a SE learning experience, such as: *resulted design assessment, processes execution quality assessment, acquired knowledge assessment, and acquired skills assessment.* Resulted design assessment includes the different processes activities and tasks results assessment, and processes outcomes-documents assessment.This list might be extended by other metrics, depending on the domain, and the organization and educators goals.

The results interpretation, by the analysis of the extracted information are not in the field of interest of this thesis. In other terms, we will not provide specific methods about how to use the extracted information.

5.7 Chapter Summary

The presented approach for SE education is based on the use of international standardized processes within a project-based-learning approach. Thanks to this concept, students will learn to not only engineer the requested system, but also to engineer it the right way, using real-life SE practices conveyed by standardized, tailored and documented processes, together with communication, team management, collaboration and related soft skills. The main advantages of this approach are the processes, life-cycle, and projects adaptation and management components, as well as the shared workspace for students engineering tasks during all the lifecycle. Another advantage of the solution resides in its ability to help in meeting the challenges of a project-based-learning approach, in particular by opening a way of assessing students by different metrics, including: the final results, the execution quality, the acquired knowledge, and the acquired skills, as well as students management within distributed PBL experience. In the next chapter we will present a new "competency management" component. We added this component to enhance the approach and improve its outcomes for both organizations, educators and students.

6

A competency-driven approach and competency management system

6.1 Introduction

The importance of systems engineering competencies management has been proven in the previous chapters. Systems engineering competency models are a unique way that enable systems engineering competencies management both for systems engineering practitioners and academic use. In industry and government institutions, their relevance is represented by the number of competency models being created, tailored, and improved over the last years. In the academic field, research has undergone significant evolution in this subject, not just for systems engineering competency managements but also in other fields of traditional engineering. Competencies management and competency models within academia (research, universities, colleges teaching systems engineering) are motivated by several relevant needs such as:

- The produced systems engineers over different promotions and from different universities should all have acquired a homogeneous knowledge in the field of systems engineering.
- Systems engineering is most acquired through On-The-Job experiences, or at least through project based learning design project at the academia, and this is done over years and over different projects both in organization and academia. We need to ensure a continuity of skills and competencies acquisition.
- The acquired competencies by a specific individual can and should be used to adapt his own learning path over years.
- Technology and related fields are experiencing a rapid growth, with emergence of new learning subject. Competency management systems using competency models, when adopted the right way, will reduce the necessary time to create new training opportunities and new standardized curriculum.
- Specifically for our work, educators need assistance in creating new learning scenarios starting from the competencies to be taught, and a competency model adoption should play a major role.

We answer these challenges in this chapter, by adding a new component called "competencies management system" to our approach.

6.2 Most suited SE competency model for SE education

Almost all systems engineering competency models affirm their ability to be used within colleges and universities providing training and teaching on systems engineering. However, most of these models do not offer techniques to assess the acquired knowledge in a systems engineering educational environment, regarding their defined competencies.

Our goal is not to provide a specific competency model, developed for this purpose, to be directly applied within an educational environment. Even if we wanted to, it is not possible to do that, as the educational environment differs from one domain to another and from one need to another. Neither is our goal to identify which competency model from the previously cited ones should be used in every educational environment. We want the proposed approach and its supporting platform to be models-independent as it is already standard and tools independent. By doing so, each organization will be able to inject its favorite competency models. For example, if an educational environment targets the space industry to produce systems engineers, it will be most suitable to adopt and tailor the NASA competency model, etc. However, we suggest that, beside of its advantages, the INCOSE competency framework is not suited as it is for now. In fact, this model should be improved, especially by a new structure that allows the devision of its competencies into competency elements. INCOSE Framework competencies descriptions should also be improved to be more understandable, before we can use it within the proposed approach and platform. To demonstrate the abilities of our proposition in the next sections, we will make use of the NASA competency model. In fact, it present a very well detailed competencies divided into competency elements, and its structure provides attributes to define acquisition and assessment methods for each competency element. However, this choice don't means that the NASA competency model is the best one for any learning experience, and independently of the domain where the learning operates.

Our first goal in this chapter is to provide a way to enable organizations (colleges and universities) to select any competency model and to tailor it to their own need, within our solution. Second, we aim to provide educators using our solution with tools that allow them to use their organization competency models and to link them with their preferred systems engineering processes. Third, when educators create a new project to be engineered by students in the proposed distant PBL context, we want them to be assisted in the selection scenario of systems engineering standard processes. So, depending on the competencies and skills students have to acquire, the system lifecycle model will be automatically generated and can be further tailored.

6.3 SE Competencies and standardized processes

Just as systems engineering standards do not have a unique format, systems engineering competency models do not have the same architecture. We want our competency management system to accept all kinds of competency models, and to be linked with different systems engineering standardized processes. At the beginning, this system should be at least compatible with the ISO/IEC/IEEE 15288 and the ISO/IEC 29110 standardized processes and activities. Figure 6.1. illustrates the structure we selected for systems engineering standardized processes to be used within the solution.

When adopting new systems engineering competency models, organizations should use the structure we defined for this purpose, as presented in Figure 6.2. The adopted structure is identical to the NASA competency model structure. However, even if we adopt the NASA competency model structure which contains Four proficiency levels, our approach is only targeting for now the two first levels. But we decided to keep the four levels in our platform design, as it don't impact the performances or learning scenario, but might be useful in the future.

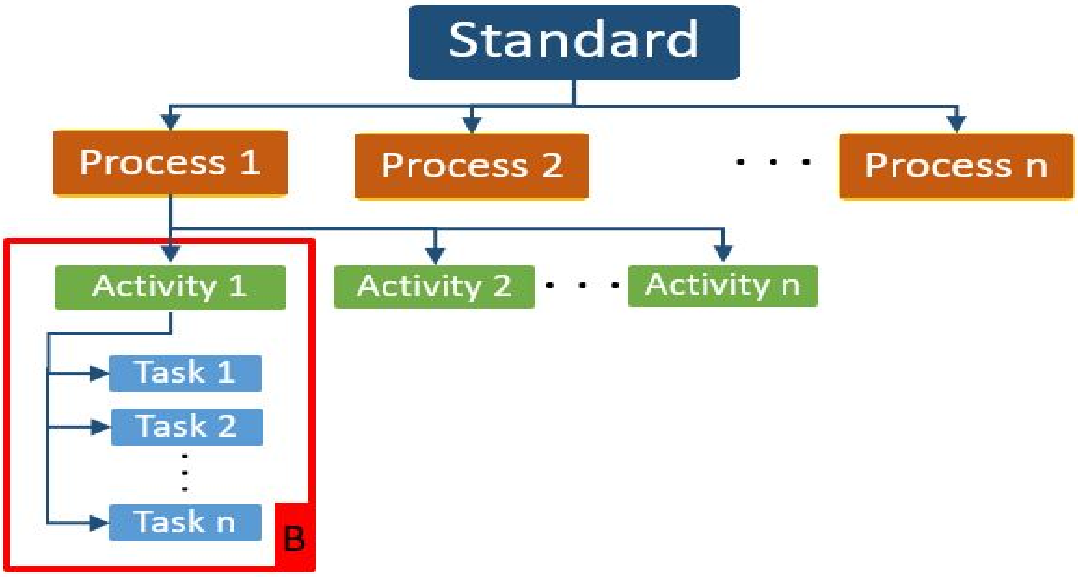

Figure 6.1: Adopted structure for standardized processes

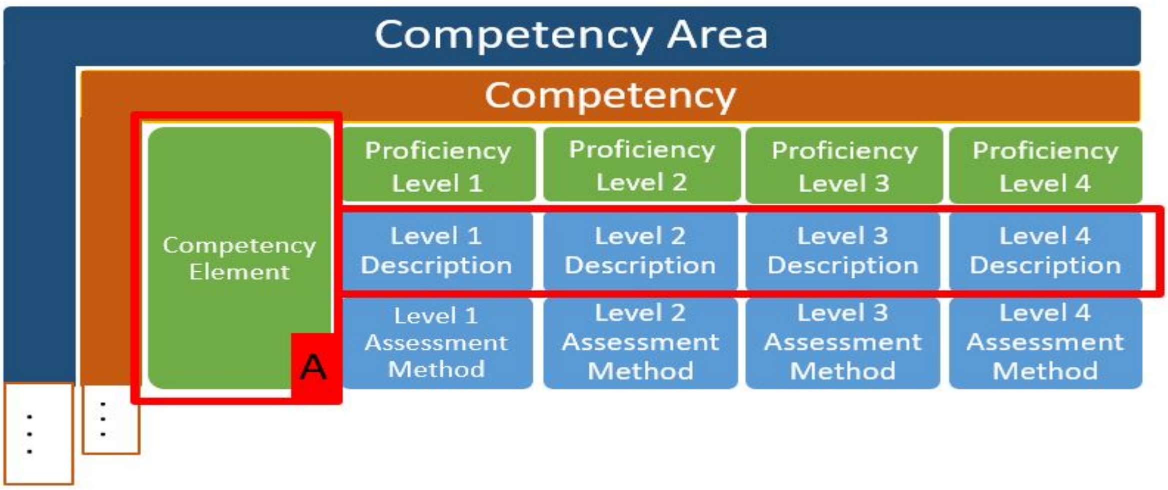

Figure 6.2: Adopted competency structure

We suggest that systems engineering competencies from the adopted competency model should be linked to the used systems engineering standardized processes, tailored or not. This association can be done by linking part B from Figure 6.1. (i.e. the process activity) to part A from Figure 6.2. (i.e. competency elements associated with a specific level). Each competency (competency element) can be linked to more than one process activity, and each process activity can be linked to more than one competency element. Therefore, close attention must be paid regarding what we call a process and what we call an activity between different standard terminologies. For example, if an organization considers using the 29110 standard, it should use its processes activities as independent processes, and their tasks as independent activities, in order to comply to this structure.

Beside the fact that only the NASA model structure divides competency into competency elements, we choose to link the activities with the competency elements instead of the competencies. We consider this to be the appropriate structure for competencies acquisition management within an educational environment. Other competency models limit themselves to the competency definition, while including several points (tasks) within each competency. If we want to use a model that does not specify the competency elements, we should change its structure slightly by classifying the set of activity tasks into several competency elements. Our motivation for this structure can be exemplified by this: the architecture allows us to link a competency element to an activity, instead of linking the competency to a process. Otherwise, it is difficult to ensure that a student acquired a competency directly by executing the process, as the process may be tailored for a specific situation to only include one activity instead of several activities which define a competency. When a student correctly performs a tailored process activity, he will only be able to learn the set of sub-competencies (competency elements), related to the activities forming this process.

6.4 New Learning scenarios

The inclusion of competencies in our approach led to the adaptation of the learning scenarios. The main new roles that have already been integrated are:

6.4.1 Organization new role: Competency model definition

In the previous approach an organization (entity using our solution for systems engineering education) had two main roles: educators and students management, and organization structure creation. With this adaptation, the organization have to manage systems engineering competencies. Their unique new role is to use the newly proposed competency management system, in order to create a new or adapted systems engineering competency model, following the structure in figure 6.2. This competency model will then be used by organization educators like in the scenarios of Section 6.4.2 and 6.4.3. The organization will be able to automatically gather information about the acquired competencies of their students, for grading and administrative purposes.

6.4.2 Educator First new role: Matching processes with competencies

We offer the educator the ability to specify a set of competencies that can be acquired by the right execution of each systems engineering processes activity. A competency here means a "competency element". When the educator creates his processes and adds them to the database, he also needs to add a set of competencies susceptible of being learned by each process activity, as shown in the red components of Figure 6.3.

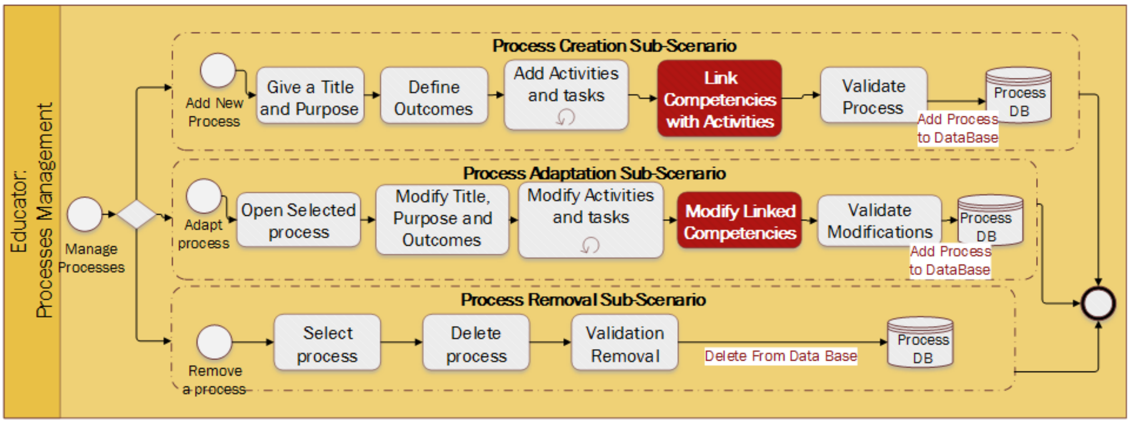

Figure 6.3: Matching processes with competencies

6.4.3 Educator Second new role: Competency based definition of the system lifecycle

Previously, educators had only one way to define the system lifecycle model associated with each new project, so that students engineering the associated system will be able to follow it. This operation was done by the educator selecting which processes to use, and by tailoring them manually if necessary.

With the new adaptations, educators have a new way to define a system lifecycle. They can just define the competencies to be taught, selecting them from the competency model, and the processes forming the lifecycle model will be automatically generated, thanks to the existing associations between the different processes activities and the competency model competencies. They can then tailor them if necessary, for example by adding activities that make students focus on a specific task, develop soft skills or domain-specific technical knowledge. Figure 6.4 illustrates this new ability. This new role should be improved, to allow the educator to choose which are the different stages of the life cycle model, the automatically selected processes will be included.

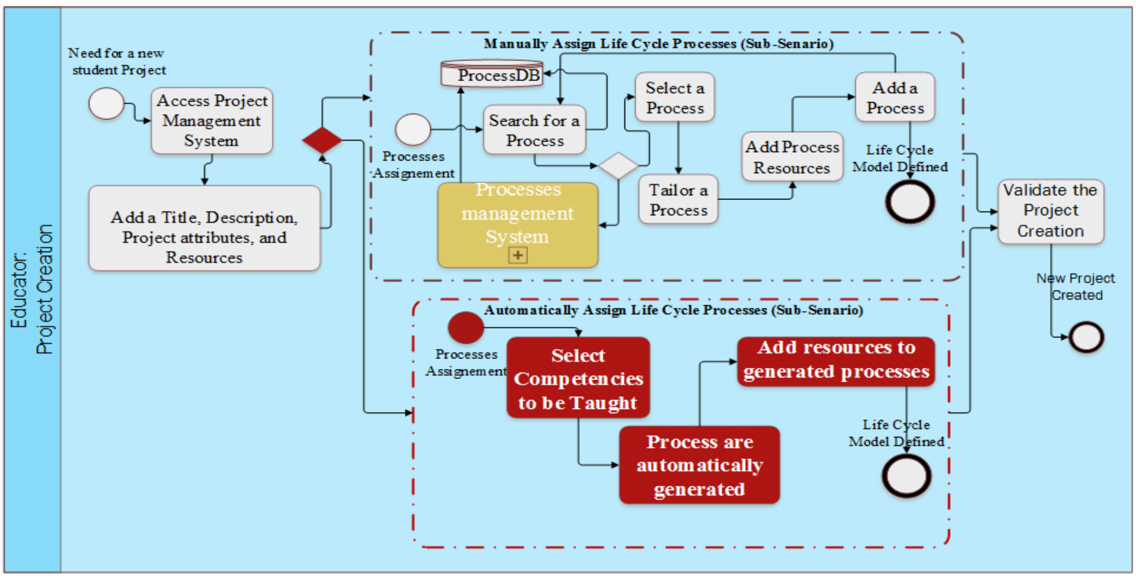

Figure 6.4: Competency based definition of the system lifecycle

6.4.4 Competency based students-projects assignments

In the future, we hope to provide educators with more features, so that they are able to take into consideration their students actual competencies in order to adapt the learning scenario to the competencies to be learned.

6.4.5 Student competencies acquisition and verification process

We also hope that new students assessment methods will be designed to support this solution, with regard to the competencies and skills of the competency models.

6.5 Use Case: A way to link the NASA Competency Model to the 29110 processes and activities

To demonstrate how a competency model can be used jointly with systems engineering processes, we present in this section how we successfully integrated tailored competency model, with a systems engineering standardized processes. We started by defining and integrating the tailored competency model that contain a set of NASA competency areas, along with their competencies and competencies elements within our solution.

As illustrated bellow, the selected competencies are the 17 competencies (Four competency areas) related to systems engineering technical processes and professional and leadership skills. We also added one competency, which is the "Knowledge Capture and Transfer" as you can see in the next list, which illustrates the NASA model competencies we kept. As stated before, for simplicity reasons, we are only considering for our use case the technical processes, so for the competencies too we are mainly interested by the list bellow without the technical management competencies, which should be used in a real life engineering, the same as the technical management processes.

- Competency Area: 1.0 Concepts and Architecture
  - 1.1 Mission Needs Statement
  - 1.2 System Environments
  - 1.3 Trade Studies
  - 1.4 System Architecture
- Competency Area: 2.0 System Design
  - 2.1 Stakeholder Expectation Definition and Management
  - 2.2 Technical Requirements Definition
  - 2.3 Logical Decomposition
  - 2.4 Design Solution Definition
- Competency Area: 3.0 Production, Product Transition, Operations
  - 3.1 Product Implementation
  - 3.2 Product Integration
  - 3.3 Product Verification
  - 3.4 Product Validation
  - 3.5 Product Transition
  - 3.6 Operations
- Competency Area: 9.0 Professional and Leadership Development
  - 9.1 Mentoring and Coaching
  - 9.2 Communication
  - 9.3 Leadership
- Competency Area: 10.0 Knowledge Management
  - 10.1 Knowledge Capture and Transfer

Afterwards, we deployed the entire activities of the SR (System Definition and Realization process) and PM (Project Management process) processes of the ISO/IEC 29110. These activities have been deployed using the processes structure of our solution, i.e. the 29110 activity is deployed as a process, its tasks are deployed as activities, and when tasks are defined as a set of point to perform, these points have been represented as tasks. To respect the solution structure, when a standard activity seems to not have more lower levels (points to be performed), at least one task is attached to this activity taking the name of the activity. As you note we choose the 29110 to demonstrate the concept at this stage, while we used the 15288 standardized processes for our experimentation plans. The first reason of doing that is to demonstrate that the proposed solution can be used with different standards, and second, to demonstrate that it can be used with different configurations about the stages and the processes to use. We used the entry profile of the 29110 series, illustrated by the technical report TR.29110.5.6.1 2015 [53]. It's also possible to use the Basic Profile, which might be more appropriate for academic institutions education students on systems engineering.

Finally, we added to each defined process activity one or more competency elements from the adopted (tailored) competency model, and which are susceptible to be acquired by the user performing the activity. Even if all the 29110 processes and activities are made available within the solution, we only focused on matching the SR process activities (considered as technical processes), totalizing the number of 6 processes and 34 activities. We decided to associate the same level of proficiency (the level 2) for all the associated competency elements. The matching operations were performed based on both the activity and the competency element titles and descriptions, the activity inputs and outputs (defined as tasks inputs and outputs in the standard document), and the process outcomes and goals (defined as activities outcomes). We ended up with the following Figure 6.5. to Figure 6.10., illustrating the different relationships between the competency elements and the processes activities. Note that we also did some additional matching between some activities generated from the use of our platform as illustrated by Figure 6.11. Note also that the results showed within the next tables has been obtained by an intuitive linking, and an experimented educator or professional in systems engineering may find some lacks, and can improve the obtained results. Another time, this is just a demonstration of the principles of our proposition.

SR.1 system definition and realization initiation		
	Activities	**Competency Elements**
1	SR.1.1 Revise the current Project Plan with the Work Team members	1.1.1 Mission Need
		1.1.2 Current situation
2	SR.1.2 Define the data model of the project	1.1.3 Mission Needs Statement Formulation
		1.3.1 Concept Definition
		1.3.2 System Model
3	SR.1.3 Set or update the implementation environment	1.2.1 System Environment Identification
		1.2.2 Design Guidance

Figure 6.5: The **29110 system definition and realization initiation** activities linked to tailored NASA competencies elements

SR.2 system requirements engineering		
	Activities	**Competency Elements**
4	SR.2.1 Elicit acquirer and other stakeholders requirements and analyze system context	2.1.1 Stakeholder Identification
		2.1.2 Stakeholder Expectation Definition
		2.2.1 Requirements Scope
5	SR.2.2 Verify the Stakeholders Requirements Specifications	2.1.3 Stakeholder expectation Validation
6	SR.2.3 Validate the Stakeholders Requirements Specification	2.1.3 Stakeholder expectation Validation
7	SR.2.4 Elaborate System Requirements Specification and Interfaces	2.2.2 Conversion from Expectations to Requirements
		2.2.3 Conversion from Requirements to Technical Performance Measures
8	SR.2.5 Elaborate System Elements Requirements Specifications and the System Elements Interfaces Specifications	2.3.1 Requirements Flow
9	SR.2.6 Verify and obtain Work Team (WT) agreement on the System and System Elements Requirements Specifications	2.2.4 Requirements Documentation
		2.3.2 Derived Requirements Documentation
10	SR.2.7 Validate that System Requirements Specification satisfies Stakeholders Requirements Specifications	2.2.4 Requirements Documentation
11	SR.2.8 Define or update traceability between Requirements According to the data model defined in SR.1.2, at each level of decomposition of the system	2.1.4 Stakeholder Expectation Management
12	SR.2.9 Establish or update the IVV plan and Verification, Validation Procedures for the System verification and validation.	2.3.2 Derived Requirements Documentation
		2.2.4 Requirements Documentation
		2.4.2 Design Solution Documentation

Figure 6.6: The **29110 system requirements engineering** activities linked to tailored NASA competencies elements

	SR.3 system architectural design	
	Activities	**Competency Elements**
13	SR.3.1 Document or update the Functional System Design	1.4.1 Functional Analysis
		1.4.2 Subsystem Mapping
		1.4.3 Systems Architecture Documentation
14	SR.3.2 Make trade-offs of the System Architectures	2.4.1 Alternative Designs
15	SR.3.3 Document or update the Physical System Design.	2.4.2 Design Solution Documentation
16	SR.3.4 Verify and obtain approval of the System Design.	2.4.2 Design Solution Documentation
17	SR.3.5 Establish or update the Integration plan and Integration Procedures for System integration.	2.4.2 Design Solution Documentation
18	SR.3.6 Document the *System User Manual or update the current one, if appropriate.	2.4.2 Design Solution Documentation
19	SR.3.7 Verify and obtain approval of the * System	2.4.2 Design Solution Documentation

Figure 6.7: The **29110 system architectural design** activities linked to tailored NASA competencies elements

	SR.4 system construction	
	Activities	**Competency Elements**
20	SR.4.1 Construct or update Software System Elements.	3.1.2 Product Purchase
		3.1.3 Product Fabrication
		3.1.4 Product Reuse
		3.1.5 Product Implementation Documentation
21	SR.4.2 Construct or update Hardware System Elements.	3.1.2 Product Purchase
		3.1.3 Product Fabrication
		3.1.4 Product Reuse
		3.1.5 Product Implementation Documentation
22	SR.4.3 Verify that the System Elements satisfy their System Elements Specification	3.1.2 Product Purchase
		3.1.3 Product Fabrication
		3.1.4 Product Reuse
		3.1.5 Product Implementation Documentation
23	SR.4.4 Correct the defects found until successful verification is achieved	

Figure 6.8: The **29110 system construction** activities linked to tailored NASA competencies elements

	SR.5 system integration, verification, and validation	
	Activities	**Competency Elements**
24	SR.5.1 Verify IVV plan and IVV Procedures.	3.2.1 Product Integration Preparation
25	SR.5.2 Integrate the System using System Elements (HW, HW+SW).	3.2.2 Lower Level Product Procurement
		3.2.3 Product Assembly
		3.2.4 Product Integration Documentation
26	SR.5.3 Verify the System against its Requirements	3.3.2 Product Verification Preparation
		3.3.3 Product Verification Documentation
27	SR.5.4 Validate the System against its Stakeholders Requirements	3.4.2 Product Validation Preparation
28	SR.5.5 Correct the defects found and retest to detect faults introduced by the modifications.	

Figure 6.9: The **29110 system integration, verification, and validation** activities linked to tailored NASA competencies elements

	SR.6 product delivery	
	Activities	**Competency Elements**
29	SR.6.1 Review configuration of the Product to be delivered	3.5.1 Product Transition Preparation
30	SR.6.2 Document the System Maintenance Document and *System Operation Guide or update the current one(s).	3.5.1 Product Transition Preparation
31	SR.6.3 Identify training needs and develop System User and Maintenance Training Curriculum and Material in accordance with the Project Plan.	3.5.1 Product Transition Preparation
32	SR.6.4 Verify and obtain approval of the System maintenance and operation documentation.	3.5.1 Product Transition Preparation
33	SR.6.5 Perform delivery	3.5.2 Product Transition Execution
34	SR.6.6 Transition to Manufacturing and In-service/After-sales Support.	3.6.2 Operations Execution

Figure 6.10: The **29110 product delivery** activities linked to tailored NASA competencies elements

SR.7 Additional platform related activities		
	Activities	**Competency Elements**
35	Students team leader actions	9.1.1 Mentoring and Coaching Tasks
		9.3.1 Delegating Function
		9.3.2 Influencing Role
		9.3.3 Decision-Making Solving Problems
36	Tasks execution and results upload in the platform	9.2.1 Technical Communication
		9.2.2 Reporting results
37	A various choice of tasks results types	9.2.3 Interpersonal communication
38	Lifecycle model guidance	9.3.3 9.3.3 Decision-Making Solving Problems
39	Results assessment by the educator	10.1.1 Lessons Learned Documentation
40	Coach assistance and advices	10.1.1 Lessons Learned Documentation
41	System and sub-systems shared workspaces	10.1.2 Work Products

Figure 6.11: The **29110 additional platform related** activities linked to tailored NASA competencies elements

6.6 Chapter Summary

In this chapter, we presented some of the most known systems engineering competency models, and how we used them within our proposed approach, especially how they impacted the different learning scenarios of the approach proposed in the previous chapter. Considering the fact that matching processes activities with systems engineering competencies have been done manually and following an intuitive method, we can consider in the future implementing a matching algorithm for that. The algorithm should employ text mining techniques and consider the semantic aspect of the different activities with their relationships, in order to produce a better matching results. More importantly, doing that will allow the definition of real impact of each activity in acquiring a given competency, we talk here about competencies that are linked with more than one activity.

It may be useful in the future to include the notion of "Roles" in this approach, so that the competencies can be linked to roles. Educators will be able to assign students to different roles directly within the project-team assignment. By doing so, this approach will not only help in teaching systems engineering fundamentals while simplifying the learning scenario adaptation, but it will also be useful to specify and tailor the learning goals to individuals. For example, it may be possible to focus on training system architects, system designers, etc. in the same learning environment.

7

The Supporting Platform, Experimentation and Results

7.1 Introduction

In the previous chapters we demonstrated the need to use and tailor systems engineering standardized processes for education purposes, as well as the impact of systems engineering competencies and managing them during the learning experience. We also suggested an approach of how to do so. However, to be efficient and to have an impact on the systems engineering education practices, this is not sufficient. These approaches must to be supported by means and tools to be practicable and to play a major role in an efficient educational environment. We cannot just suggest that educators should use standardized processes and tailor them to their learning goals and to their projects characteristics, without providing them with a set of tailored processes, either, because each experience and each requested systems is unique in complexity, stakeholders, goals, etc. In this chapter, we will present our proposed platform that support the approach described previously. We will also present two experimentation plans we executed to evaluate the acceptance rate of this solution, along with the obtained results

7.2 The platform

For the previous reasons, to support our proposed approach for systems engineering education, we proposed a systems engineering educational environment as a web-based platform. The proposed platform encompasses all the necessary concepts of the previously presented approach (processes-centered, competencies management, distant users, etc.). The platform is intended to be used by four different users/instances, which are:

- Education entities/organizations (organizations providing training in SE)
- Educators in the field of systems engineering
- Students in the field of systems engineering
- Industrial experts in the field of systems engineering, playing the role of tutors

7.2.1 Organization's features

An education organization using this platform to teach systems engineering is provided by the following features:

- Users management: The organization is responsible for creating and managing students, educators, and industrial experts accounts, within the platform. The organization is also responsible for the creation and the management of its different academic departments and specialties, and for assigning students and educators to the adequate departments. See Appendix A, Figure A.1. for all features available to the organization.
- Competency management: The second important role of the organization is the creation and the management of a systems engineering competency model that will be used for managing students competencies and learning scenarios. The organization is provided with an interface to define its competency model, or more than one competency model. It can modify or duplicate them to instantiate a new ones, and delete a competency model. When creating a new competency model, the organization should follow the the structure described in Figure 6.2. The interface for

adding a competencies to a competency model is illustrated in Appendix A, Figure A.2.

Figure A.3 displays the organization competency model, tailored from the NASA competency model, and shows the "Mission Needs Statement" Competency with two added competency elements. This competency is part of the " Concepts and Architecture" competency area.

7.2.2 Educator's features

The processes management system

The proposed solution provides a processes management system that allows educators to define systems engineering processes (standards or not, and mainly technical processes). These processes will be used by the educators to define the lifecycle model to be followed by teams during their projects execution. Interfaces are available to create processes, modify existing ones, and duplicate others for tailoring purposes. The processes management system main interface is available in Appendix A, Figure A.4.

- **Create new processes:** process creation consists in defining the goal of the process, its outcomes, its activities and tasks, according to standards recommendations and/or project requirements. In addition, it is possible to link a set of competencies of the organization competency model to the process activities. See Appendix A, Figure A.5.
- **Duplicate and personalize existent processes:** by adapting an existent process, title, purpose, tasks, activities, resources, and outcomes to meet a specific situation. The competencies should equally be adapted.

The projects management system

We distinguish three kinds of projects: **active projects** are projects where students are assigned to, **pending projects** are projects created by the educator but not yet assigned to students, and closed **projects** are submitted by students for evaluation and closed by educators. Educators have a component allowing them to manage their projects in the platform:

- **Create new projects:** By defining the project name, description, resources, duration, and level. The educator can also attach files for a more detailed description of the project. The project contains the requested system to be engineered by the students. The interface of the project definition can be found in Appendix A, Figure A.6.

  Another important task during the project definition scenario is the Lifecycle Model definition. This task consists in defining the set of systems engineering processes (standards or not) to be followed by the student during a specific project execution. The processes can be tailored by the educator, when defining the lifecycle or through the processes management system. See Appendix A, Figure A.7 for the real time processes tailoring interface.

  In fact, this only represents one way of defining the system lifecycle model for a specific project. In an additional proposed feature, the educators should only specify the competencies they want students to acquire after engineering the requested system, and the lifecycle model will be automatically generated, as seen in Appendix

A, Figure A.8. The resulted lifecycle model should also be managed, for example if a selected competency is attached to more than one versions of the same process (a process and its tailored one for example, as seen in Figure A.9, Appendix A), the educator should decide which one to remove, and which one to use/tailor.

Teams management system :

- **Assign a project to a team of students** A project can be assigned to one or multiple teams. The project-team assignment is done as illustrated in Figure A.10, in Appendix A.

- **Follow teams progress within an assigned project** In order to support students in their work and to assess their results, educators have access to the students workspace of any active project for which they are responsible. The workspace visualization from an educator's perspective is illustrated in Figure A.11 of Appendix A.

7.2.3 Student's features

Team-project engineering workspace

This is the most important part of the student space in our platform, and the most important part of the entire platform. It allows students of the same team to consult the details of their project, and to pass through the lifecycle model, defined by the educator, in order to engineer the requested system. Its main interface is shown in Appendix A, Figure A.12, and its elements are illustrated in Figure 7.1. of the next chapter.

Process access within team-project workspace

Students and educators can access each process from the lifecycle, in the left down menu, and they have all the necessary information about it, as illustrated in Appendix A, Figure A.13.

Tasks results uploading system

Inside the different processes that form the lifecycle model, and within each activity, students can consult the associated tasks to be performed. After performing the tasks, they upload the acquired results in the specific space as shown in Figure A.14 of Appendix A.

System structural architecture definition

Another feature of the students workspace is the system structure definition. As this platform is meant to deal with systems engineering education, and as a system is a set of system elements in interaction, we provide students with the ability to divide the requested system into multiple subsystems and system elements. For the time being, the hierarchy is limited two levels: level 1 is the division of the system into different subsystems, and level 2 is the division of the subsystems into system elements. The interface is illustrated in Appendix A, Figure A.15. Each subsystem and system element has his own engineering workspace, including the lifecycle model, so that students can switch between the system, subsystems and system elements at any time, by going to the "Structure" button in the menu.

7.3 Experimentation and results

The proposed solution has to be evaluated with regard to multiple aspects. The first and most important element aspect is its ability to allow students to get familiar with systems engineering fundamental principles. Then, it would be interesting to know more about the potential of this solution in learning soft-skills, such as communication and team-work, especially in a geo-distributed and culturally-diversified context. Through the following experimentation plans, we also hope to identify and distinguish the advantages of Systems Engineering Standards integration, technology related benefits, and the learning context impact. Finally, we aim to understand the key of success of this solution compared to other traditional teaching programs in systems engineering based on systems engineering academic and industrial experts reviews.

7.4 First Plan: Educators and students evaluation of the platform

In this section, we discuss the acceptance of our solution by the SE academic community. We targeted a specific public within this community with a presentation of the solution and a survey.

7.4.1 Methodology

The targeted public were students and tutors participating in the 2016 Robafis challenge. Organized since 2006 by AFIS, the French chapter of INCOSE, the Robafis challenge is a yearly student competition for robot design [109], whose main goal is the promotion of SE. About ten student teams from French universities and engineering schools participate in this competition. Each team can consult a SE teacher, and they can also question AFIS experts. The road-map starts about eight months before the final stage competition, when AFIS communicates the general schedule, the regulations, specifications, and a reference development document. Three months before the final stage, the teams register and receive a LEGO Mindstorms Robotics kit, in order to physically implement their solution. Fifteen days before the final stage, the teams send their development document to systems engineering experts for evaluation. The competition concludes by a final stage where all teams meet and operationally validate their works, along with project and configuration audits. Few weeks after the competition, students receive a detailed debrief regarding their work.

In 2016, eight student teams participated in the Robafis challenge. Some mixed teams featured students from several engineering schools. We got in touch with four teams, totaling in 25 students and 9 tutors. The tutors are mainly educators in the field of SE. Since the solution was not shared for public, we produced a video tutorial showing the first version of the platform at work (ProCASEE) and explained its most relevant features. The video was shared with our targeted public, together with a questionnaire they had to answer. The video can be viewed here [128], and the questionnaire used for the educators can be found in Appendix B. Note that there are not many differences between the two questionnaires, as we were mainly interested in feedback regarding the implemented/to-be-implemented features of the solution. The few differences will be apparent during the result analysis.

We received responses from ten students and six tutors, which represents a response rate of respectively 40% and 66%. The feedback from both students and tutors showed a high interest in the features of this solution.

7.4.2 Respondents profile

Nine students out of ten who answered this questionnaire are undergraduate students, whereas the last one is a post-graduate student. 70% of students estimate that they are beginners in SE, and the most usual way of learning SE were university lectures for 70%, and academic project-based learning for 60%. It appears from their responses that most of them have an overview of different topics of SE, with a focus on three topics, for which more than 90% think that their level is between medium and good: *"Design, Analysis, and Implementation"*, *"Operation and Management"*, and *"Technical Management"*. Also, 80% think they are good, or at least having a medium level in "*Requirements Management*", "*Architecting*", and "*Project-Enabling Management*".

Tutors respondents are mainly academic SE practitioners, exercising in this fields for more than two years for 67% among them, and more than five years for 17%. 50% used to teach and promote SE through *"university lectures"*, *"competitions and challenges organization"*, and *"academic or academy-industry project-based learning"*.

7.4.3 Results from the students perspective

The following list itemizes the received student feedback concerning

- **The usefulness of the current features of the solution:** The two features that appear not to be very useful, from a students perspective, are *"the ability to add resources to different processes"*, and *"reviewing the 3D design of the system"*, where only 50% estimate that they are useful or necessary. All the other features are considered useful by at least, 60 to 70% percent. For details, please refer to Table 7.1.

Table 7.1: Students appreciation about current features usefulness

Current features usefulness	**Mandatory** %	**Very Useful** %	**Useful** %	**Not very interesting** %	**Needs Improvements** %	**Not useful at all** %
Enabling geo-distributed students to work together	10	10	50	10	20	00
Learning through SE processes	10	50	00	10	30	00
Reporting all tasks results in the same shared space	10	40	20	10	20	00
Virtual 3D design of the system	00	30	20	10	30	10
Processes presentation	10	30	30	10	20	00
Processes related resources	10	20	20	30	20	00

- **The ease-of-use of the current features:** Most features are considered simple or very simple to use. About 70% to 90% of students share the same opinion about all features, except for the *"virtual 3D design component"*, where only 50% think that it is simple to use, while the other 50% rate this feature to be hard or very hard. See table 7.2.

Table 7.2: Students appreciation of current features ease-of-use

Current features ease-of-use	**Very simple** %	**Simple** %	**Pretty hard** %	**Very hard** %
Enabling geo-distributed students to work together	10	60	30	00
Reporting all tasks results in the same shared space	30	60	10	00
Reviewing the 3D design of the system	00	50	40	10
Notification management system	10	60	30	00
Processes presentation	20	50	30	00

- **The additional features to be implemented:** Table 7.3 shows that the suggested features which we plan to add to this solution, will be very useful, if not obligatory, except maybe the chat and video-call systems.

Table 7.3: Students appreciation of additional features usefulness

Additional features usefulness	**Mandatory**	**Useful**	**Not Useful**
Chat system	40	20	40
Video-call system	10	40	50
Assisting student through annotations	30	60	10
Engineering at both system and subsystems level	30	70	00
Tasks management	50	40	10
Direct web-access to SE tools	20	70	10
Direct web-access to learning resources and platforms	10	70	20
Downloading a project synthesis, at any moment	50	50	00

- **Students evaluation methods:** Students think that they should be evaluated in the context of learning SE, using different methods at the same time. 50% of students agree on evaluating them by the educator throughout the project (evaluation of processes execution quality), and regarding the acquired skills and knowledge (using questionnaires, for example). However, only 40% of the respondents approve of self evaluation methods and final results evaluation. Regarding peer evaluation techniques, only 30% believe that this is a good way of evaluation.

- **Advantages of this solution compared to their traditional way to learn SE**: The ease of use, the implementation of the project-based learning approach, and the use of SE standard processes are the most appreciated features of this solution (respectively by 50%, 50%, and 40% of the responses), followed by the ability to evaluate students regarding different metrics (30%).

7.4.4 Results from the educators perspective

- **Current features usefulness and ease-of-use:** Educators think that all features are useful without any exception. However, they showed a special interest in the processes, lifecycle and projects management systems, the ability to learn through real SE processes, and the ability to supervise students performing tasks in one shared space. See table 7.4. Educators are also unanimous about the ease-of-use of the current features, see Table 7.5.

Table 7.4: Educators appreciation about current features usefulness

Current features usefulness	**Mandatory** %	**Very Useful** %	**Useful** %	**Moderately useful** %	**Not useful at all** %	**Need Improvements** %
Processes management system	17	33	50	00	00	00
Adding resources to processes	00	50	50	00	00	00
Projects creation and management	17	50	33	00	00	00
Life-cycle model definition	00	50	50	00	00	00
Notification management system	00	00	83	17	00	00
Enabling the use of SE processes	00	50	33	17	00	00
Reviewing the 3D design	00	17	67	17	00	00
Supervising one shared space	17	33	50	00	00	00
Enabling distributed engineering	00	33	67	00	00	00

Table 7.5: Educators appreciation of current features ease-of-use

Current features ease-of-use	**Very simple** %	**Simple** %	**Pretty hard** %	**Very hard** %
Processes management system	17	50	33	00
Adding related resources to processes	17	50	33	00
Projects creation and management	17	83	00	00
Life-cycle model definition	17	50	33	00
Notification management system	17	50	33	00
Reviewing the 3D design of the system	33	33	33	00
Supervising one shared space	33	33	33	00
Enabling distributed engineering	67	17	17	00

- **Additional features usefulness:** Regarding the proposed additional features, educators mostly agreed on their utility. According to their responses, the most important extensions are assisting students throughout the execution of SE processes, the ability to consider and engineer subsystems as a system, and students managing their tasks.

Table 7.6: Educators appreciation of additional features usefulness

Additional features usefulness	**Mandatory** %	**Useful** %	**Not Useful** %
Chat system	00	83	17
Video-call system	00	83	17
Assisting student through annotations	33	67	00
Engineering at both system and subsystems level	33	67	00
Tasks management	17	83	00
Direct web-access to SE tools	33	50	17
Direct web-access to learning resources and platforms	33	33	33
Downloading a project synthesis, at any moment	17	50	33

- **Students evaluation methods:** Educators also believe that students should be evaluated in the context of learning SE, using different methods and metrics at the same time. The most expected method by educators is the ability to assess students throughout the project execution process (the engineering of the system), recommended by about 83% of them. This mainly represents the evaluation of SE processes execution quality. The second method favored by about 67% is student evaluation regarding the acquired knowledge and skills, by using surveys at the end of the project, but also by extracting useful information from the learning process. The third method recommended by 50% of educators, was self-evaluation and evaluation of the final results in order to see if they match with the starting requirements. Just as students, only 33% of educators approved of peer evaluation.

- **The advantages of this solution compared to their traditional way to teach SE:** Unlike students, educators did not think that the ease-of-use and the project-based learning approach were the most important aspect that differentiate this solution from traditional SE teaching. However, 50% of educators appreciate the use of SE standard processes and the ability to evaluate students using different metrics. More importantly, about 67% declare that the ability to manage the systems life-cycle model is a good idea, along with the ability to manage geo-distributed students.

7.5 Second Plan: Students experimentation of the solution

The goal of this experimentation plan is apply our solution by students in a real context of use, in the scope of a training course on systems engineering.

7.5.1 Methodology

We launched an introductory course to systems engineering at Paris Mechanical Engineering School (Supméca) as a doctoral training for PhD students. PhD students from different doctoral schools at UPSaclay (Paris Saclay University) were invited. The course took place during 3 days, between May 9th, 2017 and May 11th, 2017. We had nine students participating, eight PhD students and one engineering student starting his internship in the field of systems engineering. The goal of the training course was to introduce students to the discipline of systems engineering, and to deepen their knowledge in three selected topics of systems engineering: the stakeholders needs and requirements definition, the systems requirement definition, and the systems architecture definition. This course was based on systems engineering principles and processes from the ISO/IEC/IEEE 15288.

The course was organized into theoretical and practical sessions: during the **theoretical sessions** the educator explained systems engineering fundamental principles and presented the three ISO/IEC/IEEE 15288 processes in relation to the previous three selected topics, while describing their activities and the expected outcomes from their execution. During the **practical sessions**, students worked in teams of two individuals (an exception was made for one team who had three members including two PhD students and the engineering student). The objective of the practical sessions was to engineer the requested systems, while considering its lifecycle model as defined by the educator, and executing the corresponding processes. During the practical sessions students had access to our proposed solution.

7.5.2 Course Outline

The following 11 steps explain how the course took place.

1. Prior to the course start, two systems engineering projects were created on the proposed web platform. The first one asked for the design of an "inspection drone" and the second one for a "surveillance drone".

2. The educator started the course by an introduction to systems engineering during the morning of the first day.

3. The web platform was presented during the beginning of the first afternoon, and they received their logins and passwords at the same time.

4. The two projects were introduced, and each student team selected its preferred project.

5. Teams-project assignments were done.

6. The educator provided a tutorial about different SysML Diagrams that had to be used during the course, and made them practice some examples (the course is centered around Model Based Systems Engineering).

7. During the same afternoon the educator introduced a tailored version of the *"stakeholder needs and requirements definition process"* from the ISO/IEC/IEEE 15288 standard, and asked students to perform its related activities in the context in their project. The tailored processes can be found in Section 7.4.3. The activities were mainly performed using SysML Language and MagicDraw as a supporting tool. For this reason, each time a new activity was presented, the educator gave a demonstration of the expected results (SysML Diagrams) applied to a sample project. The different processes activities and tasks had an associated text explaining how to perform each step of the process and with which tools.

8. The teams executed the different activities and reported their results to the dedicated tasks and activities space, inside the team-project workspace, using the provided web platform.

9. The educator had a direct access to the teams workspaces, in order to monitor their progress and assist them if necessary.

10. During the second and the third days, the same actions were performed respectively regarding the *"systems requirement definition process"* and the *"systems architecture definition process"*.

11. At the end of the third and last day of the course, students were given a two parts survey to fill out. The first part of questions addresses the usefulness and the ease-of-use of the current features, as well as the usefulness of some additional planned features. This questionnaire can be found in Appendix C. The second part contains twelve questions about the used systems engineering processes, and aims to check students acquired knowledge regarding systems engineering principles and processes. These questions are presented in Section 7.4.4.

7.5.3 The used tailored systems engineering processes

We present the set of three processes from the ISO/IEC/IEEE systems engineering standard [4], tailored to meet the course project requirements. The tailoring operations used for this purpose concern the reduction of the number of activities and tasks (i.e the educator decided not to use some of the standardized activities and tasks related to these processes).

Stakeholders needs and requirements definition process

- Activity 1.0: Prepare for stakeholder needs and requirements definition
  - Task 1.1: Identify the stakeholders who have an interest in the system throughout its lifecycle.
- Activity 2.0: Define stakeholder needs.
  - Task 2.1: Define context of use.
  - Task 2.2: Identify stakeholder needs.
  - Task 2.3: Prioritize and down-select needs.
  - Task 2.4: Define stakeholder needs and rationale.
- Activity 3.0: Develop the operational concept and other lifecycle concepts.
  - Task 3.1: Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other lifecycle concepts.
  - Task 3.2: Identify the interaction between users and the system.
- Activity 4.0: Transform stakeholder needs into stakeholder requirements.
  - Task 4.1: Identify the constraints on a system solution.
  - Task 4.2: Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health.
- Activity 5.0: Analyze stakeholder requirements.
  - Task 5.1: Analyze the complete set or stakeholder requirements.
  - Task 5.2: Define critical performance measures that enable the assessment or technical achievement.

System requirements definition process

- Activity 1.0: Prepare for system requirements definition.
  - Task 1.1: Define the functional boundary of the system in terms of behavior and properties to be provided.
- Activity 2.0: Define system requirements.
  - Task 2.1: Define each function that the system is required to perform.
  - Task 2.2: Define necessary implementation constraints.
  - Task 2.3: Identify system requirements that relate to risks, criticality of the system, or critical quality characteristics.
  - Task 2.4: Define system requirements and rationale.
- Activity 3.0: Analyze system requirements.
  - Task 3.1: Analyze the complete set of system requirements.

System architecture definition process

- Activity 1.0: Prepare for architecture definition.
  - Task 1.1: Review pertinent information and identify key drivers of the architecture.
  - Task 1.2: Identify stakeholder concerns.
  - Task 1.3: Define evaluation criteria based on stakeholder concerns and key requirements.
- Activity 2.0: Develop architecture viewpoints.
  - Task 2.1: Select, adapt, or develop viewpoints and models based on stakeholder concerns.
  - Task 2.2: Establish or identify potential architecture framework(s) to be used in developing models and views.
  - Task 2.3: Capture rationale for selection of framework(s), viewpoints and model types.
  - Task 2.4: Select or develop supporting modeling techniques and tools.
- Activity 3.0: Develop models and views or candidate architectures.
  - Task 3.1: Define system context and boundaries in terms of interfaces and interactions with external entities.
  - Task 3.2: Identify architectural entities and relationships between entities that address key stakeholder concerns and critical system requirements.
  - Task 3.3: Allocate concepts, properties. characteristics. behaviors, functions, or constraints that are significant to architecture decisions or to architectural entities.

The system lifecycle model within teams shared workspace had this exact structure, and the students uploaded their results to each of the cited tasks. The configuration and the different components of the team's shared workspace is shown in Figure 7.1.

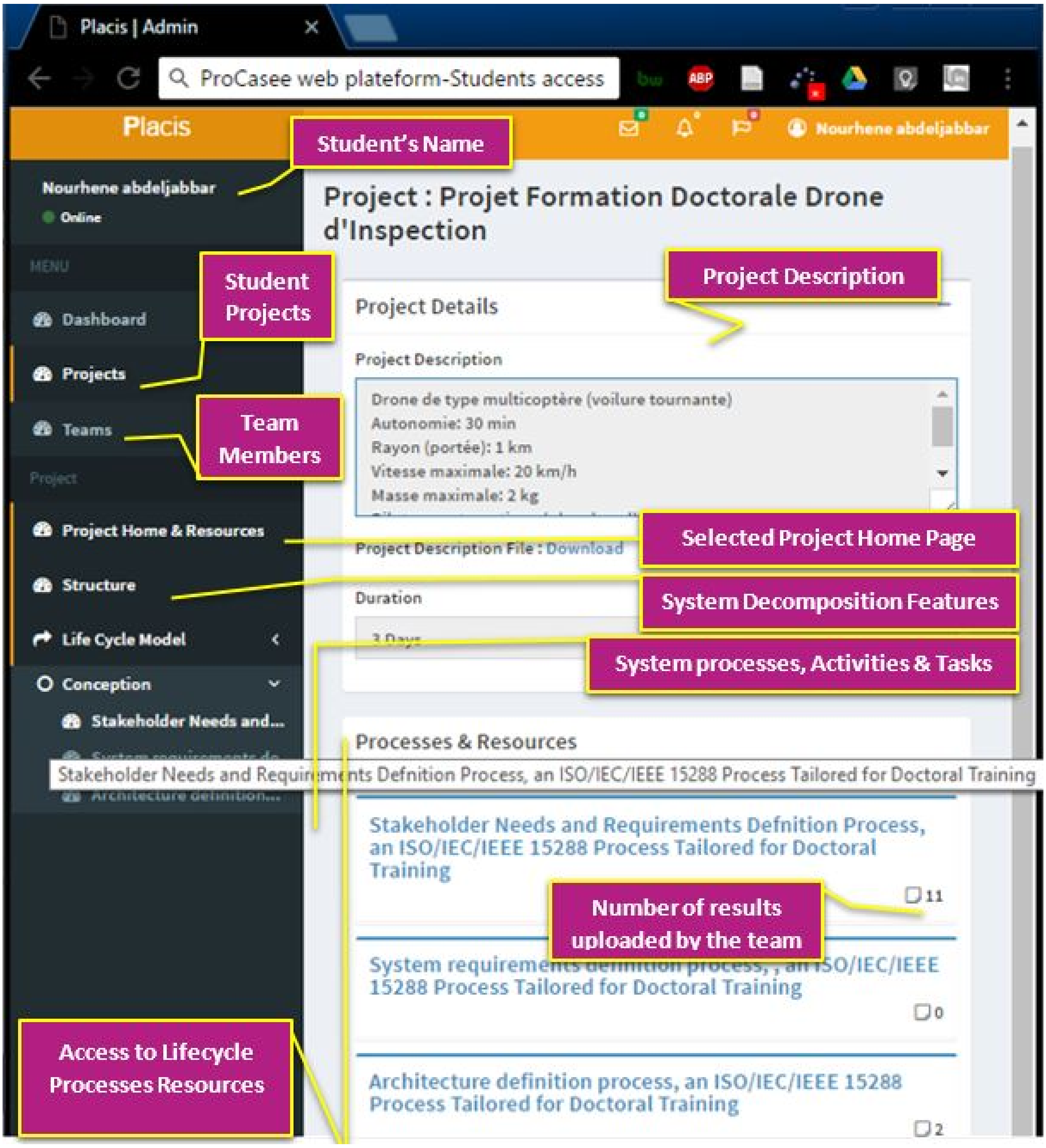

Figure 7.1: Team's shared workspace interface

7.5.4 The SE knowledge quiz questions

Students were asked to answer the following questions, built on the same format as the INCOSE certification knowledge exam:

Question 1: Systems engineering should be applied to:

- System conception, design, and implementation stages
- System requirements, verification and validation stages
- The entire lifecycle of a system

Question 2: Which systems engineering standard deals with the entire lifecycle of a system ?

- IEEE 1220
- EIA 632
- ISO/IEC/IEEE 15288

Question 3: Requirements should be S.M.A.R.T, what does this mean ?

- Specific, Measurable, Achievable, Relevant, Traceable
- Special, Mature, Achievable, Redundant, Traceable
- Semantic, Mature, Analyzable, Relevant, Traceable

Question 4: The purpose of the system requirements definition process is to transform the stakeholder, user oriented view of desired capabilities, into a technical view of a solution that meets the operational needs of the user.

- True
- False

Question 5: The results of the architecture definition process are widely used across the lifecycle processes:

- Yes
- No

Question 6: One of the purposes of the architecture definition process is to express the selected architecture alternative(s) in a set of consistent views.

- Yes
- No

Question 7: Analyzing system requirements is an activity of:

- The system architecture definition process
- The system requirements definition process
- The stakeholder needs and requirements definition process

Question 8: Which TWO among the next statements are OUTCOMES of the system requirements definition process ?

- System requirements analysis results
- Development of architecture viewpoints
- System stakeholders definition
- Traceability of systems requirements to stakeholder requirements

Question 9: Which THREE among the following activities are part of the architecture definition process ?

- Assess architecture candidates

- Develop models and views of candidate architectures
- Manage the design
- Prepare for architecture definition

Question 10: Which one is NOT the OUTCOME of the system requirements definition process ?

- The system description, including system interfaces, functions and boundaries, for a system solution are defined
- System requirements analysis
- Identified stakeholder concerns are addressed by the architecture

Question 11: Iteration of the architecture definition process with the business or mission analysis process, system requirements definition process, design definition process, and stakeholder needs and requirements definition process is often employed to ensure:

- A negotiated understanding of the problem to be solved and the identification of a satisfactory solution
- The system will be realized in a minimum amount of time
- A good system delivery for at least one principal stakeholder

Question 12: Which TWO among the next statements are GOALS of the stakeholder needs and requirements definition process?

- Identification of stakeholders involved throughout the entire lifecycle
- Define the business or mission problem or opportunity
- Convert stakeholder needs into stakeholder requirements definition
- Determine potential solution classes

7.5.5 Survey results

Part One: Proposed platform features

- **The usefulness of the current features of the solution:** The current features of the solution are considered useful or very useful by 90% to 100% of students . An exception is made for the "process related resources", where about 33% think that they are moderately useful. The verdict about process related resources will be confirmed later where we present the students' wishes about the resources they want to see in the platform.

Table 7.7: Students appreciation about current features usefulness

Current features usefulness	**Very Useful** %	**Useful** %	**Moderately useful** %	**Useless** %
Enabling geo-distributed students to work together	3	67	0	0
Learning by executing the Standard Systems Engineering Processes	22	67	11	0
Reporting all tasks results in the same shared space	56	44	0	0
Processes related resources	11	56	33	0
System structure creation and manipulation	44	44	11	0
Accessing and reviewing other team members uploaded results	44	56	0	0

- **The ease-of-use of the current features:**

  Three out of five features that students were asked about, are considered simple or very simple to use. As to the two others, which are the "processes presentation" and the "navigation between their description, activities and tasks", these were seen as moderately simple to use. As an explanation, it was difficult to rapidly find a task or an activity. We needed to change the design and the ergonomy of the processes content presentation and interactions with it.

Table 7.8: Students appreciation of current features ease-of-use

Current features ease-of-use	**Very simple** %	**Simple** %	**Moderatly Simple** %	**Hard** %
Accessing and reviewing other team members uploaded results	56	33	11	0
Reporting all tasks results in the same shared space	22	67	11	0
Processes presentation	11	33	56	0
Navigating through processes description	11	22	67	0
System structure creation and manipulation	11	78	11	0

- **The additional features to be implemented:** It appears that three additional features out of the suggested four were seen as useful or very useful. However, only five persons out of nine thought it would be useful to add a feature that allows educators to assist students by adding notes about the tasks being realized.

The reason for that is obvious. This feature is intended to be useful when the solution is used in a geo-distributed context, where students from the same team and/or the educator work in different locations with a minimum or no face-to-face communication. However, this was not the case during this experimentation plan. Students and the educator were in the same classroom, and students were directly assisted by the educator. The students could not fully appreciate the usefulness of being assisted by the educator making notes from inside the platform.

Table 7.9: Students appreciation of additional features usefulness

Additional features usefulness	**Very Useful** %	**Useful** %	**Moderately useful** %	**Useless** %
Allowing educators to assist students by adding notes about the tasks being realized, during their execution.	33	22	45	0
Adding a feature that enables students to manage and assign the different tasks to the different students of the team.	22	56	11	11
Adding a feature that enables students getting direct web based access to systems engineering tools (such as SysML Tools)	45	33	22	0
Giving the ability to students to download at any time, a synthesis of the project as a Pdf File	89	11	0	0

- **The most important features of the proposed solution**

  In the previous analysis of the responses, students did not seem to see the importance of being assisted by the educator from inside the platform. This however does not mean that they are not aware of the importance of the platform as a whole for managing a geo-distributed learning experience. About 57% responded to the question about "the most important features of the solution" by "the management of geo-distributed students" (first place). This result is confirmed by students responses to the first question concerning the usefulness of the different features. In fact 100% of them responded by useful or very useful to the three features in relation to managing the geo-distributed work dimension of the platform. These three features are: "enabling geo-distributed students to work together", "reporting all tasks results in the same shared space", and "accessing and reviewing other team members uploaded results".

Table 7.10: Most important features

Features	**Importance (%)**
The ease of use of the solution	33
The management of geo-distributed students	56
The implementation of the Project-Based Learning approach	22
The use of systems engineering standard processes	44
The ability to evaluate students regarding different metrics	44
The ability to upload results and share them with others at the same shared space	44
The ability to manage the system structure (different sub-systems and system-elements)	33

- **Requested additional pedagogical resources**

  In addition to the current pedagogical resources, provided within the different lifecycle processes, the most desired two additional pedagogical resources are: a "sample use case system" of engineering results uploaded over the entire lifecycle tasks, to give students examples of what the expected results should look like (requested by eight students of nine). The second desired resource is adding metrics that show the project progress on the project home page (requested by six students our of nine).

Table 7.11: Requested pedagogical resources

Pedagogical resource	**Requests (%)**
More explanations about the different processes and activities	33
More guidance about which tools and methodologies to use for each task	33
"A sample use case" of systems engineering results uploaded over the entire lifecycle tasks	89
Information about the competencies to be learned after executing each activity or process	33
An overview about the entire set of standard processes	22
Adding metrics that show the project progress on the project home page	67

- **Students evaluation methods:** Students were unanimous regarding the best way to evaluate them. About 89% among them prefer to be evaluated by the educator regarding the final results, followed by 22% regarding both processes execution quality, and the acquired skills and knowledge. They thereby joined the opinion of the first plan students. As a reminder, most of students (about 50%) agreed that they should be evaluated by the educator, too, throughout the project (evaluation of processes execution quality), and regarding the acquired skills and knowledge (using questionnaires, for example).

Part Two: Systems Engineering Acquired Knowledge

Student responses to the twelve questions: The correct answers vary from one question to another. The questions are responded correctly by 44% to 100%. The worse result of 44% of correct answers is related to questions 7, 9, and 10, followed by questions 4, 5, and 8, which have been answered correctly by 56%. However, these results show that an average of 68% of questions were answered right, in addition to 8 percent of the remaining questions which were partially answered. Partially answered means that the question had multiple responses, and the student selected less correct answers. The average of "correct responses by student" is about 66% (8 responses/12).

Table 7.12: Student responses to each of the knowledge acquisition test

Additional features usefulness	**True Responses** %	**False Responses** %	**Partial Responses** %
Question 1	89	11	0
Question 2	100	0	0
Question 3	78	22	0
Question 4	56	44	0
Question 5	56	44	0
Question 6	78	22	0
Question 7	44	56	0
Question 8	56	11	33
Question 9	44	0	56
Question 10	44	56	0
Question 11	78	22	0
Question 12	89	0	11
Average	68	24	8

Table 7.13: Number of correct answers given by each student

Student	Number of correct responses
Student 1	7/12
Student 2	7/12
Student 3	9/12
Student 4	10/12
Student 5	7/12
Student 6	12/12
Student 7	5/12
Student 8	7/12
Student 9	7/12
Average	7.9/12 (65.8 %)

7.6 Chapter Summary

In this chapter, we presented the second and enhanced version of our approach and the web-based platform. This version is characterized by the consideration and implementation of the system structure creation module, generating a shared workspace for each subsystem and system element. It also includes the competencies management system, and the coach (educator or industrial expert) implication during the learning scenario. In addition, it withdraws the development efforts of the 3D virtual assembly environment. Note that the competencies management system is integrated as an independent module, so that organizations and educators can decide to use it or not, depending on their goals.

The two experimentation plans demonstrated a high level of acceptance of the solution, regarding its usefulness, ease-of-use and the need for the suggested additional features. However, because of the low sample size, the results should be verified in the future by a larger study, within real conditions of use, implicating a significant number of students working in a distant teams environment, while being assisted by industrial coaches. A good way to do that would be the use of the solution in the Robafis [109] challenge. Finally, if we had to keep in mind only one thing from these experimentation, we should consider the implementation of a new feature allowing the educators to easily propose a "sample project", going through all selected systems engineering processes, so that students better understand the expected outcomes of each process activity.

8 General conclusion and Perspectives

8.1 General conclusion

After performing an in-depth investigation about the current practices of systems engineering education within industrials and organizations dealing with systems engineering, and within academia, we understood that, in order to provide the most benefits in a systems engineering educational environment, we have to consider different points. We presented most of these points in Table 4.1. (Chapter 4). The most relevant points for efficient learning experiences in systems engineering, including soft skills and systems thinking development, are:

- Systems engineering is better taught through real experiences, and at least within a distant project based learning approach. Students should learn systems engineering in reality like environments and constraints.
- Educators and industrial experts will be of good assistance to the engineering teams, applying systems engineering principles to the learning project teams.
- Digital environments can facilitate and improve the achievement of the previous requirements if well conceived and implemented, while respecting the characteristics of this discipline.
- Systems engineering processes (especially standardized one) should take a significant place, and more, they should guide the learning path, while still giving students freedom to develop their critical thinking and systems thinking skills.
- One of the main outcomes of a good education solution for systems engineering resides in its ability to make students understand and practice the ideas of wholeness and interrelationships, and to experience at each step that they are part of a team, and that their system is part of a whole.

By taking into consideration the entire set of characteristics, and in order to create an efficient solution (an approach with its supporting platform) for systems engineering education, we presented our vision of a good environment. This environment should make use of disruptive technologies and practices. For feasibility reasons, we decided to focus only on the most important part of that vision, i.e. to go further in conceiving, designing and producing the Collaborative Virtual Environment (CVE) which became our solution for systems engineering education.

It is particularly interesting to assess how we responded to our four initial research questions presented in chapter 1.

First research question:

The proposed solution is based on the use of systems engineering processes in a new way, by simplifying their adaptation, and by making them the essence of the learning scenarios for both students and educators. This helps managing the complexity of the learning process without limiting students and educators to one kind of tools or methods, as we profoundly agree with the diversity and the high level of uncertainty of programs, projects or systems, which need the use of systems engineering. This uncertainty often applies to their lifecycle processes, too.

The main advantage of the proposed solution is its ability of adaptation. It is built in a way that enables educators to create and control their own content and learning scenarios. It allows easy adaptation of the learning processes to the students' level (acquired competencies), the learning goals (competencies to be acquired), and the project complexity (expected number of system elements, and number of disciplines and stakeholders involved). In fact, it can be used by different educators and organizations depending on their needs. No limits exist regarding to which technical discipline should use it. It can be used within any academic or industrial organization having an interest in educating people in systems engineering, within a PBL environments, distant or not, and independently from their specialization, processes, tools, methodologies, etc. This fact brings us to highlight another kind of adaptation ability. The proposed solution is domain-independent, standard-independent, methodology-independent, competency models-independent, and tools independent. This enhances its ability to be rapidly updated, expanded, and adapted as needed to meet different learning constraints, and to accept any kind of tools and methodologies, so that each institution can find its interest in using it. Organizations and educators can even use their domain-specific processes, in addition to their tailored generic/standardized processes.

The last kind of adaptation ability resides in targeting different levels of systems engineering education. It can be used to provide students with a high level overview of systems engineering and its entire lifecycle processes. Also, it can be used to deepen students knowledge in a specific topic of systems engineering. More than that, it can be used to deepen the knowledge of one team in a specific systems engineering topic, while deepening the knowledge of another team working on the same project in another topic. This can be done by having each team focus on a specific topic and specific systems engineering processes. It also can deepen interpersonal and systems thinking skills by making students collaborate in the context of the same projects.

The main lesson we can learn from this analysis, is that the proposed process-centered approach for SE education responds clearly to the expectations illustrated in our first research question. In fact, this approach suggests a good way to incorporate SE processes within the learning scenario, by taking all their advantages and without giving up the constraints of systems engineering education. These constraints are: for students, the ability to acquire the necessary critical thinking, soft skills, the holistic view capabilities when engineering a system, and to master the different SE principles incorporated within the used processes. And for educators, the ability to manage the size of their learning projects, to adapt the adopted processes to their project size and their learning goals, and to ensure that the processes are really executed as intended, and that the student really learned the fundamental and critical principles of SE. To better ensure the learning outcomes, especially regarding considering the system as a whole and the importance of understanding the interrelationships, we added a critical feature, "the system structure creation", that allows students to define their system structure, and automatically generates for the team a shared workspace for each subsystem and system element.

Second research question

The proposed approach has been supported by a web-based platform. To ensure long-term organizations satisfactions, learning processes improvements, and students SE competencies and skills management, we recommended the adoption and tailoring of SE competency models. We added a competency management system to the proposed solution. In response to our second research question, we were not able to recommend any specific competency model to be incorporated in our approach, and to be used within the proposed platform. However, we want our approach and platform to be competency models independent, therefore each organization can select and define its own competency model. For this purpose, we proposed a unified structure for the competency model to be used, with a method on how to link this competency model to the used SE processes. As an example, we presented the results of a use case, linking a tailored NASA competency model to the technical processes and activities of the 29110 series. Regarding student assessment sub-question, we can consider this by making use of competency models. Students can henceforth be better evaluated regarding their work and their acquired competencies and skills.

Third and forth research questions

In response to the third research question, and to substitute the need of on-the-job experience, the proposed solution is built upon a distant project based learning paradigm, where teams can include students from different organizations, and from different countries. One student can be assigned to more than one SE project, and a SE project can be assigned to more than one team. We have ensured that the same team members have access to the same shared workspace, and anyone can see what another uploaded as a result to a given task.

As a response to the fourth question, in addition to the educator's assistance, students can be coached throughout the different system engineering stages by industrial experts. In fact, educators and assigned industrial experts can access the shared workspace of the team at any time, and see the performed tasks and the uploaded results. They can also put notes beside any task to say if the work is well done, or if they need to do further actions.

Finally, we performed two experimentation plans which demonstrated a high level of acceptance of the solution, regarding its usefulness, ease-of-use and the needs for the suggested additional features. However, because of the low number of students and educators during the two experimentation plans, these results should be confirmed by a larger experimental study. It might be useful to also include some non systems engineers, some junior systems engineers, and some senior engineering from the industry, within this experimentation.

In addition to the previous contributions, we think that this solution can have an extraordinary impact, within two domains, in addition to SE education:

1. The first one is the ability to use it as a collaborative framework for systems engineering without educational context, especially for small entities. This can be demonstrated for example by its capacity to integrate at least the entire technical processes and activities of the ISO/IEC/IEEE 15288 standard, or the entire technical and technical management processes and activities of the ISO/IEC 29110 series.

2. The solution may have high potential to be used as a support to Project Based Learning experiences in general, in other domains than systems engineering, as it

responds to all PBL challenges cited by Dym et al. [9]. In this context, organizations can replace the competency model by those of their own discipline, and educators can model their learning projects as a set of processes and activities. Once deployed on the cloud, and once efficient assessment methods are operational, we think that this solution will have a great potential of becoming the next Open edX [12] for Project Based Learning. It can even be used as a part of existing solutions such as EdX [3] or Coursera[4], to add PBL experiences to their theoretical courses.

8.2 Perspectives

To take this approach to higher level of efficiency in addition to its ability of adaptation and use in almost any context, future efforts of improvement should start by understanding its true added value in terms of systems engineering learning outcomes. This can be done by a larger experimentation plan, including educators and students from different universities and different countries, industrials providing the systems to be engineered, systems engineering expert practitioners playing the role of coaches, and if possible different teams working on different parts of the same project.

Future efforts should also address the following key directions:

- **Effective assessment methods:** Students assessment was not a focus of this thesis, but since we cannot talk about educational environment without talking about students assessments, we took this problem into consideration throughout the design of the entire solution. The resulting solution is indeed built in a way to open doors to new assessment methods implementation, especially regarding the previously cited four elements:
  - Result Assessment: Evaluating the quality of outcomes compared to the project requirements (expressed by the teacher or an industrial). We can also call it "design assessment".
  - Execution Assessment: Evaluating how well systems engineering standard/generic processes have been applied during the project.
  - Knowledge Assessment: Evaluating the acquisition of fundamental knowledge on systems engineering, an additional planned knowledge acquisition.
  - Skills Assessment: Evaluating the acquired skills, such as collaboration, communication, team work, etc.
- **Systems engineering roles integration:** Introduce the notion of systems engineering roles to the proposed solution. These roles are commonly known and can be found in the literature. Once introduced, organizations should be able to tailor and adapt them depending on their domain of activity, and assign competencies from the adopted competency model. Then, we can conceive a new kind of learning scenario adaptation by crossing different data such as the selected student role and the project learning goals. Systems engineering roles may be defined by using/adapting the role structure of the U.S. FAA, illustrated in [70], seen in Figure 3, as a way for defining a systems engineering role profile for Senior Engineers.

[1]Open edX is an open-source platform software, used to provide MOOC services [Wikipedia]

[2]MOOC for Massive Open Online Course, is an online course aimed at unlimited participation and open access via the web.[Wikipedia]

[3]https://www.edx.org/

[4]https://www.coursera.org

- **The ability to define system structure at more than two levels:** For now, students are only able to structure their system as: a global system, subsystems, and system elements. This operation is irreversible, and it is not possible to change the structural architecture without deleting and reworking everything. It may be a good idea to keep it that way, as it will let them learn the importance of their decisions, and the fact that we cannot adapt what we want at any time of the system engineering process, without huge costs. However, when we do not want to let the learning experience be tedious and hard, we suggest that the system structure creation needs to be dynamic and allow for more than two vertical levels.

- **Sample project:** Another point to consider, and recommended by the students when they experimented the solution, is providing educators by additional features that allow them to add a simple already engineered case study, illustrating all steps of the requested system lifecycle model. This way, students will find useful information on each activity to be performed and better understand their expected outcomes. We remind that it might also be helpful to let students define the lifecycle model by themselves, in the scope of an advanced course in systems engineering.

- **3D virtual design review:** For complexity reasons, we decided to stop our development efforts on 3D virtual assembly, illustrated by Figure A.16. in appendix A. However, we think it would be very helpful if the engineering team could collaboratively access a similar environment for design review of the virtually designed 3D system. Of course, this will only be useful if the requested system itself or one of its components consist of a tangible system design.

- **Direct access to systems engineering tools:** We certainly keep our principle that "*systems engineering educational environment should be tools independent*". Therefore, we do not recommend to add a specific tool to the platform, but we suggest more efforts towards making systems engineering tools directly available on the platform. By doing so, each organization and/or educator using the platform can choose their tools of predilection. This may also open the path to other improvements, such as interoperability issues between different tools.

- **Integrating an Artificial Intelligence (AI) Engine:** First, to assist students and educators in their engineering actions, and second, to help them learn lessons from their actions.

  For example, if the AI engine detects a system with a large number of subsystems and systems elements having an insignificant time for the requirements engineering processes, it should alert the students and advise them to take more time for this critical step. It even may immediately redirect them to appropriate resources to learn more about the topic. If the students change the system architecture, the AI engine should be able to advice them to review their requirements and other elements susceptible to be affected by this change.

- **Adding additional relationships between the subsystems and system elements:** In addition to the structural relationships between the system elements, it would be helpful if the students are able to add additional data to the system elements and subsystems. For example, the students could flag different components impacted by a change in the current subsystem or system element. These additional data and relationships would be especially useful for the AI engine.

Bibliography

[1] Richard Beasley and Richard Partridge. 11.3.2 the three ts of systems engineering – trading, tailoring, and thinking. *INCOSE International Symposium*, 21(1):1356–1372, 2011.

[2] David D. Walden and Garry J. Roedler. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Wiley, no, 4 edition, 2015.

[3] J. Estefan. Survey of Model-Based Systems Engineering (MBSE) methodologies. Technical report.

[4] Iso/iec/ieee international standard - systems and software engineering – system life cycle processes. *ISO/IEC/IEEE 15288 First edition 2015-05-15*, pages 1–118, May 2015.

[5] Paul G. Walter. A model for effective systems engineering workforce development at space and naval warfare systems center (ssc) atlantic. Master's thesis, Naval Postgraduate School, Monterey, California, 9 2013.

[6] Odile Mornas, Roland Mazzella, Catherine Laporte Weywada, Anne Sigogne, and Patricia Pancher. Development of systems engineering people to support major transformation plans in thales (process, roles, methodology and related tools). *INCOSE International Symposium*, 22(1):1884–1898, 2012.

[7] Jr Andrew P. Sage, James E. Armstrong. *Introduction to Systems Engineering.* John Wiley and Sons, Inc, New York, 2000.

[8] Guey shin Chang, Horng linn Perng, and Jer nan Juang. A review of systems engineering standards and processes. *Journal of Biomechatronics Engineering*, pages 71–85, 2008.

[9] C.L. Dym, A.M. Agogino, O. Eris, D.D. Frey, and L.J. Leifer. Engineering design thinking, teaching, and learning. *Journal of Engineering Education*, 94(1):103–120, 2005.

[10] Charles S. Wasson. Ac 2012-3389: System engineering competency: The missing course in engineering education. *INCOSE International Symposium Journal*, 20(1):21–36, 2010.

[11] T. S. S. Subramanian, M. Mukundlal, P. Prempyari, and P. Dubey. Systems engineering: A new approach to engineering education in india. In *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*, pages 1–3, July 2012.

[12] Mohammed Bougaa, Stefan Bornhofen, Hubert Kadima, and Alain Rivière. Virtual reality for manufacturing engineering in the factories of the future. *Applied Mechanics & Materials*, 2015.

[13] Aaron W. Johnson, Sara Willner-Giwerc, and Paul T Grogan. Developing a systems engineering activity for middle school students using lego robotics. In *ASEE 123rd Annual Conference and Exposition*, new Orleans, LA, USA, 2016.

[14] Gerrit Muller. Didactic recommendations for education in systems engineering, 2010.

[15] Clive L Dym. Design, systems, and engineering education. *International Journal of Engineering Education*, 20(3):305–312, 2004.

[16] Caroline Twomey Lamb and Donna H. Rhodes. 9.2.3 standardized process as a tool for higher level systems thinking. *INCOSE International Symposium*, 17(1):1492–1502, 2007.

[17] Mohammed Bougaa, Stefan Bornhofen, and Alain Rivière. A survey of current practice in systems engineering education. In *44th SEFI Annual Conference*, Tampere, Finland, September 2016.

[18] Peter Senge. *The Art and Practice of the Learning Organization*. Doubleday, no, 2006.

[19] Melissa T. Greene and Panos Y. Papalambros. A cognitive framework for engineering systems thinking. In *2016 Conference on Systems Engineering Research*, 2016.

[20] Heidi L. Davidz and Deborah J. Nightingale. Enabling systems thinking to accelerate the development of senior systems engineers. *Systems Engineering*, 11(1):1–14, 1 2008.

[21] Brigite Daniel Allegro, Alain Leput, and Jean Claude Tucoulou. *Guide: Introduction au Penser Système*. AFIS, Paris, France, 2014.

[22] John Boardman, Brian Sauser, Larry John, and Robert Edson. The conceptagon: A framework for systems thinking and systems practice. San Antonio, TX, USA, 2009.

[23] Joseph E. Kasser and Moti Frank. 1.2.2 a maturity model for the competency of systems engineers. *INCOSE International Symposium*, 20(1):37–50, 2010.

[24] Ieee guide for information technology - system definition - concept of operations (conops) document. *IEEE Std 1362-1998*, pages 1–24, Dec 1998.

[25] Barry Richmond. Systems thinking: Critical thinking skills for the 1990s and beyond. *System Dynamics Review*, 9(2):113–133, 1993.

[26] BKCase. Guide to the systems engineering body of knowledge (sebok). `http://sebokwiki.org/wiki`. [Online;Accessed: 07-February-2017].

[27] Moti Frank. Engineering systems thinking: Cognitive competencies of successful systems engineers. *Procedia Computer Science*, 8:273–278, 2012.

[28] Moti Frank. Engineering systems thinking and systems thinking. *Systems Engineering*, 3(3):163–168, 2000.

[29] L. von Bertalanffy. *General System Theory: Foundations, Development, Applications*. George Braziller, New York, no, revised edition edition, 1968.

[30] John Boardman and Brian Sauser. *Systems Thinking: Coping with 21st Century Problems, Industrial Innovation Series*. Taylor and Francis, no, first edition edition, 2008.

[31] James N. Martin. 1.6.4 systems engineering is not just engineering—or is it? a critical look at the scope of our profession. *INCOSE International Symposium*, 10(1):727–734, 2000.

[32] INCOSE. What is systems engineering? `http://www.incose.org/AboutSE/WhatIsSE`. [Online;Accessed: 26-Mai-2017].

[33] M. Maier and E. Rechtin. *The Art of Systems Architecting*. CRC Press, no, second edition edition, 2000.

[34] Joseph Kasser and Angus Massie. 8.5.3 a framework for a systems engineering body of knowledge. *INCOSE International Symposium*, 11(1):841–847, 2001.

[35] Benjamin S. Blanchard and Wolter J. Fabrycky. *Systems Engineering and Analysis*. Prentice Hall, New Jersey, 1998.

[36] National Aeronautics and Space Administration. *Systems Engineering Handbook*. Nasa, no, 2007.

[37] William H. McCumber and Crystal Sloan. 1.1.3 educating systems engineers: Encouraging divergent thinking. *INCOSE International Symposium*, 12(1):8–15, 2002.

[38] Scott A. Hyer. An integrated approach to systems engineering training. *INCOSE International Symposium*, 4(1):323–330, 1994.

[39] Joseph Kasser and Derek Hitchins. 3.4.1 yes systems engineering, you are a discipline. *INCOSE International Symposium*, 22(1):416–431, 2012.

[40] Hall. A. *A methodology for systems engineering*. Princeton, NJ, no, 1962.

[41] INCOSE. History of systems engineering. `http://www.incose.org/AboutSE/history-of-systems-engineering`. [Online;Accessed: 26-Mai-2017].

[42] Whitcomb Clifford, White Corina, and Khan Rabia. Development of a system engineering competency career model: verification of the model using opm method. `https://calhoun.nps.edu/handle/10945/47492`. [Online;Accessed: 07-February-2017].

[43] MITRE Corporation. *Systems Engineering Guide*. MITRE, no, 2014.

[44] INCOSE. Systems engineering vision 2025. `http://www.incose.org/AboutSE/sevision`. [Online;Accessed: 26-Mai-2017].

[45] A. Ei. *ANSI/EIA-632 Standard, Processes for Engineering a System.* EIA, 1998.

[46] Ieee trial-use standard for application and management of the system engineering process. *IEEE Std 1220-1994*, pages i–, 1995.

[47] Ieee standard for application and management of the systems engineering process. *IEEE Std 1220-1998*, pages i–, 1999.

[48] Iso/iec/ieee international standard – systems engineering – system life cycle processes. *ISO/IEC 15288 First edition 2002-11-01*, pages 1–70, Jan 2002.

[49] Ieee standard for application and management of the systems engineering process. *IEEE Std 1220-2005 (Revision of IEEE Std 1220-1998)*, pages 1–87, 2005.

[50] Iso/iec/ieee systems and software engineering - system life cycle processes. *IEEE STD 15288-2008*, pages c1–84, Jan 2008.

[51] Adoption of iso/iec 15288:2002 systems engineering-system life cycle processes. *IEEE Std 15288-2004 (Adoption of ISO/IEC Std 15288:2002)*, pages 1–67, 2005.

[52] Claude Y Laporte, Ronald Houde, and Joseph Marvin. Systems engineering international standards and support tools for very small enterprises. In *INCOSE International Symposium*, volume 24, pages 551–569, 2014.

[53] Systems and software engineering – lifecycle profiles for very small entities (vses) – part 5-6-1: Systems engineering – management and engineering guide: Generic profile group: Entry profile. *ISO/IEC TR 29110-5-6-1:2015*, pages 1–50, Jan 2015.

[54] Systems engineering standards. `http://www.incose.org/AboutSE/SEStandards`. [Online;Accessed: 01-February-2017].

[55] Swaminathan Natarajan, Padmalata V. Nistala, and Kesav V. Nori. Instantiating product-specific processes: A framework for product quality achievement. *INCOSE International Symposium*, 26(1):1731–1746, 2016.

[56] Rory V. O'Connor and Claude Y. Laporte. The evolution of the iso/iec 29110 set of standards and guides. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 10(1):1–21, 2017.

[57] Rory V. O'Connor and Claude Y. Laporte. An innovative approach to the development of an international software process lifecycle standard for very small entities. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 7(1):1–22, 2014.

[58] C. Y. Laporte and R. V. O'Connor. Systems and software engineering standards for very small entities: Accomplishments and overview. *Computer*, 49(8):84–87, Aug 2016.

[59] Corina White, Clifford A. Whitcomb, Rabia Khan, Dana Grambow, Jessica Delgado, and José G. Vélez. Development of a systems engineering career competency model for the u.s. department of defense. *INCOSE International Symposium*, 26(1):1864–1874, 2016.

[60] Joseph Kasser, Moti Frank, and Yang Yang Zhao. Assessing the competencies of systems engineers. Stockholm, Sweden, 2010.

[61] Ella-Mae Hubbard. 9.6.2 developing systems engineering competencies in undergraduate students for industrial placements. *INCOSE International Symposium*, 22(1):1379–1386, 2012.

[62] Chris Unger. Comprehensive approach to systems engineering capability development in ge healthcare. *INCOSE International Symposium*, 26(1):115–124, 2016.

[63] Mitre systems engineering competency model. In *The Mitre Institute*, Bedford, MA, September 2007. Mitre Corporation.

[64] DAU. Defense acquisition university (dau) eng competency model. `http://www.hci.mil/docs/compentency_assessment/ENG_Competency_Model_20130612_Final.pdf`. [Online;Accessed: 07-February-2017].

[65] NASA. Nasa project management and systems engineering competency model, academy of program / project engineering leadership (appel). `https://appel.nasa.gov/competency-model/`. [Online;Accessed: 07-February-2017].

[66] INCOSE. Incose systems engineering competency framework, international council on systems engineering (incose) united kingdom (uk). `http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_6.aspx?CatID=Publications`. [Online;Accessed: 07-February-2017].

[67] INCOSE. Annex a - guide to competency evaluation, international council on systems engineering (incose) united kingdom (uk). `http://www.incoseonline.org.uk/Program_Files/Publications/Publication_Search.aspx?CatID=Publications&query=SE%20Competencies%20Framework%20-%20Annex`. [Online;Accessed: 07-February-2017].

[68] Moti Frank. 7.2.1 towards a 4-d systems engineering cognitive competency model. *INCOSE International Symposium*, 24(1):617–634, 2014.

[69] INCOSE Website. Policy on se education. `http://www.incose.org/AboutSE/SEEducation/SEEducationPolicy`, 2016. [Online;Accessed: 07-February-2017].

[70] Gregory D. Burke, Michael J. Harrison, Robert E. Fenton, and Paul G. Carlock. 2.6.2 an approach to develop a systems engineering curriculum for human capital and process improvement. *INCOSE International Symposium*, 10(1):743–749, 2000.

[71] Christine Ng. Findings from a web-based survey of degree programs in engineering systems.

[72] Engineering education for a changing world. *American Society for Engineering Education*, 1994.

[73] A.P. Sage. Systems engineering education. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(2):164–174, May 2000.

[74] Systems engineering capability model interim standard, eai/is 731. 1999.

[75] Moti Frank. Knowledge, abilities, cognitive characteristics and behavioral competences of engineers with high capacity for engineering systems thinking (cest). *Systems Engineering*, 9(2):91–103, 1 2006.

[76] David D. Walden. Back to basics – up, out, and down: Three critical perspectives for the systems engineer. *INCOSE International Symposium*, 26(1):318–325, 2016.

[77] Peter L. Jackson. 2.2.2 getting design right: Systems engineering for the non-engineer. *INCOSE International Symposium*, 20(1):217–229, 2010.

[78] O.A. Asbjornsen and R.J. Hamann. Toward a unified systems engineering education. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(2):175–182, May 2000.

[79] Jan Herrington and Lisa Kervin. Authentic learning supported by technology: Ten suggestions and cases of integration in classrooms. *Educational Media International*, 44(3):219–236, 2007.

[80] Barbara B. Pfarr, Caroline Twomey Lamb, Maria M. So, and Donna H. Rhodes. 3.3.2 collaborative systems thinking: A response to the problems faced by systems engineering's "middle tier". *INCOSE International Symposium*, 19(1):477–489, 2009.

[81] Mark S. Avnet. A network-based analysis of team coordination and shared cognition in systems engineering. *Systems Engineering*, 19(5):395–408, 2016.

[82] Joseph Kasser, Phil John, and Yeoh Lean Weng. Systems engineering a 21st century introductory course on systems engineering: the seraswati project. In *Proceeding of APCOSE, Asia-Pacific Conference on Systems Engineering*, Hibiya, Japan, 2008.

[83] Dinesh Verma, Alberto Sols, Wolter J. Fabrycky, and Benjamin S. Blanchard. Computer-based courseware for systems engineering education and training. *INCOSE International Symposium*, 2(1):543–550, 1992.

[84] Michael J. Prince and Richard M. Felder. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, 95(2):123–138, 2006.

[85] Mark A. Turnquist, Raffaelo D'Andrea, Albert R. George, Peter Jackson, Linda K. Nozick, Donna Rhodes, Robin Roundy, Bart Selman, Christine A. Shoemaker, and Robert J. Thomas. 2.6.3 designing a systems engineering educational program using academic/industry collaboration. *INCOSE International Symposium*, 10(1):750–756, 2000.

[86] Moti Frank, David Elata, and Ilana Lavy. 5.6.3 developing capacity for engineering systems thinking in a project-based learning environment. *INCOSE International Symposium*, 13(1):957–964, 2003.

[87] Colin J. Neill and Joanna F. DeFranco. Facilitating collaborative learning in online engineering student teams. *INCOSE International Symposium*, 26(1):1219–1230, 2016.

[88] John Pourdehnad, Dennis Wilson, and Erica Wexler. Systems & Design Thinking: A Conceptual Framework for Their Integration. In *Proceedings of the 55th Annual Meeting of the ISSS-2011, Hull, UK*, volume 55, 2011.

[89] KINDA Khalaf, SHADI Balawi, GEORGE WESLEY Hitt, and AHMAD Radaideh. Engineering design education: Where, when and how. *Journal of American Education*, 2013.

[90] K. Craig and P. Voglewede. Multidisciplinary engineering systems graduate education: Master of engineering in mechatronics. In *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, 2010 IEEE*, pages 1–14, April 2010.

[91] Ronald R. Luman Samuel J. Seymour. Academic perspectives of systems engineering. *JOHNS HOPKINS APL TECHNICAL DIGEST*, 29(4), 2011.

[92] LTC Michael J. Kwinn, Lt Col Edward A. Pohl, COL Michael L. McGinnis, and William B. Carlton. 1.2.5 capstone design in education: Systems engineering and the west point way. *INCOSE International Symposium*, 12(1):211–217, 2002.

[93] Michael B. Harris. 1.6.1 development of systems engineers: A structured approach based upon international experience. *INCOSE International Symposium*, 10(1):703–709, 2000.

[94] Gunter Hettich. Graduate course for systems engineering at the fachhochschule münchen. *INCOSE International Symposium*, 7(1):763–768, 1997.

[95] A. Pyster, D.H. Olwell, T.L.J. Ferris, N. Hutchison, S. Enck, J. Anthony, D. Henry, and A. Squires. Graduate reference curriculum for systems engineering (grcse^{TM}) v1.1. *Trustees of the Stevens Institute of Technology*, 2015.

[96] Stuart Kramer and Edward Pohl. Graduate systems engineering education at the air force institute of technology. *INCOSE International Symposium*, 7(1):769–775, 1997.

[97] Jeffrey M. Newcamp. Integration of se design principles in a capstone design course: Airborne birdstrike countermeasure. *INCOSE International Symposium*, 26(1):600–613, 2016.

[98] Mark S. Kaufman. Internal development of systems engineers. *INCOSE International Symposium*, 7(1):750–754, 1997.

[99] Omar Hammami and Jean-René Ruault. 1.4.2 systems engineering education for confirmed engineers:experience feedback. *INCOSE International Symposium*, 22(1):139–156, 2012.

[100] Sarah A. Sheard and Matthew T. Swayhoover. 1.1.4 structuring a professional systems engineering development program. *INCOSE International Symposium*, 15(1):36–49, 2005.

[101] Heidi L. Davidz and Mark W. Maier. 1.4.1 an integrated approach to developing systems professionals. *INCOSE International Symposium*, 17(1):95–112, 2007.

[102] James Armstrong and Jon Wade. Learning systems engineering by teaching it. *INCOSE International Symposium*, 25(1):211–220, 2015.

[103] P. A. ". Jansma and M. E. Derro. If you want good systems engineers, sometimes you have to grow your own! In *2007 IEEE Aerospace Conference*, pages 1–15, March 2007.

[104] Jon P. Wade, George Kamberov, Douglas A. Bodner, and Alice F. Squires. The architecture of the systems engineering experience accelerator. *INCOSE International Symposium*, 22(1):1806–1820, 2012.

[105] Alice Squires, Jon Wade, Pete Dominick, and Don Gelosh. Building a competency taxonomy to guide experience acceleration of lead program systems engineers. Redondo Beach, CA, 2011.

[106] SupMeca Website. Placis: Collaborative platform for systems engineering. http://www.supmeca.fr/fr/international/placis-plateforme-collaborative-d-ingenierie-systemes, 2016. [Online;Accessed: 20-Mai-2016].

[107] E. Gill, J. Kreisel, and D. Verma. 10.1.2 integrating systems and business engineering in an international context: The spacetech postgraduate program. *INCOSE International Symposium*, 19(1):1457–1470, 2009.

[108] William T. Scherer and Douglas J. White. A systems engineering capstone design experience. *INCOSE International Symposium*, 2(1):527–531, 1992.

[109] Jean-Claude Tucoulou, David Gouyon, and Eric Bonjour. A student challenge in systems engineering: RobAFIS 2010. *INSIGHT - International Council on Systems Engineering (INCOSE)*, 14(4):9–11, December 2011.

[110] G. M. Bonnema, I. F. Lutters-Weustink, and F. J. A. M. van Houten. Introducing systems engineering to industrial design engineering students with hands-on experience. In *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on*, pages 408–413, Aug 2005.

[111] Gordon W. Grange and Matra BAe Dynamics. 3 systems engineering education - academic or vocational? *INCOSE International Symposium*, 9(1):316–320, 1999.

[112] W. Bauer, W. Biedermann, B. Helms, and M. Maurer. A student laboratory for systems engineering: Teaching systems engineering to students without previous se-knowledeg based on an industry-oriented example. In *Systems Conference (SysCon), 2012 IEEE International*, pages 1–6, March 2012.

[113] C. B. Nielsen and P. Adams. Active learning via lego mindstorms in systems engineering education. In *Systems Engineering (ISSE), 2015 IEEE International Symposium on*, pages 489–495, Sept 2015.

[114] M. Kudret Yurtseven. Teaching systems thinking to industrial engineering students.

[115] Timothy L.J. Ferris. 5.2.3 product and process – student learning in an undergraduate systems engineering course. *INCOSE International Symposium*, 14(1):969–982, 2004.

[116] Ralf Hartmann and Ulrich Knirsch. 6.3.5 an implementation of a process centered collaborative engineering environment within the satellite industry. *INCOSE International Symposium*, 12(1):648–655, 2002.

[117] Azad M. Madni. Adaptable platform-based engineering: Key enablers and outlook for the future. *Systems Engineering*, 15(1):95–107, 1 2012.

[118] Mario Simoni, Eva Andrijcic, Bill Kline, and Ashley Bernal. Helping undergraduate students of any engineering discipline develop a systems perspective. *INCOSE International Symposium*, 26(1):495–511, 2016.

[119] Abneesh Singla, Visvanathan Thanigai Nathan, and S Vageesh. Model based self-learning system engineering approach. *INCOSE International Symposium*, 26(s1):219–233, 2016.

[120] José L. Fernández and Guillermo Moreno. Mbse for engineering students. *INCOSE International Symposium*, 26(1):1231–1245, 2016.

[121] David Gouyon, Fabien Bouaron, and Gerard Morel. Specifying some key se training arti-facts. *Fourth International Conference on Complex Systems Design and Management, CSDM*, pages 207–218, Dec 2013.

[122] Adedeji B. Badiru and Rochelle R. Jones. A systems framework for distance learning in engineering graduate programs. *Systems Engineering*, 15(2):191–202, 1 2012.

[123] Srujal Patel, Dirk Schaefer, and Daniel Schrage. A pedagogical model to educate tomorrow?s engineers through a cloud-based design and manufacturing infrastructure ? motivation, infrastructure, pedagogy, and applications. In *ASEE 2013 Annual Conference and Exposition*, 2013.

[124] Zoe Szajnfarber Madeleine C Brannon and Thomas Andrew Mazzuchi. Introducing the fundamentals of systems engineering to freshman through various interactive group activities. In *2014 ASEE Annual Conference*, Indianapolis, Indiana, June 2014. ASEE Conferences. https://peer.asee.org/20705.

[125] Samuel Rochet, Claude Baron, Hugues Malgouyres, and Daniel Esteve. 1.1.3 propositionofa formal frameto facilitatethe applicationof processesin systems engineering. *INCOSE International Symposium*, 18(1):31–45, 2008.

[126] Mark S. Avnet. 1.3.1 information flow, team coordination, and shared knowledge in integrated concurrent engineering. *INCOSE International Symposium*, 24(1):41–57, 2014.

[127] Laurence Bellagamba. 6.3.2 testing knowledge of systems engineering process fundamentals. *INCOSE International Symposium*, 5(1):76–83, 1995.

[128] Video tutorial: A process-centred and project-based learning solution for teaching systems engineering. `https://www.youtube.com/watch?v=ANy67Jxy3to&feature=youtu.be`. [Online;Accessed: 01-February-2017].

Appendices

A

Appendix: An illustration of relevant features from the proposed solution

Organization Space

Organization features

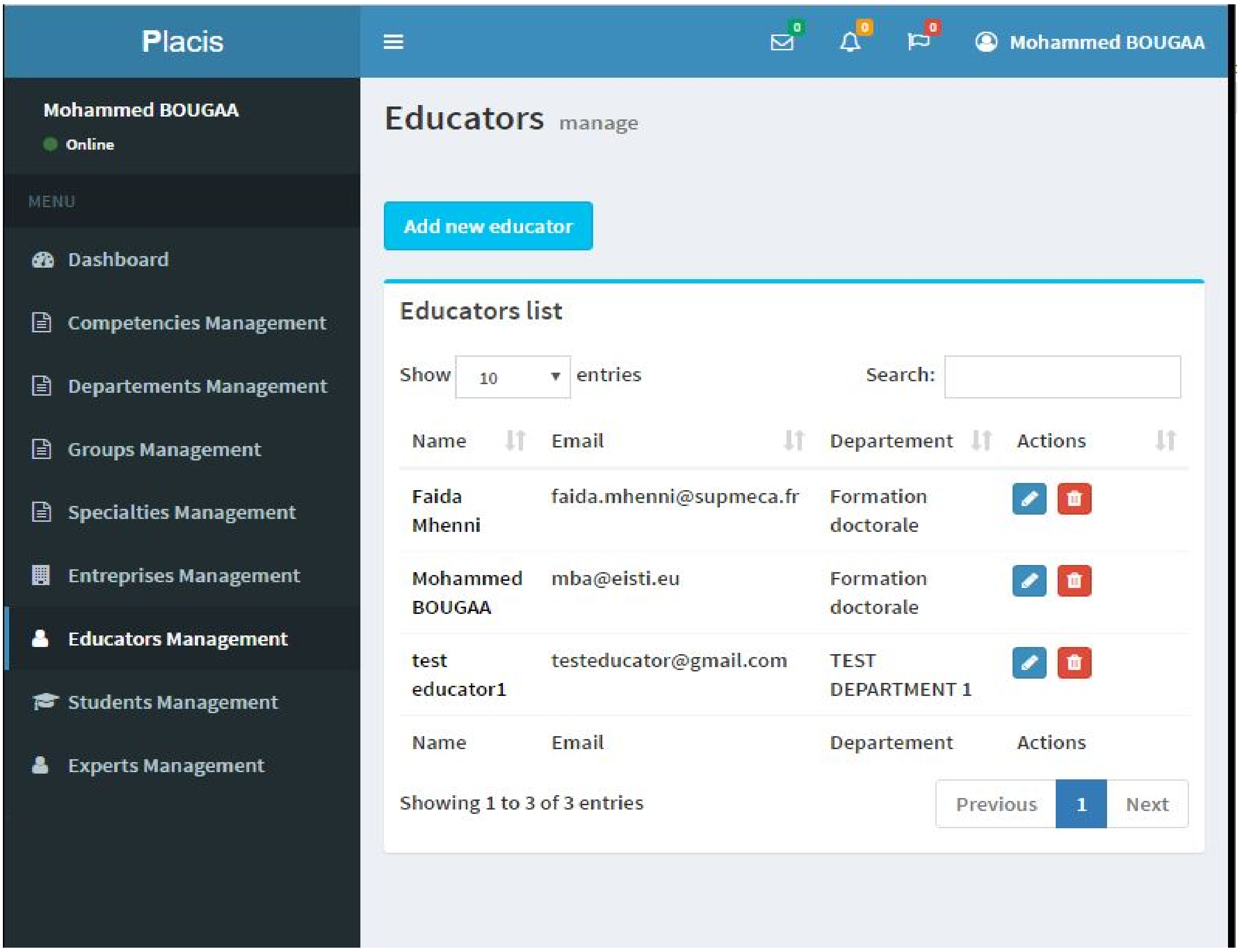

Figure A.1: Different features available to an organization

Competency model creation: adding competencies

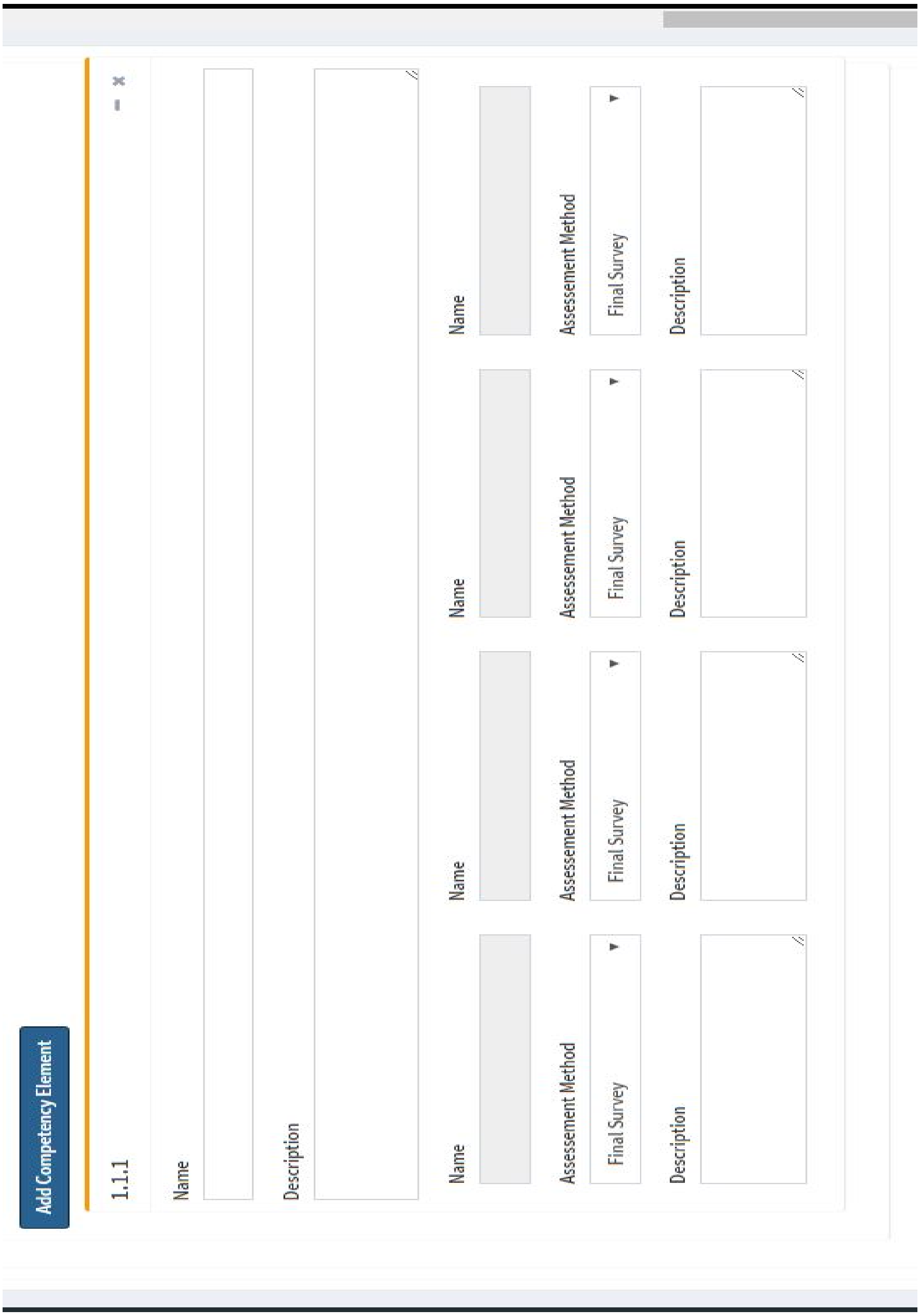

Figure A.2: Adding a competency element with its associated levels to a competency model

Competency model display:

Figure A.3: Displaying a competency model: Concepts and Architecture Area, Mission Needs Statement Competency and its elements

Educator space

Processes management interface

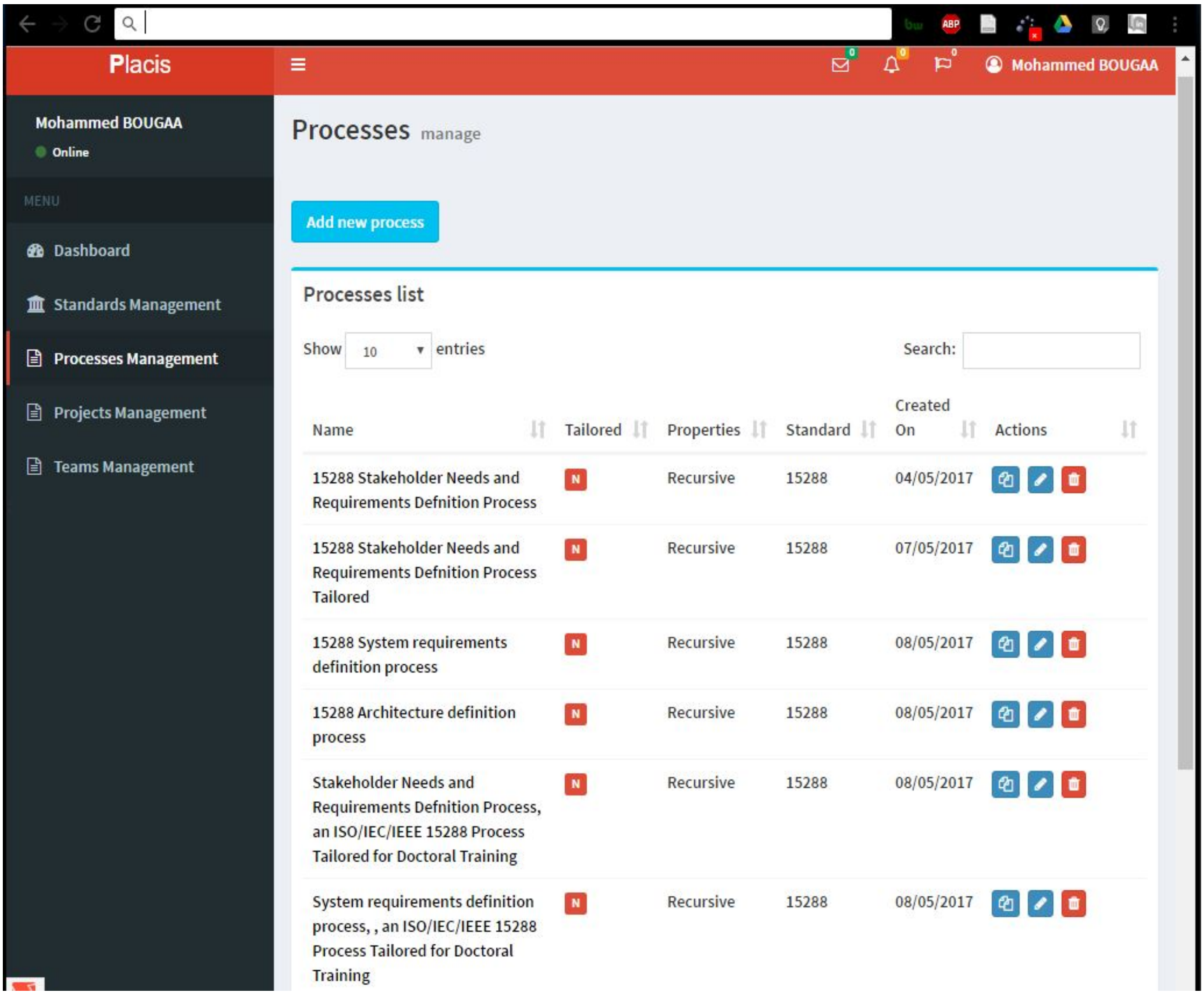

Figure A.4: Processes management system interface

Processes creation interface: Adding activities and competencies

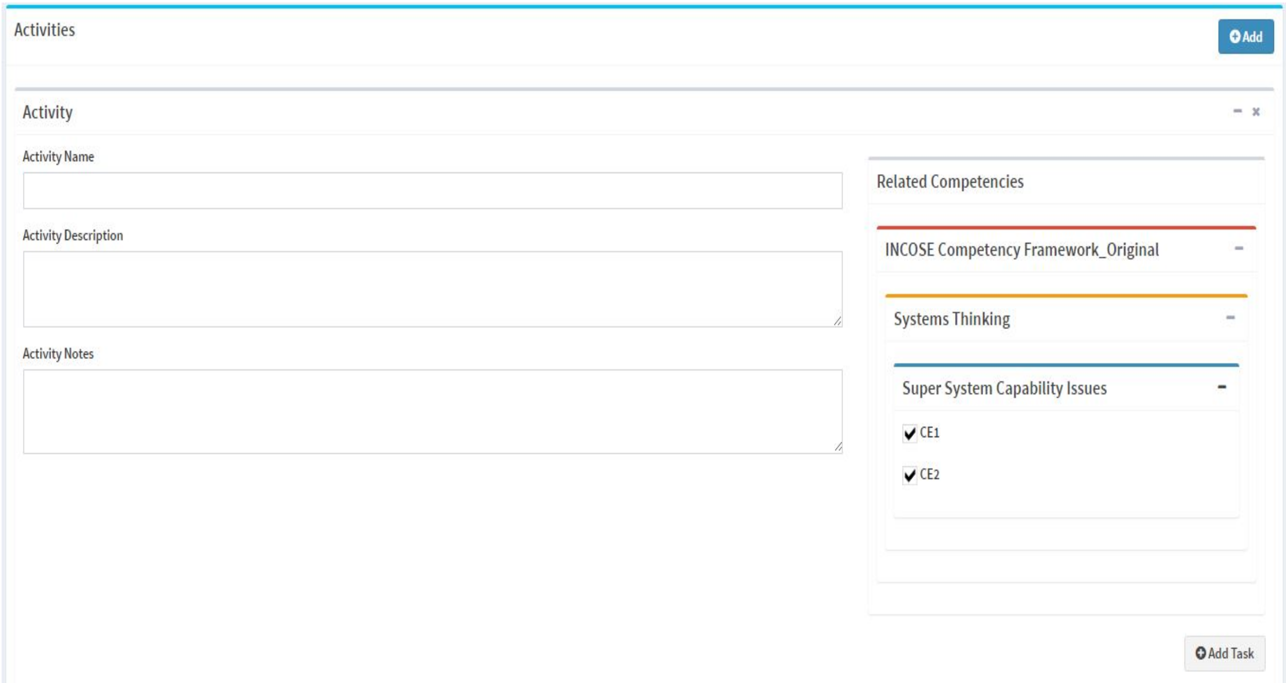

Figure A.5: Competencies attribution to process activities

New project definition interface

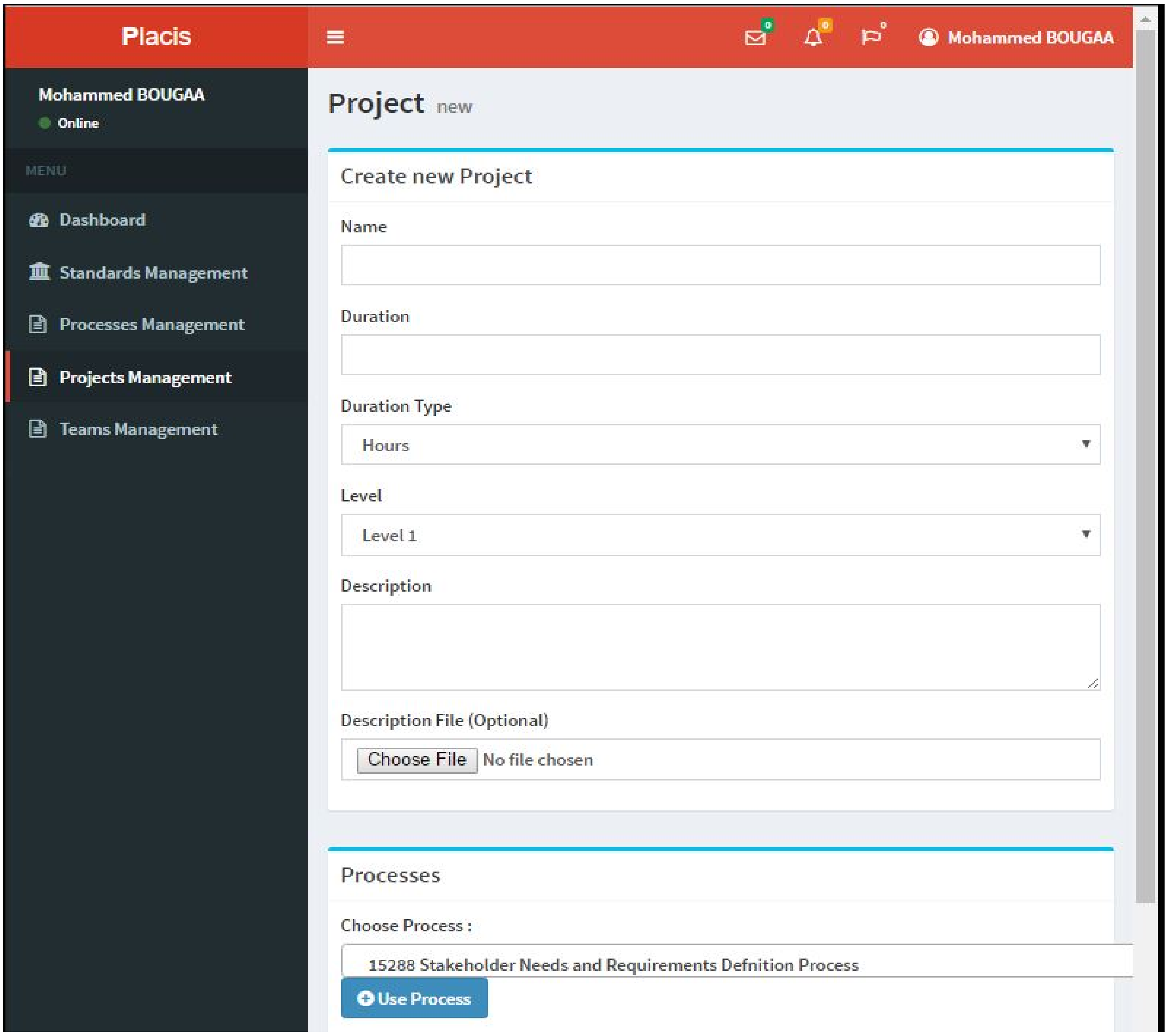

Figure A.6: New project creation

Processes tailoring when creating the project

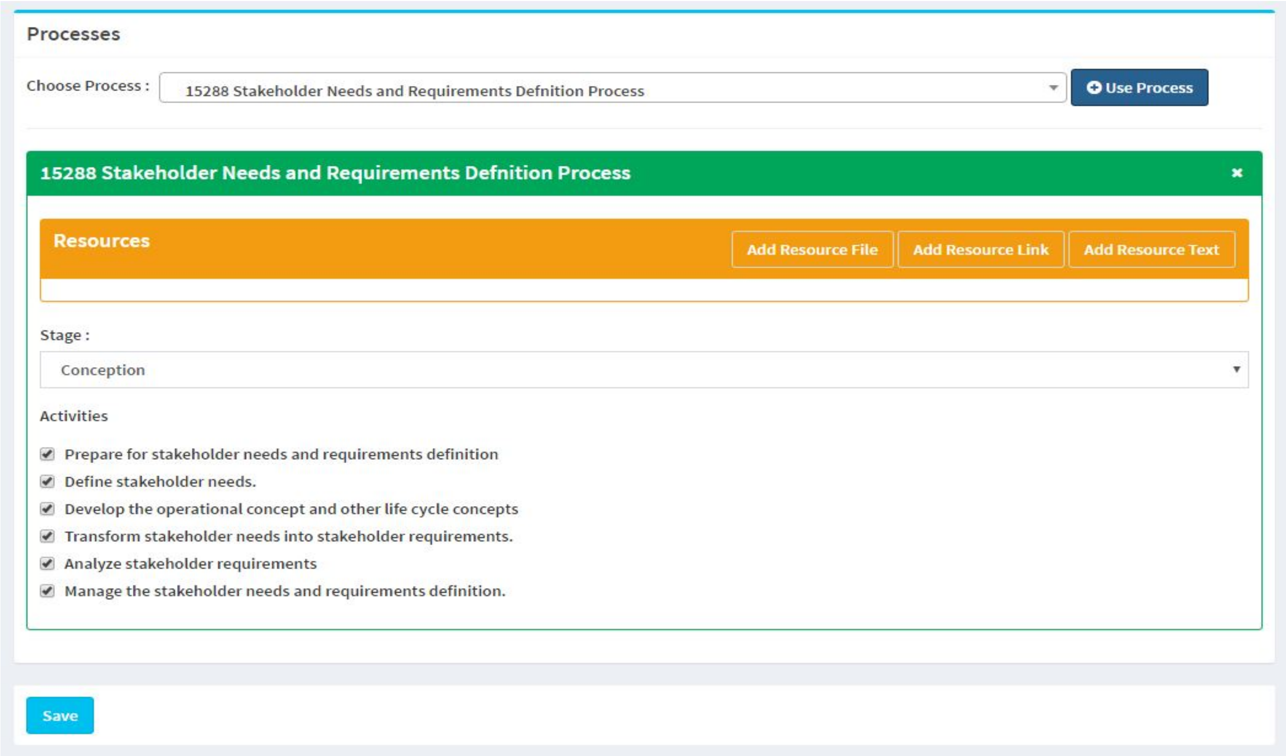

Figure A.7: processes tailoring, within project creation scenario, with its resources and stages definition

Selecting and attaching competencies to the project

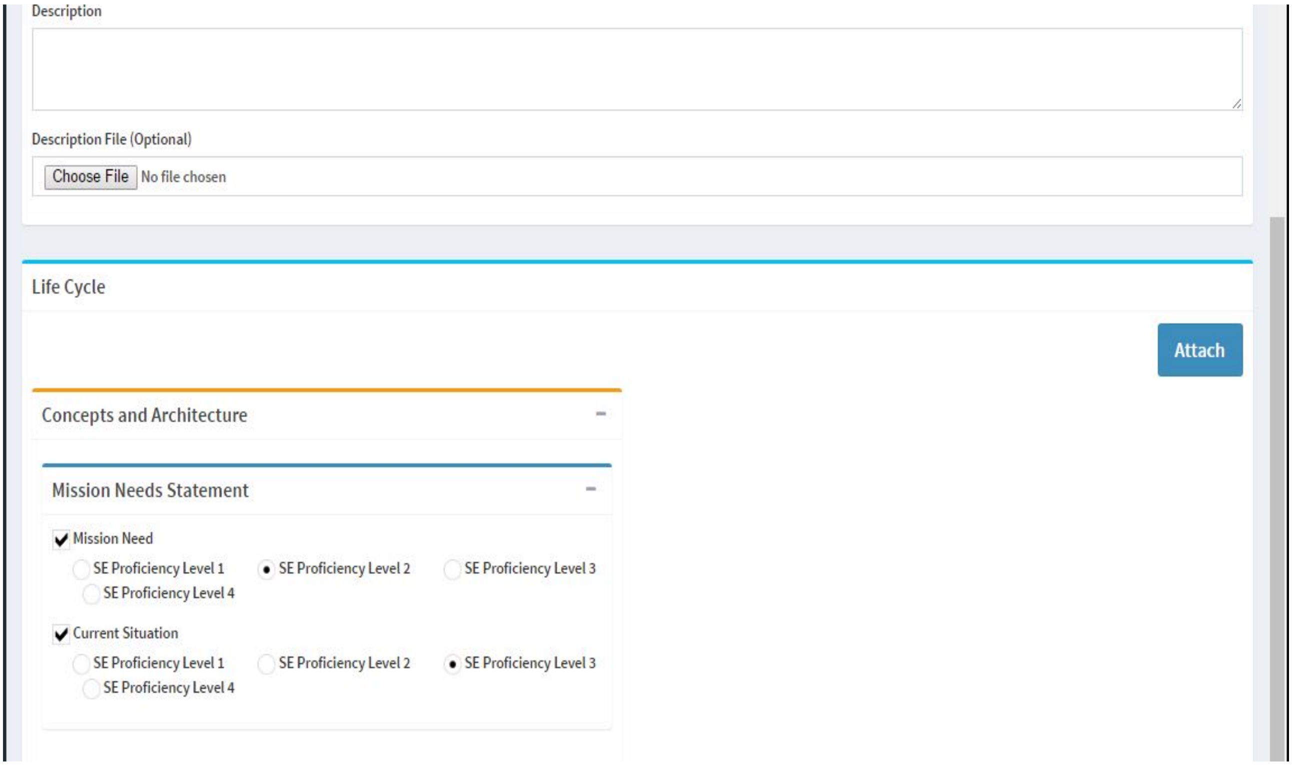

Figure A.8: Processes tailoring depending on competencies selection

Resulted life cycle model depending to attached competencies

Figure A.9: Processes tailoring depending on competencies selection

Projects-teams assignments

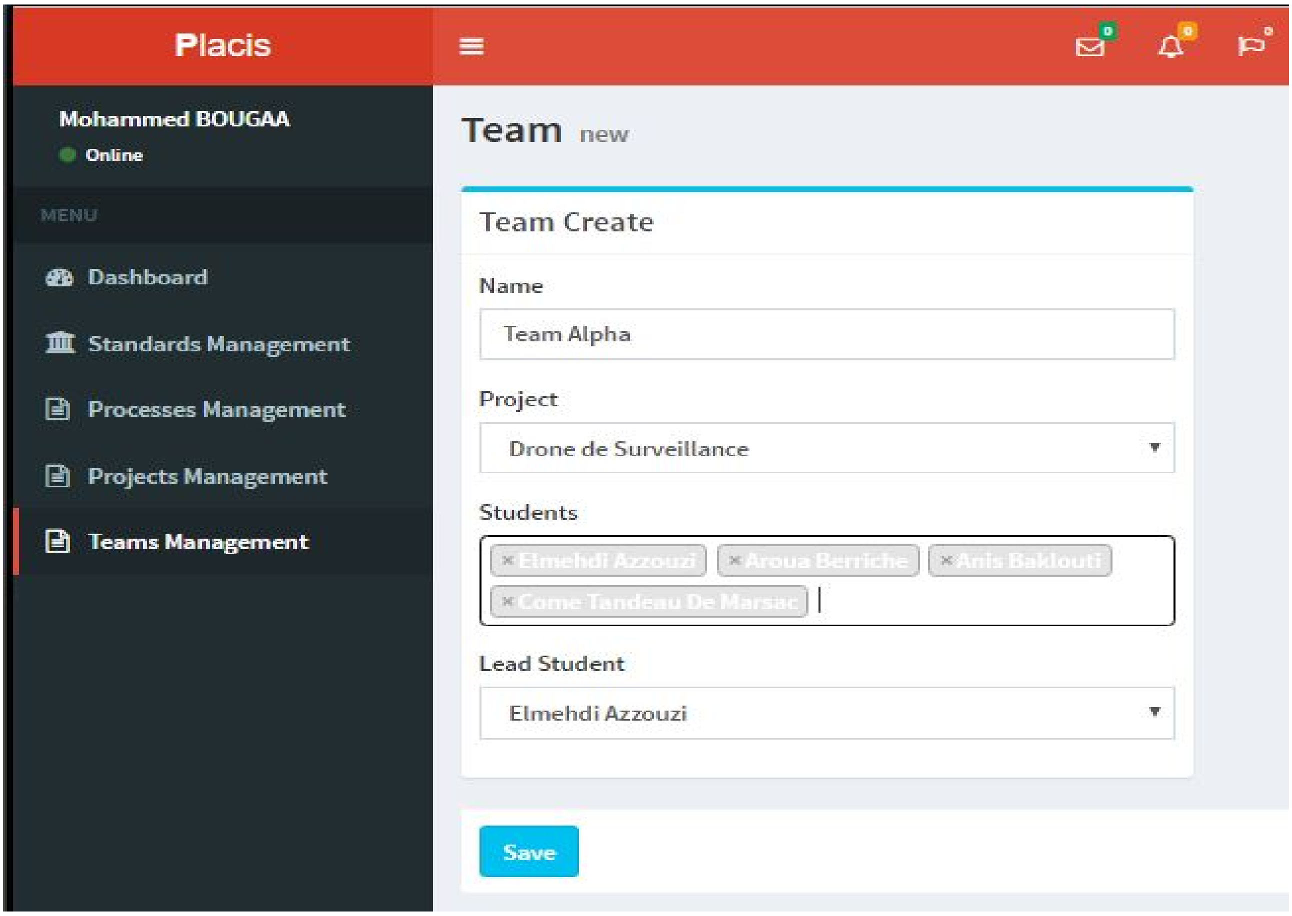

Figure A.10: Team-project assignment interface

Work progress checking (team workspace access)

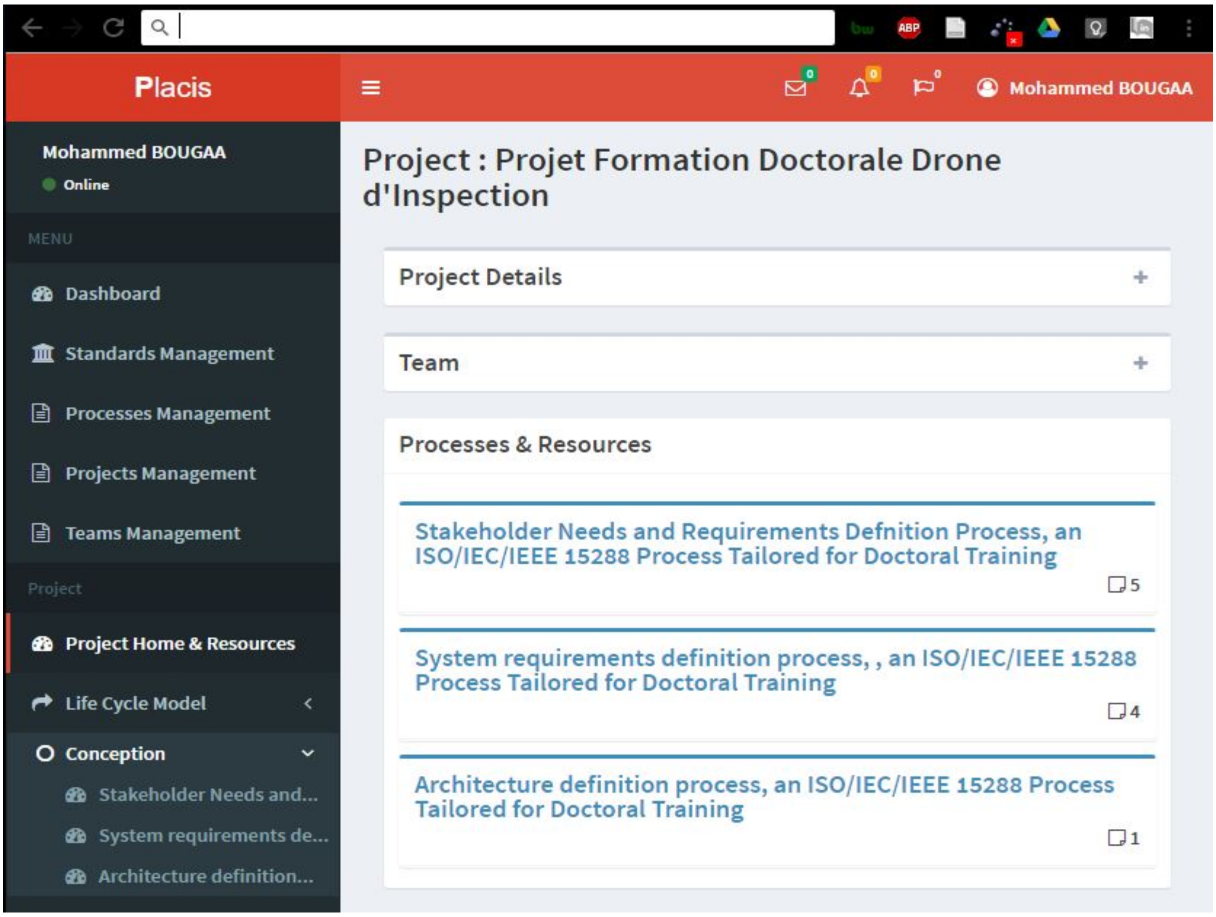

Figure A.11: Team workspace from an educator point of view

Students space

Team-project engineering workspace

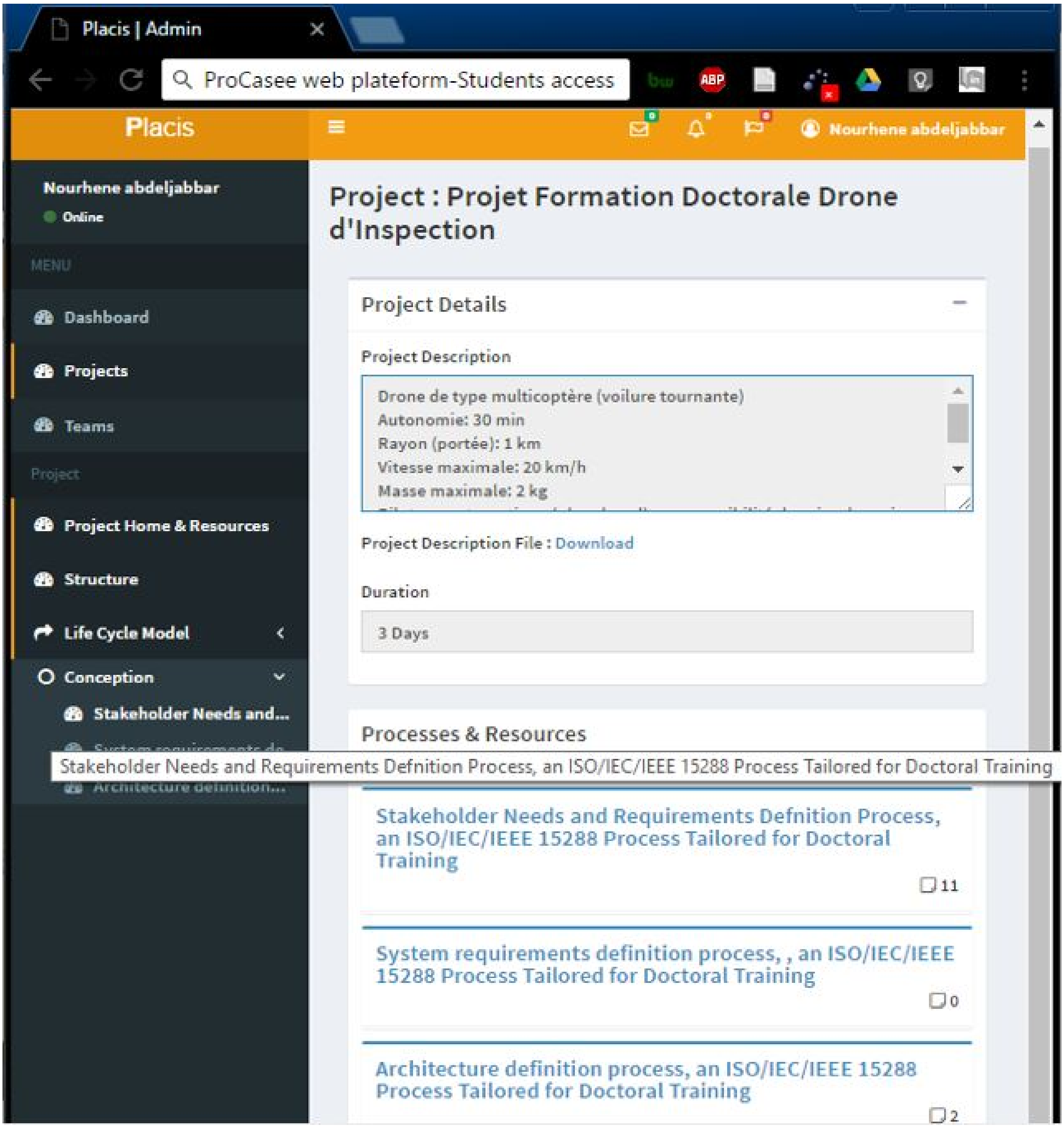

Figure A.12: Students collaborative engineering workspace

Process access within team-project workspace

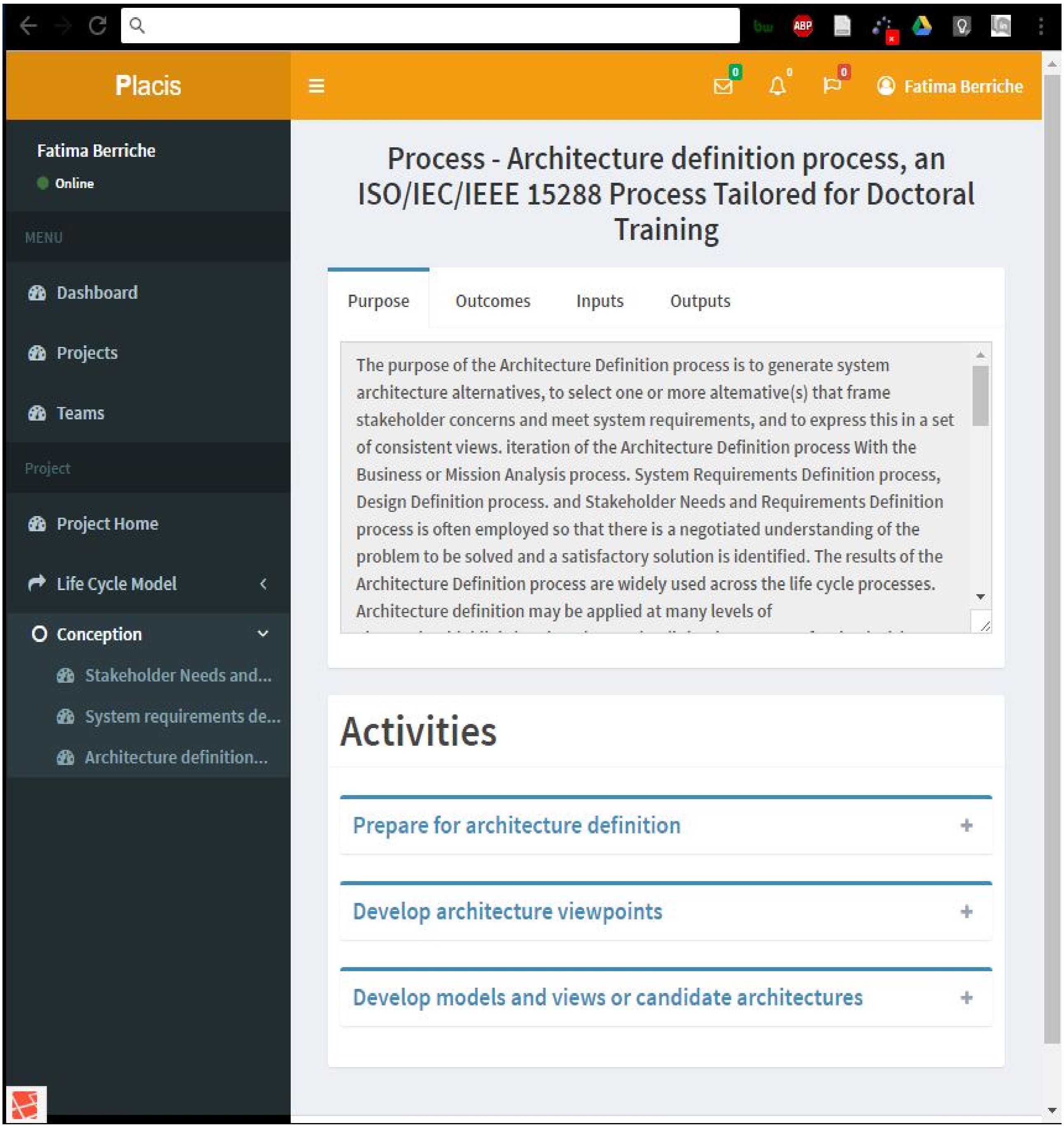

Figure A.13: Process interface within the team-project workspace

Tasks results uploading system

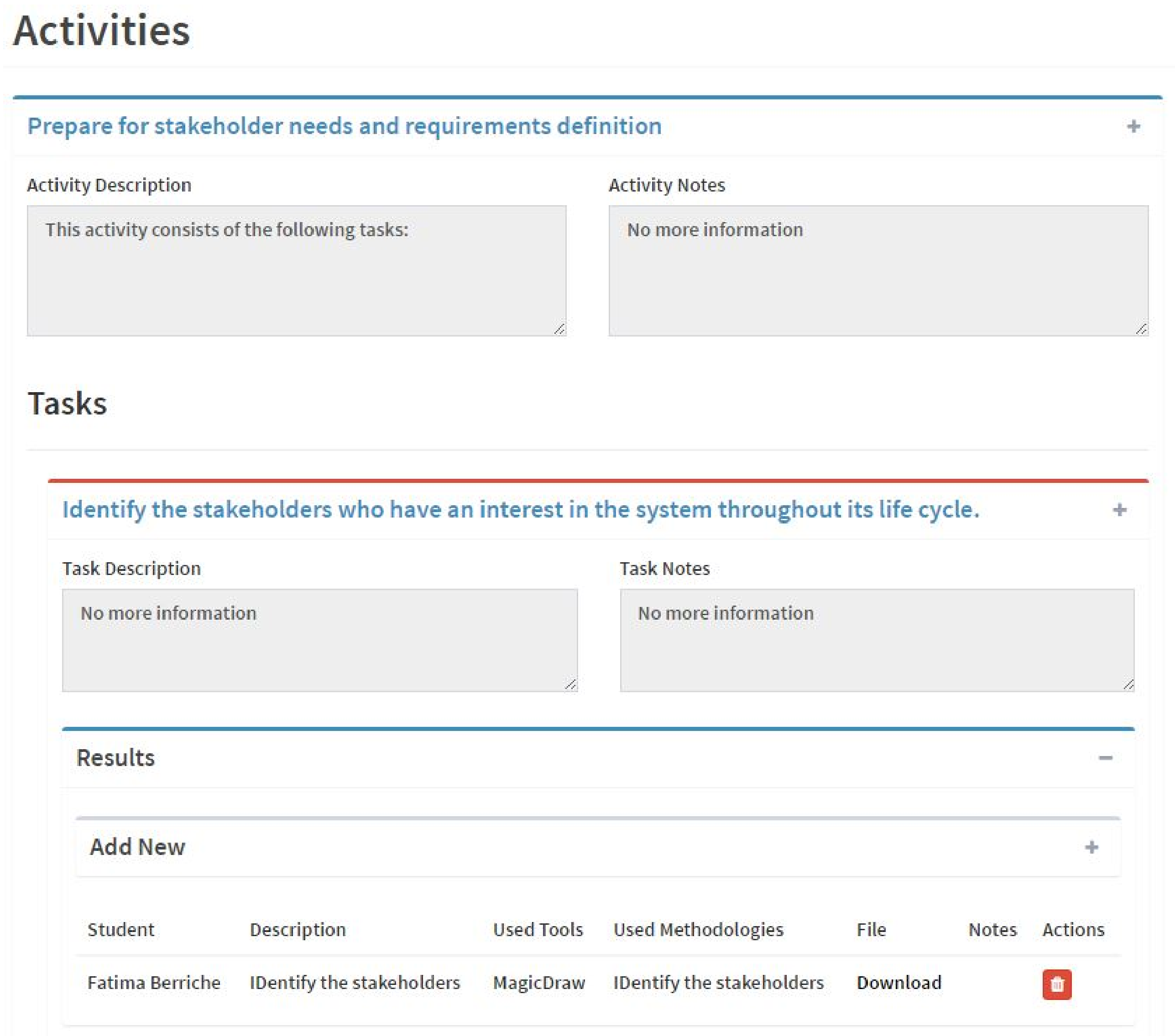

Figure A.14: Tasks results uploading system

System structural architecture definition

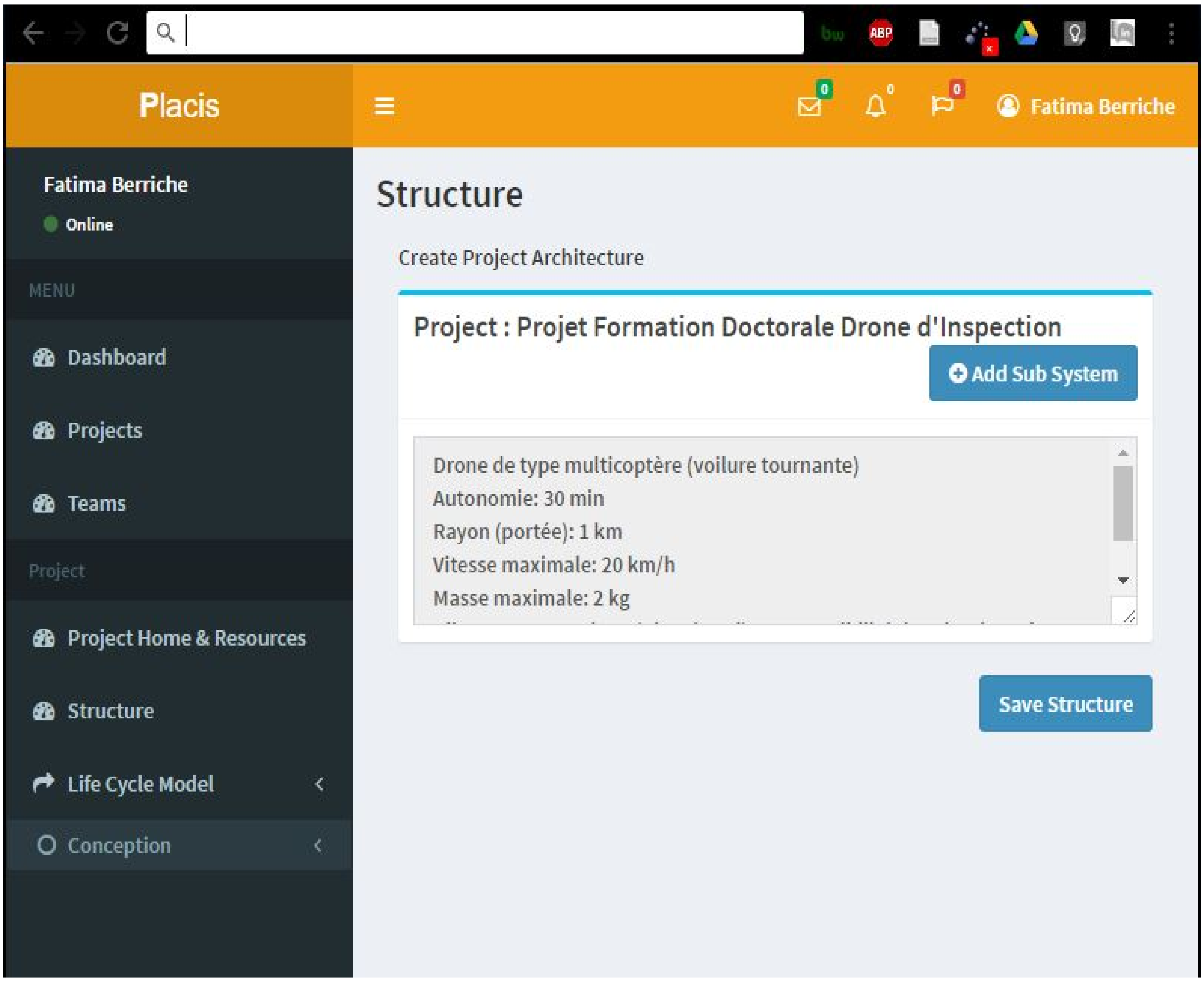

Figure A.15: Sub-systems and system elements creation

3D Design review and assembly feature, only supporting Lego Bricks for now.

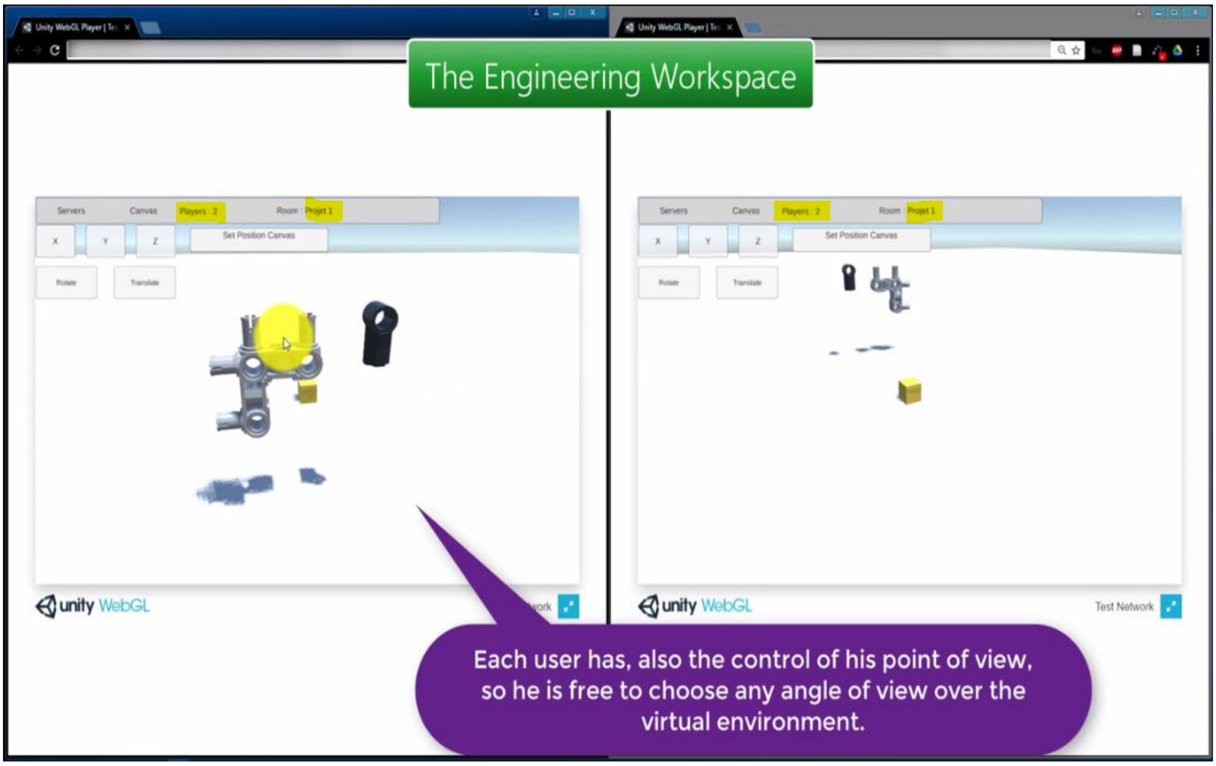

Figure A.16: a collaborative virtual 3D environment for virtual design review and assembly

B

Appendix: First experimentation plan "Evaluation Surveys"

Measuring the impact of a new learning approach for Systems Engineering.

The goal of this survey is to measure, from learners and educators' perspective, the benefits of a new approach to teach systems engineering discipline.
This new approach is implemented as a web-based platform. It's based on the use of "International Systems Engineering Standard Processes" for learning purposes, and made to be used in a geographically distant environment, by learners working as a team on a project-based learning mode.

(https://www.youtube.com/watch?v=mhD1E8oChqQ).

We thank you on advance for your response.

*Obligatoire

http://youtube.com/watch?v=mhD1E8oChqQ

Some information about you

These information will only be used for statistical goals, and will be treated anonymously.

1. **You are:** *

Une seule réponse possible.

- Industry Systems Engineering Practicioner
- Academy Systems Engineering Practicioner
- Both Industry and Academy Systems Engineering Practicioner
- Autre :

2. **How much experience you have in Systems Engineering field** *

Une seule réponse possible.

- Less than 1 year
- 2 to 5 years
- More than 5 years

3. **How you used to transfer Systems Engineering knowledge and skills** *

Plusieurs réponses possibles.

- [ ] I dont teach Systems Engineering
- [ ] Systems Engineering university lectures
- [ ] Systems Engineering Books
- [ ] Using MOOCs (Massive Open Online Courses, ex.Coursera, Edx...etc)
- [ ] E-learning plateforms (Other than MOOCs)
- [ ] Academic Project Based learning
- [ ] Inductry-Academy Project Based learning
- [ ] Competition and challenges organization
- [ ] Autre : ____________

4. **What are the topics/processes you deal with the most in Systems Engineering** *

Une seule réponse possible par ligne.

	All the time	Occasionally	Rarely	Never
Technical management (project planning & assessment, decision & risk management ...etc)	○	○	○	○
Acquisition and Suply	○	○	○	○
Project-Enabling Management (Life Cycle Model, Infrastructure, Portfolio, Knowledge...etc)	○	○	○	○
System & Stakeholder Requirements	○	○	○	○
Architecturing	○	○	○	○
Design, analysis and implementation	○	○	○	○
Integration	○	○	○	○
Test, verification and Validation	○	○	○	○
Operation and Maintenance	○	○	○	○

What do you think about the actual features of the proposed solution

5. **Please give your opinion about usefulness of these features** *

Une seule réponse possible par ligne.

	Obligatory	Very Useful	Useful	Moderately Useful	Not useful at all	Need Improvements
Processes Management System						
Adding related resources to processes						
Projects Creation and Management						
Life Cycle Model Definition						
Notification Management System						
Enabling learning by executing the real Systems Engineering Processes						
Reviewing the 3D Design of the System						
Supervising all tasks results in the same shared space						
Enabling Geodestributed students to work together						

6. **Please give your opinion regarding the ease-of-use (the ergonomy) of these features** *

Une seule réponse possible par ligne.

	Very simple to use	simple to use	Prety hard to use	Very hard to use
Processes Management System				
Adding related resources to processes				
Projects Creation and Management				
Life Cycle Model Definition				
Notification Management System				
Reviewing the 3D Design of the System				
Supervising all tasks results in the same shared space				
Enabling Geodestributed students to work together				

What do you think about the following additional features:

These are some additional features we think useful to improve this solution, please provide us with your opinion.

7. **Enabling students in the same team to communicate in real time, by adding a chat system** *

Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ....

8. **Enabling students in the same team to communicate in real time, by adding a video call system** *

Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ....

9. **Allowing educators to assist students by making notes about the tasks being realized durring their execution.** *

Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ....

10. **Allowing student to Manage and execute the processes of the Life Cycle Model on the full system, but also on its sub-systems and systems elements. in other terms, adding a feature that allow students to switch between the different sub-systems and systems elements.** *

Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

11. **What about adding a features that enables students to manage and assign the different tasks to the different students of the team.** *

Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ....

12. **Would it be better if it's for the Educator himself to manage and assign the different tasks to the different students of the team.** *
Une seule réponse possible.

- Yes
- No
- Autre :

13. **What about adding a feature that enables students getting direct web-based access to systems engineering tools (such as SysML)** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

14. **What about adding a feature that enables students getting direct web-based access to documentation and learning plateforms such as EDX.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

15. **What about giving the the Educator the ability to download at any time, a synthesis of the project as a Pdf File.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

16. **What do you think will be the best way to evaluate the work done under this solution?**
Plusieurs réponses possibles.

- [ ] Self Evaluation (By taking a survey)
- [ ] Peer Evaluation (Evaluation of the results by other students)
- [ ] Educator evaluation regarding the final Results
- [ ] Educator evaluation throughout the project execution
- [ ] Educator evaluation regarding the aquired skills and Knowledge (using questionnaires and projects execution data)
- [ ] A mixture of some of the previous methods (please check the ones you think are the best)
- [ ] Autre :

Comparing this solution

17. **In which way do you think this solution will be BETTER than the solutions you used to know or to Teach systems engineering with (especially after adding the additional features).**

Plusieurs réponses possibles.

- [ ] The ease of use of the solution
- [ ] The management of Geodistributed students
- [ ] The management of a Project Based Learning approach
- [ ] The use of Systems Engineering Standard Processes
- [ ] The ability to manage the Life Cycle Model
- [ ] The ability to use any Systems Engineering Standard (Standard choice independant)
- [ ] The ability to use any Systems Engineering Tools (Tools choice independant)
- [ ] The ablity to evaluate students regarding different Metrics
- [ ] Autre :

18. **In which way do you think this solution will be WORST than the solutions you used to learn systems engineering (especially after adding the additional features).**

Plusieurs réponses possibles.

- [ ] The ease of use of the solution
- [ ] The management of Geodistributed students
- [ ] The management of a Project Based Learning approach
- [ ] The use of Systems Engineering Standard Processes
- [ ] The ability to manage the Life Cycle Model
- [ ] The ability to use any Systems Engineering Standard (Standard choice independant)
- [ ] The ability to use any Systems Engineering Tools (Tools choice independant)
- [ ] The ablity to evaluate students regarding different Metrics
- [ ] Autre : ______________________

C

Appendix: Second experimentation plan ”Doctoral training in systems engineering” Students Experimentation Survey

Doctoral training in Systems Engineering

En votre qualité de participant à cette première formation doctorale en Ingénierie Système nous aimerions avoir votre avis sur:
- Les connaissances acquises à terme des trois journées de formation en Ingénierie Système
- L'utilité et la simplicité d'utilisation de la plateforme web utilisée, et les avantages de l'utilisation des processus standards d'ingénierie système dans le cadre de l'apprentissage.

Nous vous remercions par avance pour vos réponses

*Obligatoire

1. **Adresse e-mail** *

Platform Evaluation

Please let us know what do you think about the features of the provided platform

2. **Did you find these features useful ?** *

Une seule réponse possible par ligne.

	Very useful	Useful	Moderately useful	Useless
Enabling Geo-destributed students to work together				
Learning by executing the Standard Systems Engineering Processes				
Reporting all tasks results in the same shared space				
Processes related resources				
System structure creation and manipulation				
Accessing and reviewing other team members uploaded results				

3. **Did you find these features easy to use ?** *

Une seule réponse possible par ligne.

	Very simple to use	Simple to use	Moderately simple to use	Hard to use
Accessing and reviewing other team members uploaded results				
Reporting all tasks results in the same shared space				
Processes presentation				
Navigating through processes description				
System structure creation and manipulation				

4. **Will the following additional features be useful ?** *

Une seule réponse possible par ligne.

	It will be very useful	It will be useful	It will be moderately useful	It will be useless
Allowing educators to assist students by adding notes about the tasks being realized, durring their execution.				
Adding a feature that enables students to manage and assign the different tasks to the different students of the team.				
Adding a feature that enables students getting direct web-based access to systems engineering tools (such as SysML Tools)				
Giving the ability to students to download at any time, a synthesis of the project as a Pdf File.				

5. **What do you think are the most important features of this solution ?** *
Plusieurs réponses possibles.

- [ ] The ease of use of the solution
- [ ] The management of geo-distributed students
- [ ] The implementation of the Project-Based Learning approach
- [ ] The use of systems engineering standard processes
- [ ] The ablity to evaluate students regarding different metrics
- [ ] The abitlity to upload results and share them with others at the same shared space
- [ ] The ability to manage the system structure (different sub-systems and system-elements)

6. **In addition to actual pedagogical resources, offered within the different life cycle processes, what do you want to see on the learning scenario?** *
Plusieurs réponses possibles.

- [ ] More explanations about the different processes and activities
- [ ] More guidance about which tools and methodologies to use for each task
- [ ] "A sample use case" A real case systems engineering results uploaded over the entire life cycle tasks
- [ ] Information about the competencies to be learnt after executing each activity or process
- [ ] An overview about the entire set of standard processes
- [ ] Adding metrics that show the project progress on the project home page

7. **What do you think will be the best way to evaluate students work, performed inside this solution ?** *
Plusieurs réponses possibles.

- [ ] Self Evaluation (By taking a survey)
- [ ] Peer evaluation (evaluation of the results by other students)
- [ ] Educator evaluation regarding the final Results
- [ ] Educator evaluation throughout the project execution (Evaluation of processes execution quality)
- [ ] Educator evaluation regarding the acquired skills and Knowledge (using questionnaires and projects execution quality)
- [ ] A mixture of some of the previous methods (please check the ones you think are the best)
- [ ] Autre : ____________________

8. **Do you think that this platform should be used for education in other domains than systems engineering ?** *
Une seule réponse possible.

- ( ) Yes
- ( ) No

D
Appendix: Published Papers

First Paper: (SEFI) A Survey of Current Practices for Systems Engineering Education

- **Paper title:** A Survey of Current Practices for Systems Engineering Education
- **Authors:** Mohammed Bougaa, Stefan Bornhofen, Alain Rivière.
- **Conference:** SEFI 2016 Annual conference "Engineering Education on Top of the World: Industry University Cooperation"
- **Location:** Tampere, Finland
- **Date:** 12-15 September 2016
- **Keywords:** Systems Engineering (SE), Systems Engineering Education, Project-Based Learning (PBL).

A Survey of Current Practice in Systems Engineering Education

M Bougaa*
Quartz Laboratory EA7393, EISTI
Cergy-Pontoise, France
E-mail: mba@eisti.eu

S Bornhofen
Quartz Laboratory EA7393, EISTI
Cergy-Pontoise, France
E-mail: sb@eisti.eu

A Rivière
Quartz Laboratory EA7393, SUPMECA
Saint-Ouen, France
E-mail: alain.riviere@supmeca.fr

Keywords Systems Engineering (SE), Systems Engineering Education, Project-Based Learning (PBL).

Introduction

As defined by the International Council on Systems Engineering (INCOSE): "*Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs*". [1]. In simple terms, Systems Engineering is a structured approach focusing on the design and the management of complex engineering projects over their entire life cycle.
During the last decade, the complexity of systems has increased to an unprecedented level, due to multi-disciplinary stakeholders with various resources and considerations

*Corresponding Author, mba@eisti.eu

to be taken into account from the first system concept until the retirement stage. Much effort has been made to address these challenges, especially by standard organizations which published a number of handbooks and guidelines for Systems Engineering in order to develop a harmonized view of all involved processes that are required throughout a system's life cycle to transform customer needs into a system solution [2]. Several standards have been defined, from the early US Military Standard MIL-STD-499 dating back to 1969, to the last revision of the ISO/IEC/IEEE 15288 published on May 15th, 2015 [3]. Thanks to the creation of such standards, the engineering industry may better cope with the complexity problems that occur within the stages of a system life cycle. However, this requires well-trained human resources who master the fundamental and domain-specific Systems Engineering principles and their corresponding standard processes.

In order to be effective as Systems Engineer, engineering students need practical and real world experience in addition to the necessary knowledge in their traditional engineering discipline [4]. The difficulty of teaching the Systems Engineering approach for academic institutions is amplified by the fact that today's complex systems often involve the use of emerging technologies such as physical environments, visualization, virtualization, or Internet Of Thing (IoT), which are also considered as the heart of any efficient Factory of the Future [5]. In this context, the Placis[1] project has been initiated by three french engineering schools, together with industrial partners, in order to create a collaborative educational platform for Systems Engineering.

The work in this paper has been conducted in the scope of the Placis project. On the one hand, it surveys the pedagogical goals and specificities in Systems Engineering Education. We investigate the requirements for a powerful solution, present the most relevant up-to-date practices and discuss their advantages and their limits. In particular, we study the Project-Based Learning approach which is the preferred pedagogical model by most educators and researchers in this field. On the other hand, we outline some important open questions and suggest possible starting points for further research. The paper concludes by summarizing its most significant results, and provides perspectives for our future work.

1 Objectives of Systems Engineering Education

According to Muller [6], Systems Engineering Education differs from traditional mono-disciplinary engineering courses, since the training needs to focus more on skills and less on transferable facts. The author gives a set of recommendations to consider for a good Systems Engineering Education program, including interaction with students, soft skill development, media use and student feedback. Dym [7] believes that "*a good Engineering Education is about process, about learning how to think like an engineer; its much more than a prescription of content*".

Asbjornsen and Hamann [8] provide an overview of Systems theory and Systems Engineering methodology in order to design a pedagogical concept for both Engineering Education in general and Systems Engineering Education in particular. They argue that the initiative to take up Systems Engineering at a university level has come from industry and not from academia. By examining the industrial motivations, the authors identify a list of learning targets for high-quality Systems Engineering Education, such

[1] Placis Project: Collaborative Platform for Systems Engineering, driven by SupMECA. And funded by the French Ministry of Higher Education and Research under the future investments program with the reference ANR-11-IDFI-0029

as:

- *Broad-Based Qualitative Knowledge*
- *Deep Quantitative Knowledge*
- *Systems Engineering Ability and Insight*
- *Learning Ability*
- *Human Factors*
- *Loyalty and Individual Responsibility*
- *Global and Environmental Concerns*

For Sage [9], a major goal of Systems Engineering Education should be to acquire the abilities relative to each of the 19 focus areas for Systems Engineering identified by the Systems Engineering Capability Model (SECM) [10]. The author presents Systems Engineering Knowledge as a composition of three aspects:

- **Knowledge Perspective** which allows forecasting the need for innovation, including innovation principles to identify the appropriate systems planning and marketing directions.
- **Knowledge Principles** as formal problem solving approaches, which are usually linked to fundamental knowledge needed for research and development.
- **Knowledge Practices** representing the accumulated experience that has led to standard operating policies for well-structured problem solving.

The American Society for Engineering Education suggests consideration of: [11]:

- *Team skills, and collaborative, active learning.*
- *Communication skills.*
- *A systems perspective.*
- *An understanding and appreciation of diversity.*
- *Appreciation of different cultures and business practices, and understanding that engineering practice is now global.*
- *Integration of knowledge throughout the curriculum a multidisciplinary perspective.*
- *Commitment to quality, timeliness, continuous improvement.*
- *Undergraduate research and engineering work experience.*
- *Understanding of social, economic, and environmental impact of engineering decisions.*
- *Ethics.*

Dym et al. [12] recommend the following three activities for a powerful learning environment for Systems Engineering and similar disciplines:

- *Instrumenting the learning process to obtain quantitative and qualitative data that support metrics consistent with quality control.*
- *Teaching Design Engineering and other disciplines such as Systems Engineering across geographically dispersed, culturally diverse, international networks*
- *Engage design coaches to help manage the contextualization of engineering design theory and practice.*

Finally, in a broader context, Herrington and Kervin [13] specify nine main characteristics that any learning environment, technology-based or not, should feature:

- *Provide authentic context that reflect the way the knowledge will be used in real life*
- *Provide authentic activities*
- *Provide access to expert performances and the modeling of processes*
- *Provide multiple roles and perspectives*
- *Support collaborative construction of knowledge*
- *Promote reflection to enable abstractions to be formed*
- *Promote articulation to enable tacit knowledge to be made explicit*
- *Provide coaching by the teacher at critical times, and scaffolding and fading of teacher support*
- *Provide for authentic, integrated assessment of learning within the tasks*

At this stage, the need for an appropriate pedagogical model for Systems Engineering Education becomes apparent, a model that meets most of the previously cited goals and considerations while ensuring efficient learning outcomes and correct student assessment. This topic will be treated in the third Section. Prior to that, the next section presents some significant use cases.

2 Advances in Systems Engineering Education

This section presents some important currently proposed solutions for Systems Engineering Education around the world. Interestingly, no common pedagogical model can be identified. Each institution has its own teaching format varying from one-weak crash courses to multiple-year programs. Systems Engineering Education is often regarded as an extension to regular Engineering Education, typically taught to graduate students along with interdisciplinary studies, and sometimes included in undergraduate university programs [4]. INCOSE has formulated a policy statement that emphasizes the importance of Systems Engineering and its impact on regular engineering disciplines: "*INCOSE believes strongly that a systems perspective and the fundamental principles of systems engineering have an important role in the education of all engineers regardless of their specialty. This will strengthen the general recognition that most of today's engineering tasks are performed in multidisciplinary teams, and degree granting programs in systems engineering must be encouraged and supported*" [14].

Engineering schools do not always offer independent Systems Engineering courses or programs. Systems Engineering may be taught in the scope of a single module, under the label of product development, product design, design engineering or design thinking. Dym et al. [12] state that *"Design Thinking reflects the complex processes of inquiry and learning that designers perform in a systems context, making decisions as they proceed, often working collaboratively on teams in a social process, and "speaking" several languages with each other (and to themselves)"*. Other schools have created dedicated programs and curriculums for Systems Design, Systems Engineering and closely related areas [15]. We can therefore distinguish two approaches of Systems Engineering Education: Systems-centric and Domain-centric. *"Systems-centric programs treat systems engineering as a separate discipline and most of the courses are taught focusing on systems engineering principles and practice. While, Domain-centric programs offer systems engineering as an option that can be exercised with another major field in engineering*" [6].

2.1 Master Programs with Academia-Industry Partnerships

To meet the needs of modern engineers, and to address the innovation crisis in the U.S, Craig and Voglewede from Marquette University, Milwaukee, presented in 2010 a Master program of Engineering in Mechatronics [16]. Their approach focuses on finding a balance between academic rigor and best practice. The program includes 12 one-credit key modules covering fundamental engineering knowledge such as mathematics, physics, mechanics and electronics, along with Systems Engineering related knowledge such as control, analysis tools, systems modeling and design. The lessons learned in these modules are applied to four three-credit case study courses: *transportation system, home and office system, energy system, and automation system*. During these courses students get comfortable with the most important practices in Systems Engineering, including user and problem understanding, design, implementation, integration, trade-offs and optimization. Finally, a six-credit on-site experience allows the students to put it all together in a genuine industrial context.

Another significant example is the Master program created by the Johns Hopkins University (JHU) Whiting School of Engineering. In 2011, it was considered as the largest Systems Engineering program of the United States in terms of enrollment [17]. The program balances theory and practice, and offers different learning methods combining in-person classes, online classes and industry partnerships. It also includes a challenging capstone Systems Engineering Project. According to the authors, many students who followed the JHU program, pursued doctoral studies in Systems Engineering at other universities such as George Washington University, George Mason University, Stevens Institute of Technology, University of Virginia, and Old Dominion University.

2.2 Few-months international Academia-Industry Projects

Paris Higher Institute of Mechanics (SupMECA), leader of the Placis Project, has created a Systems Engineering Education program under the same name [18]. It aims to train engineers in a new format, asking students from different engineering schools, different countries and different disciplines, to work collaboratively on an international and multidisciplinary project. The students use the most recent engineering tools and technologies, including Catia V6, SysML, Abaqus, WebEx and Sharepoint.

Depending on the configuration of the program, it teaches to both Master and Bachelor students. In the case of Masters students, the projects are generally carried out during one semester with the following course of events:

- A multidisciplinary project is proposed or re-conducted by a company.
- The project is approved by the Industrial and Academic partners.
- Multidisciplinary student teams from different universities are formed (6-10 students per group).
- A kick-off meeting with all involved persons is organized (in-person or video conference).
- The students work on the project by distant collaboration. They are followed by teachers and industrial tutors.
- Teachers and tutors assess the students' results and performance (reports, models, behavior, final presentation, etc.)

2.3 Student challenges

AFIS, the French chapter of INCOSE, organize since 2006 a yearly student competition for robot design, called RobAFIS [19]. Around 10 student teams from french universities and engineering schools, inexperienced in Systems Engineering, participate each year in this competition.

Each team can consult a Systems Engineering teacher, and they can also question other AFIS experts. The roadmap starts about 8 months before the final stage competition, when AFIS communicates the general schedule, the regulations, specifications, and a reference development document. Three months before the final stage, the teams register and receive a LEGO Mindstorms Robotics kit in order to physically implement their solution. Fifteen days before the final stage, the teams send their development document to Systems Engineering experts for evaluation. The competition concludes by a final stage where all teams meet and operationally validate their works, along with project and configuration audits. Few weeks after the competition, the students receive a detailed debrief regarding their work.

2.4 Few-weeks Projects within regular Engineering Curriculums

In 2004, Bonnema et al. presented a solution to introduce Systems Engineering to third-year students in Industrial Design Engineering at the University of Twente [20]. The SAS project (Sensors, Actuators and Systems) applies Systems Engineering tools and techniques in a concrete situation. More specifically, the project allowed students to learn the basics and goals of Systems Engineering, and to keep an overview in a complex design project. The students worked in large groups of 12-14 persons in a project-based learning approach, without a tutor except for the possibility to discuss with some specific staff members. They were provided with lectures on a selected set of subjects on Systems Engineering, which represents 34 credit hours of the entire 140 credit hours of the SAS project, together with 53 hours for Sensors and Actuators, and 53 hours for the Assignment. The main study material was the 2.0 edition of the INCOSE handbook, the "Systems Engineering and Analysis" book [21] and the Introduction to Systems Engineering book [22].

There was no planning given to the students, except for two milestones: *Customer requirements, systems requirements, system concept, and sub systems*, that needed to be done during the first three weeks of the project, and *Sub-system design, plan for system integration and test, final system design*, scheduled for the following fours weeks. In 2004, the students were asked to design an intelligent climate-control system for houses, and in 2005 an intelligent car. They were evaluated based on poster sessions, a Sensors and Actuators exam, and a short essay on the application of Systems Engineering methods in their project.

2.5 Theoretical Courses within Industrial Engineering Curriculums

Already in 2000, Yurtseven from Dogus University in Turkey, presented two courses that dealt with Systems Engineering and Design for students in Industrial Engineering [23]. These courses were mainly theoretical. The first one addressed senior level students and provided a background on the fundamentals of Systems Engineering. After introducing the main concepts of design and engineering, the course included several topics such as: Design Options, Engineering Systems Modeling, Analysis of System Reliability, System Dynamics and State Transition Matrix Models, Modeling the Research and Development Process, Systems Life-Cycle and Optimization, and the Management of Engineering Systems Design and Operations.

The second course addressed graduate level students and introduced some unconventional methodologies in Systems Engineering. It included the following topics: Introduction to the Management of Advanced manufacturing Technology, Introduction to Sociotechnical Systems Theory, Cognitive Systems Engineering , and Soft Systems Methodology.

2.6 Few-Days Laboratories

The *Technische Universität München*, *Germany*, proposes since 2012 a laboratory on Systems Engineering in the context of product development[24]. This laboratory is held by three researchers in form of a five-day event, targeting Master and Bachelor students in Mechanical Engineering without previous knowledge in Systems Engineering. An industrial case study is used as a teaching framework, along with a subset of aspects and processes from a typical Systems Life Cycle in Systems Engineering. The selected processes are explicitly highlighted in the paper as: "*planning of activities and responsibilities for various tasks; assessment, control, and decision-making concerning organizational processes, time management, tasks, design concepts, and chosen methods; stakeholder requirements definition, requirement analysis, and architectural design of the given system; implementation, verification, and validation of the system design*". Methods related to these processes are taught by examples, after which the students (as individuals or teams) make their choice in terms of which method may be best suited for each task and apply them to the use case of the laboratory. By this means, the students do not only acquire the Systems Engineering methods, but also several important soft skills, such as moderation, presentation, and discussion.

A similar approach has been adopted at Cranfield University, UK, but as a part of a whole Masters degree program in Systems Engineering. It takes form of a one-week full-time laboratory based on a LEGO Robotics Kit [25].

2.7 LEGO-Based Programs

LEGO Robotic kits, especially LEGO Mindstorms, have been widely adopted for educational purposes. They are mainly used for two goals. The first one is to stir the interest of high school students for STEM education (Science, Technology, Engineering and Mathematics), by allowing them to discover scientific fields like electronics, mathematics, design or programming. Relevant examples are the efforts at Wichita State University [26], or the Stevens Institute of Technology BUILD IT Project, which is a university-school collaboration to increase interest and achievement in engineering, science, mathematics, and information technology [27]. It was also used by Georgia Tech School of Electrical and Computer Engineering to help students decide whether or not to major in electrical engineering or in computer engineering [28]. The second goal is using LEGO Mindstorms to teach future engineers the Systems Engineering approach. Two significant experiences are discussed in this section.

Khalaf et al. [29] propose an innovative and interdisciplinary engineering *Design-and-Build* course for the cornerstone level, to improve three aspects of Design Engineering Education: placement, content, and pedagogy. The authors use an inductive problem-based learning method of delivery through open-ended problems inspired from industry, a LEGO Mindstorms robotics kit, a C++ interface, and a 3D printer. The student teams iteratively proceed through four predefined stages, from problem formulation, to conceptual design, to preliminary and detailed design, and finally to design communication.

For student assessment, the authers created a 24 statements survey for their design course, containing eight statements for each of the three considered dimensions: problem-solving, teamwork, and communication. They evaluate the acquisition of expertise and not the skills, by making students take the survey twice, once before and once after the cornerstone design course, so that gains in favorable aptitudes and attitudes can be analyzed based on pre-/post- test scores. This solution is a good implementation of problem-based learning, especially when it comes to student and process assessment, however it is not a technology-based solution even if it deals with Robot Control, because it does not support the engineering design process using advanced technological tools. Moreover, the solution is not appropriate for student teams working from geographically distant locations, and the assessment is only based on subjective methods by asking students to fill in a survey.

In order to teach Systems Engineering fundamentals and to raise interest in STEM education in the United States, Patel et al. [30] proposed a more complete and innovative model that takes the form of a challenge, through an engineering-based product development Capstone project for US K-12 students, and also for Cornerstone undergraduate students. Unlike the previous solution, it is technology-centered and incorporates some key principles of Systems Engineering in the provided teaching model.

The authors implement their teaching model through an Integrated Design and Manufacturing Infrastructure (IDMI). It is essentially based on CATIA V6, a commercial tool from Dassault Systèmes, and employs both virtual resources such as CAD systems, and physical resources such as 3D printers. The solution includes five modules:

- *Introduction to Product Life Cycle Management: using provided video tutorials.*
- *Introduction to Systems Engineering Principles using moderately complex Electro-Mechanical Systems: LEGO Mindstorms*

- *Computer Aided Design: using Dassault Systemes CATIA V6*
- *Additive Manufacturing: using 3D printers and STL files*
- *Collaborative Tools: using SwYm (See What You Mean), Dassault Systemes online social network.*

The solution has been experimented in the Prize Challenge Summer Camp at Georgia, where students went through different stages of a product life cycle: Co-create, Design, Build and Operate, to build a LEGO Mindstorms-based product for the challenge. Compared to other Systems Engineering project-based courses and programs, this solution appears to be the most complete and the most efficient until now, especially with regard to the intergration of technological tools.

2.8 Other Programs

Many other methodologies are used to teach Systems Engineering especially in the United States. Szajnfarber et al. [31] studied some of them and classified them into *Quizzes, Lab Reports, Design Projects, Arduino Projects, Exams, Homework, Labs, Lecture and class discussion, Predominately Exams and a Design Project, Design Challenges, Research Papers, Research Projects, Case Studies.*

3 Project-Based Learning

General Engineering Education was for a long time centered around discipline-related knowledge like Mathematics, Physics or Mechanics. More recently, there has been a significant shift of focus towards softer skills, such as design thinking and systems thinking, as requested by industry [32]. This change in Engineering Education was motivated by employers who expressed their need for engineers who are not only experts in their domains, but also adept communicators, good team members and lifelong learners [12].

According to Dym et al. [12], ”*the currently most-favored pedagogical model for teaching Design is Project Based Learning*”. Despite the differences between design and systems thinking (a core aspect of the Systems Engineering discipline) [33], both Engineering Design and Systems Engineering mostly deal with processes and skills, and not with transferable and fundamental knowledge. Engineering Design is defined as *”a systematic, intelligent process in which designers generate, evaluate, and specify concepts for devices, systems, or processes whose form and function achieve clients' objectives or users' needs while satisfying a specified set of constraints.”* [12]. Therefore, we can assume that Project-Based Learning (PBL) is actually the most-favored pedagogical model for both Engineering Design and Systems Engineering Education. According to Khalaf et al. [29], the nature of these disciplines is in inherent alignment with the PBL pedagogy, so PBL is recommended especially for developing analytical and problem-solving skills needed to address multidisciplinary and complex engineering problems.

Most of Systems Engineering Education programs that we presented previously, especially the ones that deal with LEGO or other in-practice situations, make use of PBL as their pedagogical model.

3.1 Inductive Teaching

Traditional *deductive teaching* of Engineering and Science starts with theory and progresses to their applications. The educator typically introduces a topic and explains its general principles, derives mathematical models from these principles, presents illustrative applications, makes students practice on similar applications, and finally tests their acquired knowledge in an exam [34].

In contrast to the deductive teaching approach, *inductive teaching* is about letting and helping students discover and learn theories only after the need to know them. This process is usually started by the educator presenting specific observations, case studies or problems. Some examples of inductive methods are: problem and project-based learning as two different methods, discovery learning, just-in-time teaching, etc. In the context of this paper we are interested in Project-Based Learning, as defined by Prince et al.: *"Project-based learning begins with an assignment to carry out one or more tasks that lead to the production of a final product-a design, a model, a device or a computer simulation. The culmination of the project is normally a written and/or oral report summarizing the procedure used to produce the product and presenting the outcome"* [34].

3.2 Current PBL Challenges

For many researchers and educators, PBL seems to be the most adequate model to teach general engineering, and especially for Systems Engineering. However, this does not mean that PBL is a perfect model that does not suffer from some limits, or that does not need adjustments to be more efficient. It remains some open research questions regarding PBL. Some of them have been identified by Dym et al. [12]

- *What are the best proportions of problems, projects, teamwork, technology, and reality for a given state of student development? In other words, how authentic should PBL experiences be compared to industry design experiences? Some work has begun to emerge in this area, but the answers are not yet definitive.*
- *How do the proportions change with regard to the context of different engineering disciplines and institutional missions?*
- *How should multidisciplinary design-learning teams be managed?*
- *Can a pedagogic framework developed for co-located learning teams be distributed in time and place? If so, how?*
- *How can students be authentically evaluated and graded in design courses with regard to, for example,*
  - *the quality of the design produced vs. the quality of the process demonstrated; and*
  - *individual cognitive development vs. collective team development?*

Along with the next open questions, these PBL challenges, related to Systems Engineering Education, represent the first open question that needs to be considered.

4 Open Research Questions

4.1 Technologies

Special attention to new and disruptive technologies may have a good impact on Systems Engineering Education. As stated by Martin, ”*the capabilities and limitations of technology must be considered when developing a systems engineering development environment*” [35].

We have to find the appropriate way to integrate technology into Systems Engineering experiences, by idetifying the advantages of a specific technology and matching it to the specific requirements and problems to solve. Technology should not be used just for the purpose to use technology. A first technology we suggest to consider is Virtual Reality.

According to ”The Treaty of Virtual Reality” [36], Virtual Reality (VR) is a technology that allows users to immerse in an artificial reality and to have interactive experiences via sensorimotor channels. In [37], the authors discussed the work of Abulrub et al. [38] who explored the benefits of VR for engineering education and training through a series of case studies at the University of Warwick. The authors showed how VR can encourage the creative learning of engineering material and environments, and concluded that some VR promises for Systems Engineering are: the development of autonomous problem solving skills, the sharing of complex information with team members, and the analysis of engineering problems under different points of view.

In the same work, the authors assumed that ”*thanks to its interaction and immersion components that provide the students with a high level of realism and interactivity, VR is a well-suited tool for education and training. It offers a safe, fully controllable and cost-effective learning experience. VR teaches the students how to develop autonomous problem solving skills, and gives the instructor the ability to create realistic learning situations which are difficult, unaffordable or even impossible to set up in a classic learning context*”. It has been shown that teaching and training is considerably improved by having the students apply theoretical knowledge to concrete industrial problems using VR technologies. Creativity, innovation, communication, problem solving, team work and business skills can be improved by using VR environments, which offer an unlimited experience on virtualized real-life situations [38]”.

Many systems will be connected in future factories (Industry 4.0), and we also suggest to investigate the role of Internet-of-Things (IOT) in Systems Engineering Education. We believe that IoT can be of great assistance by providing, for example, the ability to immerse students in realistic situations, thanks to an IoT architecture adopted by an industrial partner. Coupled with Virtual Reality, the IoT sensor information could be projected on a virtual factory where Systems Engineering students work on virtualized copies of the industrial product.

However, an important point to be beared in mind when creating technological-based learning environments for Systems Engineering is user experience. It needs to be as plain and simple as possible, in order not to add unnecessary cognitive load to the student who is already engaged in the assimilation and application of Systems Engineering processes. Attention should be paid when creating learning processes to not make non-technological students use complex computer tools, like management students manipulating sophisticated CAD applications only for visualization or other simple tasks.

4.2 Standards

The learning methods adopted by most of the programs discussed in this paper focus rather on tools and how to use them, than on the fundamental principles of Systems Engineering. Therefore, systems engineers may perfectly use SysML editors and other related tools, while they do not really know and understand Systems Engineering processes, their goals, their execution or the links between them. The solution to this problem, especially, when the goal is to teach the fundamental concepts of Systems Engineering, may be to consider the use of Systems Engineering standards, such as the most recent 2015 edition of the ISO/IEC/15288 standard [3]. It define a set of processes and associated terminology from an engineering viewpoint. One way on how to use this standard for educational purposes might be to give related materials or lectures to students, and ask them to respect the standard while they realize their own project. However, even if we consider this solution, we still need to find responses to some questions such as: which set of processes can and should be adopted in the learning process? How much detail does the teacher give to the students on how to execute the processes, which tools to use for each process and for each activity and task? How to ensure that the students really pass through the correct processes, the right way and the right time?

4.3 Assessment

Another point that needs specific attention is the ability to efficiently evaluate student work and the acquired knowledge and skills. For that purpose, we suggest four types of assessment which any Systems Engineering Education solution should take into account:

- Result Assessment: Evaluating the quality of the outcome compared to the project requirements (expressed by the teacher or an industrial).
- Execution Assessment: Evaluating how well Systems Engineering Processes (standard or not) have been applied during the project.
- Knowledge Assessment: Evaluating the acquisition of fundamental knowledge on Systems Engineering.
- Skills Assessment: Evaluating the learned soft skills, such as collaboration, communication, team work, etc.

But the remaining question is how to deploy and efficiently apply these criteria, depending on the nature of the project?

5 Conclusion

In this paper, we investigated the main objectives of Systems Engineering Education and presented the current state-of-the-art in this field. We particularly discussed the most suited pedagogical model, which is Project-Based Learning. We also provided some open questions that need further research. This study may be a valuable starting point for educational institutions aiming to create solutions for Systems Engineering Education. It also highlights some challenges that might interest researchers from both educational and engineering sciences. It appears very clearly from this study, that specific effort needs to be dedicated to the subject of students assessment in PBL/Systems Engineering context, and also to facilitate the adoption of Systems Engineering standards and enhancing their roles in Systems Engineering Education solutions.

From our perspective, this survey helped especially in defining the specifications of our future platform for Systems Engineering Education, which is under development. In fact, in order to address the previously cited challenges, we are creating a new Collaborative Platform for Systems Engineering Education. For that, we are using some of the most recent and most disruptive technologies, such as Virtual Reality, Web 3D and Internet of Things. Our solution is a Standard-Based Solution, because the proposed learning scenario is based on Systems Engineering Standard Processes. By doing that, our goal is to make learners use the standardized processes of Systems Engineering, when engineering the requested system in a Systems Engineering learning environment. Also, we aim to be able to assess their acquired knowledge and skills, along with the processes execution quality and the obtained results.

References

[1] Cecilia Haskins. *Systems Engineering Handbook, a guide for system life cycle processes and activities*. INCOSE, no, 3.2.2 edition, 2012.

[2] Guey shin Chang, Horng linn Perng, and Jer nan Juang. A review of systems engineering standards and processes. *Journal of Biomechatronics Engineering*, pages 71–85, 2008.

[3] Iso/iec/ieee international standard - systems and software engineering – system life cycle processes. *ISO/IEC/IEEE 15288 First edition 2015-05-15*, pages 1–118, May 2015.

[4] T. S. S. Subramanian, M. Mukundlal, P. Prempyari, and P. Dubey. Systems engineering: A new approach to engineering education in india. In *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*, pages 1–3, July 2012.

[5] Mohammed Bougaa, Stefan Bornhofen, Hubert Kadima, and Alain Rivière. Virtual reality for manufacturing engineering in the factories of the future. *Applied Mechanics & Materials*, 2015.

[6] Gerrit Muller. Didactic recommendations for education in systems engineering, 2010.

[7] Clive L Dym. Design, systems, and engineering education. *International Journal of Engineering Education*, 20(3):305–312, 2004.

[8] O.A. Asbjornsen and R.J. Hamann. Toward a unified systems engineering education. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(2):175–182, May 2000.

[9] A.P. Sage. Systems engineering education. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(2):164–174, May 2000.

[10] Systems engineering capability model interim standard, eai/is 731. 1999.

[11] Engineering education for a changing world. *American Society for Engineering Education*, 1994.

[12] C.L. Dym, A.M. Agogino, O. Eris, D.D. Frey, and L.J. Leifer. Engineering design thinking, teaching, and learning. *Journal of Engineering Education*, 94(1):103–120, 2005.

[13] Jan Herrington and Lisa Kervin. Authentic learning supported by technology: Ten suggestions and cases of integration in classrooms. *Educational Media International*, 44(3):219–236, 2007.

[14] INCOSE Website. Policy on se education, 2016. Accessed: 2016-02-04.

[15] Christine Ng. Findings from a web-based survey of degree programs in engineering systems.

[16] K. Craig and P. Voglewede. Multidisciplinary engineering systems graduate education: Master of engineering in mechatronics. In *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, 2010 IEEE*, pages 1–14, April 2010.

[17] Ronald R. Luman Samuel J. Seymour. Academic perspectives of systems engineering. *JOHNS HOPKINS APL TECHNICAL DIGEST*, 29(4), 2011.

[18] SupMeca Website. Placis: Collaborative platform for systems engineering, 2016. Accessed: 2016-05-20.

[19] Jean-Claude Tucoulou, David Gouyon, and Eric Bonjour. A student challenge in systems engineering: RobAFIS 2010. *INSIGHT - International Council on Systems Engineering (INCOSE)*, 14(4):9–11, December 2011.

[20] G. M. Bonnema, I. F. Lutters-Weustink, and F. J. A. M. van Houten. Introducing systems engineering to industrial design engineering students with hands-on experience. In *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on*, pages 408–413, Aug 2005.

[21] Benjamin S. Blanchard and Wolter J. Fabrycky. *Systems Engineering and Analysis*. Prentice Hall, New Jersey.

[22] Jr Andrew P. Sage, James E. Armstrong. *Introduction to Systems Engineering*. John Wiley and Sons, Inc, New York.

[23] M. Kudret Yurtseven. Teaching systems thinking to industrial engineering students.

[24] W. Bauer, W. Biedermann, B. Helms, and M. Maurer. A student laboratory for systems engineering: Teaching systems engineering to students without previous se-knowledeg based on an industry-oriented example. In *Systems Conference (SysCon), 2012 IEEE International*, pages 1–6, March 2012.

[25] C. B. Nielsen and P. Adams. Active learning via lego mindstorms in systems engineering education. In *Systems Engineering (ISSE), 2015 IEEE International Symposium on*, pages 489–495, Sept 2015.

[26] L. E. Whitman and T. L. Witherspoon. Using legos to interest high school students and imtrove k12 stem education. In *Frontiers in Education, 2003. FIE 2003 33rd Annual*, volume 2, Nov 2003.

[27] Beth McGrath, Jason Sayres, Susan Lowes, and Peiyi Lin. Underwater lego robotics as the vehicle to engage students in stem: The build it project's first year of classroom implementation. *American Society for Engineering Education Mid-Atlantic, Hoboken, NJ*, 2008.

[28] D. B. Williams. A freshman 'introduction to ece design' course based on the lego mindstorms nxt. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pages 435–439, Jan 2009.

[29] KINDA Khalaf, SHADI Balawi, GEORGE WESLEY Hitt, and AHMAD Radaideh. Engineering design education: Where, when and how. *Journal of American Education*, 2013.

[30] Srujal Patel, Dirk Schaefer, and Daniel Schrage. A pedagogical model to educate tomorrow?s engineers through a cloud-based design and manufacturing infrastructure ? motivation, infrastructure, pedagogy, and applications. In *ASEE 2013 Annual Conference and Exposition*, 2013.

[31] Zoe Szajnfarber Madeleine C Brannon and Thomas Andrew Mazzuchi. Introducing the fundamentals of systems engineering to freshman through various interactive group activities. In *2014 ASEE Annual Conference*, Indianapolis, Indiana, June 2014. ASEE Conferences. https://peer.asee.org/20705.

[32] Aruna Shekar. Project-based learning in engineering design education: Sharing best practices. In *2014 ASEE Annual Conference*, Indianapolis, Indiana, June 2014. ASEE Conferences. https://peer.asee.org/22949.

[33] John Pourdehnad, Dennis Wilson, and Erica Wexler. Systems & Design Thinking: A Conceptual Framework for Their Integration. In *Proceedings of the 55th Annual Meeting of the ISSS-2011, Hull, UK*, volume 55, 2011.

[34] Michael J. Prince and Richard M. Felder. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, 95(2):123–138, 2006.

[35] James N Martin. *Systems Engineering Guidebook: A Process for Developing Systems and Products*. CRC Press, Boca Raton.

[36] no. *The Treaty of Virtual Reality: Foundations and Behavioral Interface*. Ecole des Mines press, France, 2 edition, 2006.

[37] Mohammed Bougaa, Stefan Bornhofen, Hubert Kadima, and Alain Rivière. 3d interaction in virtual environments for systems engineering. *International Journal of Computer Theory and Engineering*, 8(6):458, 2016.

[38] A.G. Abulrub, A.N. Attridge, and M.A. Williams. Virtual reality in engineering education: The future of creative learning. In *Global Engineering Education Conference (EDUCON), 2011 IEEE*, pages 751–757, April 2011.

Second Paper: (ASEE) Toward a Process-Centered Approach for Systems Engineering Education

- **Paper title:** Toward a Process-Centered Approach for Systems Engineering Education.
- **Authors:** Mohammed Bougaa, Stefan Bornhofen, Alain Rivière, Jean-Claude Tucoulou.
- **Conference:** 2017 ASEE Annual Conference and Exposition
- **Location:** Columbus, Ohio, USA.
- **Date:** June 25-28, 2017
- **Keywords:** Systems Engineering Education,Systems Engineering Standards,Competency Models,System Lifecycle Model,Project Based Learning.

Toward a Process-Centered Approach for Systems Engineering Education

Keywords Systems Engineering Education, Systems Engineering Standards, Competency Models, System Life Cycle Model, Project Based Learning.

Introduction

Time has gone when industrial companies recruited their employees only based on a fine technical background. Nowadays, these companies deal with complex and multidisciplinary systems, and their mastering requires much more than mere technical excellence. Today's engineers need to be good team workers, adept communicators, and lifelong learners [1]. In addition to producing the expected client outcomes, a major engineering project has to satisfy various stakeholders while ensuring an optimization of time, cost, energy and other resources throughout its entire life cycle. In view of these challenges, a growing number of companies turn to the Systems Engineering (SE) approach, a discipline initially reserved for big defense and aerospace companies. "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionalities early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem".[2]

According to Wasson [3] "*Unfortunately, the engineering of systems, performed in many organizations is often characterized as chaotic, ineffective, and inefficient. Objective evidence of these characteristics is exemplified by noncompliance to requirements, cost overruns, and late schedule deliveries in program metrics for a project's contract or task triple performance constraints– i.e., technical, cost, and schedule.*". Based on his experience, the author suggest that *"many engineers are estimated to spend on average from 50% to 75% of their total career hours collaborating with others concerning the engineering of systems – i.e., systems engineering - for which they have no formal education"*

Significant efforts are made to improve the discipline of SE, by both industrial and academic players. On the one hand, industries and governments develop standards, norms, competency models, and documentation [4], while applying the SE approach in their projects and promoting it to smaller entities [5] [3]. On the other hand, academic institutions and researchers, most often in collaboration with industries, investigate new paths to teaching SE. They are typically interested in defining *competencies* which best characterize a system engineer, in order to design an efficient *pedagogical model* and an appropriate *learning environment*. In addition to these questions, the present paper particulary focuses on SE standards and on how they can and should be used for SE learning purposes.

The next section of this paper presents a state of the art introducing a number of significant works related to SE education. The following sections convey our own vision of teaching SE, together with a presentation of our developed solution as well as survey results regarding its usefulness and ease-of-use. The paper concludes by highlighting our main contributions and discusses the perspectives of our approach.

Background

Over the last decade, governments, universities, engineering schools and industrial companies have been dedicating much attention to the practice of SE. Various aspects have been addressed including people, processes and technology. In the scope of our study, we are mostly focusing on people and how to make them most efficiently learn the fundamental principles of SE. ”*Traditionally, systems engineering competencies have been developed primarily through experience, but recently, education and training have taken on a much greater role”* [6]. The following list compiles a number of significant advances:

Systems Engineering competencies

According to the Systems Engineering Body of Knowledge book (SEBoK) [6], SE competencies reflect the individual’s Knowledge, Skills, Abilities, and Attitudes (KSAAs), which are developed through education, training, and on-the-job experience. According to the same source, ”*For an individual, a set of KSAAs enables the fulfillment of the competencies needed to perform the tasks associated with the assigned systems engineering role*”.

A set of SE competencies form a SE competency model which reflects the individual’s KSAAs. The KSAAs are in turn related to different roles in the company or the project, so that they are associated to a set of tasks. A competency model is therefore a framework for organizing a collection of observable KSAAs. According to [6], ”*SE competency models generally agree that systems thinking, taking a holistic view of the system that includes the full life cycle, and specific knowledge of both technical and managerial systems engineering methods are required to be a fully capable systems engineer*”.

KSAAs can be used as learning objectives for SE competency development, especially when they are defined in terms of a standard taxonomy, as in [7]. Authors designed a SE competency career development model as an analytical approach using Bloom’s taxonomy. In addition to their use in education, training, and development, competency models can also be used for recruitment and selection, human resources planning and placements [6].

Various competency models exists in the field of SE. Most of them have been developed for specific contexts, since the required competencies can differ between organizations and projects, and they can typically be tailored to the organization or project particularities. The most well-known competency models in the field of SE are:

- INCOSE UK Working Group Competency Model: identifies the competencies required to conduct good SE projects[8].
- Defense Acquisition University (DAU) ENG Competency Model: identifies the competencies required for Department of Defense (DoD) acquisition engineering professionals [9].

- NASA Academy of Program/Project and Engineering Leadership (APPEL): identifies a project management and SE competency model to improve project management and SE at NASA [10].
- The MITRE institute SE competency model: defines new curricula for SE and assesses personnel and organizational capabilities [11]
- INCOSE multi-level professional Systems Engineering Professionals (SEP) certification program: provides a formal method for recognizing the knowledge and experience of systems engineers, regardless of their current point in career [12].

Most organizations tailor those models by including domain-specific KSAAs and other particularities of their organization. Also, several models can be used together and merge into a new competency model, as suggested by White [7]

The role of standards in SE education

Some of SE standards describe and provide a framework for system life cycle processes, such as in ISO/IEC/IEEE 15288 [13] or ISO/IEC 20110 [5]. The relation between SE competency models and a system life cycle processes is explained in [6], "SE competency must be viewed through its relationships to the systems life cycle, the systems engineering discipline, and the domain in which the engineer practices systems engineering" [6].

In this paper, we support the use of such SE standards as the basis of a SE education approach, while being in compliance with a SE competency model. As a matter of fact, these standards encompass the fundamental principles of SE which is exactly what we want to teach.

The adequate pedagogical model for SE education

According to Khalaf et al. [14], the nature of the SE discipline is in inherent alignment with the Project-Based Learning (PBL) pedagogy. PBL is especially recommended for developing analytical and problem-solving skills which are necessary to address multidisciplinary and complex engineering problems. According to Dym et al. [1], "*the currently most-favored pedagogical model for teaching Design is Project Based Learning*". Despite the differences between design and systems thinking (a core aspect of the SE discipline) [15], both engineering design and systems engineering mostly deal with processes and skills, and not with transferable and fundamental knowledge. engineering design is defined as *"a systematic, intelligent process in which designers generate, evaluate, and specify concepts for devices, systems, or processes whose form and function achieve clients' objectives or users' needs while satisfying a specified set of constraints."* [1]. Therefore, it can be assumed that PBL is actually the most appropriate pedagogical model for both engineering design and SE Education.

Even though PBL seems to be the most adequate model for teaching SE, there are a number of open research questions and challenges regarding this pedagogical model. Some of them have been identified by Dym et al. [1]

Current practices in SE education

Interstingly, current SE education programs do not pay much attention to the design of competency models, nor to the adoption of SE standards. In [16], we surveyed the current practices in SE education published by the European Society for Engineering Education (SEFI), and classified them into 8 categories:

- Master programs with academia-industry partnerships [17] [18].
- Few-months international academia-industry projects [19].
- Student challenges [20].
- Few-weeks projects within regular engineering curriculum [21].
- Theoretical courses within industrial engineering curriculum [22].
- Few-Days Laboratories [23] [24].
- LEGO-Based Programs [25] [26] [27] [14] [28]

Other less prominent SE education approaches exist, some of which can be found in [29], such as Quizzes, Lab Reports, Design Projects, Arduino Projects, Exams, Homework, Labs, Lecture and class discussion, Predominately Exams and a Design Project, Design Challenges, Research Papers, Research Projects, and Case Studies. For a more detailed compilation, Incose and the Systems Engineering Research Center (CERC) at Stevens Institute of Technology has published a 116 pages document called "*2016 World Wide Directory of systems engineering and industrial engineering academic programs*" [30]. This report lists the name of universities offering degrees in SE, and provides detailed information. It can be concluded that academia is interested more than ever in SE. However, there is no common teaching model, except for some recommendations and specifications, as highlighted in [16].

Considering the requirements of an efficient learning environment for SE [16], the next sections present our vision of how to improve SE learning experiences by adopting international standards. We are particulary interested in standards concerning the systems life cycle, such as ISO/IEC/IEEE 15288 [13] or the simplified ISO/IEC 29110 [5].

Global proposed approach and its main components

Our vision regarding the best suited environment for teaching SE focuses on new disruptive technologies such as Virtual Reality (VR), Internet of Things (Iot), 3D Printing and Machine Learning, coupled with a Project Based Learning (PBL) model, and a process based learning path. Of course, this vision isn't to be deployed today. However, in this paper, we present our current work results, that might lead to achieving this vision in the future. We aim to make the teaching of SE fundamentals true to reality. Therefore, we promote the learning-by doing-paradigm, where multidisciplinary students from different locations collaborate to engineer a system requested by an educator, this is what we call a system of interest[13]. Students can adopt different roles such as designer, production operator, requirements engineer, architect or tester. In particular, they are guided to apply SE standard processes and therefore meet situations similar to real-life SE challenges.

Another accordance with real-life SE is the fact that our approach is based on two main components, a virtual and a physical environment, operating through an Internet-of-Thing (IoT) infrastructure. A high level of connectivity between these two environments is needed, not just at the engineering level, but also with respect to the teaching process. The educator is able to track the learning activities inside both the physical and the virtual environments, in order to assist and evaluate students knowledge acquisition.

The global approach will therefore be a domain independent solution used by both educators and students in SE education organizations, enabling a high level of collaboration and interaction.

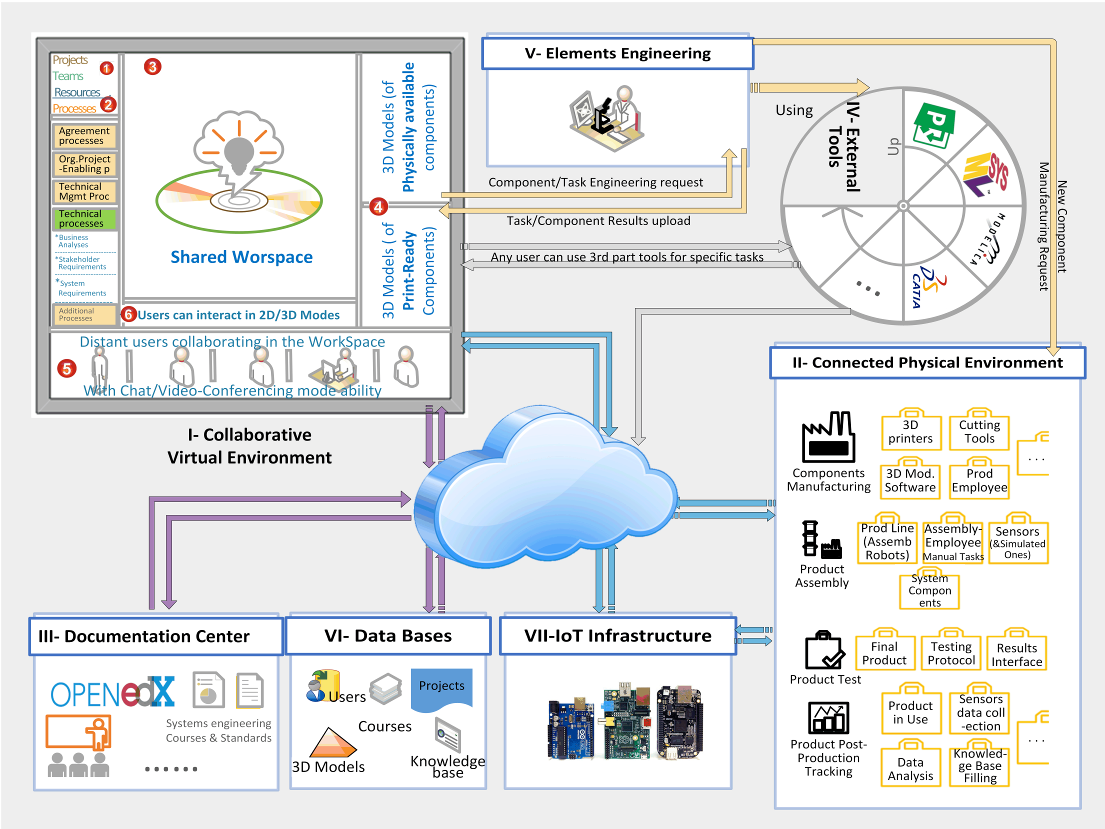

Figure 1: Main components of the proposed SE education approach

Primitive resources

Primitive resources are the atomic components used to create a new system or to modify an existing one. For each primitive resource, students have at their disposal a 3D model inside the *Collaborative Virtual Environment* (CVE), and a corresponding assembly part inside the physical environment. However, in order to satisfy the particular requirements or their mission, they may need additional components. In this case, students design a 3D Model of the missing piece, using the *elements engineering* component of this approach, and they produce the physical unit inside the *physical product assembly environment*, e.g. using a 3D printer.

The collaborative virtual environment

This is the main component of our approach, representing the engineering workspace. It is intended to be a web-based application where students can collaborate, and where they can interact with educators throughout the whole project. Figure 2 show that this environment incorporates several elements, which are:

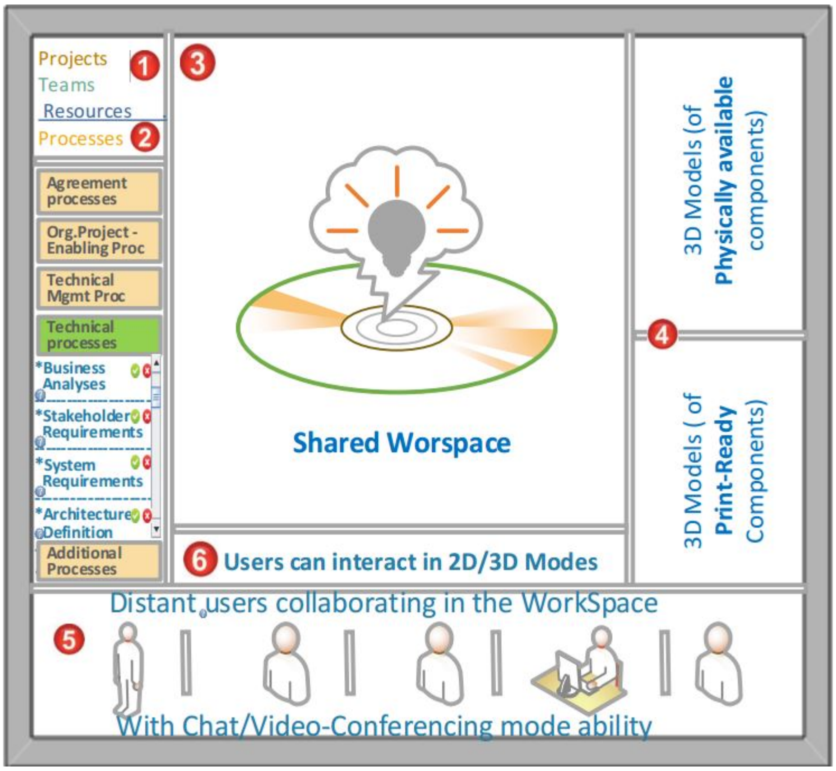

Figure 2: ICVE Elements Description

- Projects, teams, and resources **(1)**: The top left part of the screen shows a three-component menu. Its first element, *Projects*, includes a description of the project mission, given out by the educator. By clicking on the *Teams* element, students find information about the other members of their teams co-working on a specific project. They can also manage the different roles assigned to each student during the product. Under the third element of this menu, *Resources*, students find a collection of suggested resources provided by the educator, to guide them through the engineering processes. The educator plays two different roles. First, he can act as the acquirer of the system-of-interest, but another entity can replace this role and define the project mission. Second, the educator plays the teacher role, assisting students throughout the entire process of engineering the system, and evaluating them from an individual and a collective point of view.

  Prior to that, students must be recorded in the students data base, with their biography, curriculum and skills. This information helps the educator to efficiently perform student-team-project assignments. A team of students can be responsible of the engineering of an entire system, or their system can be an element of a bigger one. In this case, the educator assigns teams to specific parts of one higher-level project.

- Life cycle model processes **(2)**: This is one of the most important parts of the collaborative virtual environment where students follow the life cycle model processes, in order to engineer the requested system. The life cycle model is defined by the acquirer/educator based on the learning goals (SE competencies) and the nature of the system. In addition, for more information about the currently used processes, or for further training regarding the used resources and methods, students can always access the *Documentation Center* which may be a LMS or a MOOC platform.

- The shared workspace **(3)**: Represents a virtual place where students can report the results of their performed tasks. All team members, including the educator, have a complete overview of their progress at any time, and they are able to annotate and exchange work.
- 3D virtual models **(4)**: The shared workspace gives access to the 3D models that can be used as primitive elements in the design process. These models may already exist in the physical world, or they can be 3D-printed to assemble the final system-of-interest. As illustrated by **(6)**, students are able to interact with the shared workspace, both in 2D and 3D modes, depending on the nature of their task.
- Collaboration **(5)**: Using the collaborative virtual environment, students and educators can communicate and exchange through a chat or video-conferencing system.

The physical environment

The Physical Environment represents the traditional manufacturing factories and production lines for assembly. We distinguish the *Manufacturing Environment* with activities relative to new components production, using tools and machines such as 3D printers, and the *The Assembly Environment*, which may include a robot based production line for components assembly, with the help of an *assembly operator*. Moreover, we assume that all components contain sensors that allow a real-time tracking of the assembly operations, reporting relevant data to the CVE through an adequate IoT architecture. The physical environment includes a *testing environment* where the operator can perform a series of test procedures on the assembled system-of-interest, and report the results to the CVE. This approach is particularly interesting if the IoT infrastructure allows a post-production tracking of the system-of-interest, allowing for additional tests to be directly performed from within the virtual environment.

Learning and documentation center

The documentation center is a virtual space where students have easy access to educational resources, such as documents and videos, online libraries, LMS and MOOC platforms, etc. By this means, students can find the appropriate time to learn more about different aspects of the SE discipline, including its standards, processes, methodologies, and also consult useful information and tutorials about SE related tools.

System elements engineering

This part of the global vision is related to a scenario where students are asked to design a required component for the system-of-interest, which turns out to be so complex in itself that it needs collaboration between different students in the team or between the different teams. In that case, the component can be considered as a new system-of-interest, and students have to engineer it by applying the same SE life cycle processes as for the higher-level system-of-interest. After performing each task, the outcomes are uploaded to the collaborative virtual environment.

Third-party tools and resources

A good SE learning solution should be highly tool-independent, so that every organization, educator and student can choose the most adequate tool for their SE activities. Ideally, students should be able to access their tools directly inside the collaborative virtual environment. Otherwise, they may resort to external tools and upload the results.

Implementation

After the previous discussion of our global solution for SE education, this section presents our progress concerning its implementation. In the scope of this paper, we propose a solution for teaching SE, which promotes the use of SE international standards and allows students to learn through a project-based learning approach.

At its current stage, the solution allows working with only one kind of processes, the technical processes, where students can engineer their systems without dealing with other activities related to management, agreement, or project-enabling processes. Students can be asked to use the technical processes of a given standard, such as the ISO/IEC/IEEE 15288 [13], or the ISO/IEC 29110 [5]. However, educators are free to define other process flows, by adding new unstandardized processes, or by inserting processes from other standards. Two use-cases applying our solution are described in [31].

In this solution, students and educators pass through well-defined scenarios. The reader can find the most important components of the solution in Appendix A.

- Main learning scenario

  This represents the high-level learning scenario. As illustrated in Figure 3, it encompasses other sub-scenarios. The proposed solution has two main players. On the one hand, educators are responsible for creating projects, by defining their goals and life-cycle models. They also assign student teams, ensure assistance and assessment. Students, on the other hand, are responsible for collaboratively engineering the system, with respect to the processes defined by the educator.

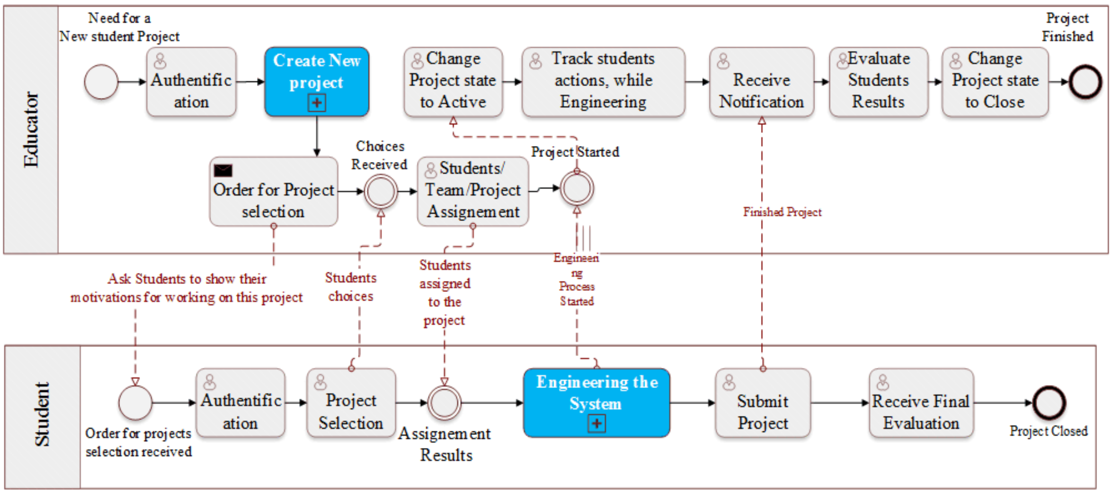

Figure 3: Global Learning Scenario

Note that it is possible to extend the use of standard processes even to the definition of the mission objectives. As a matter of fact, the solution may implement the acquisition process, the first process of the ISO/IEC/IEEE 15288 agreement processes, where educators and students can negotiate and agree on the work to be done.

- Educator: Project creation scenario

  As illustrated in Figure 4, creating a new project goes through several stages. The educator defines the project title and description, as well as the life-cycle model which will be followed by students. For this purpose, the educator selects a number of processes from the processes database. If a specific process does not exist in the database, it can be added using the processes management system, as illustrated by Figure 5. Finally, the educator specifies the resources and tools to be used by students.

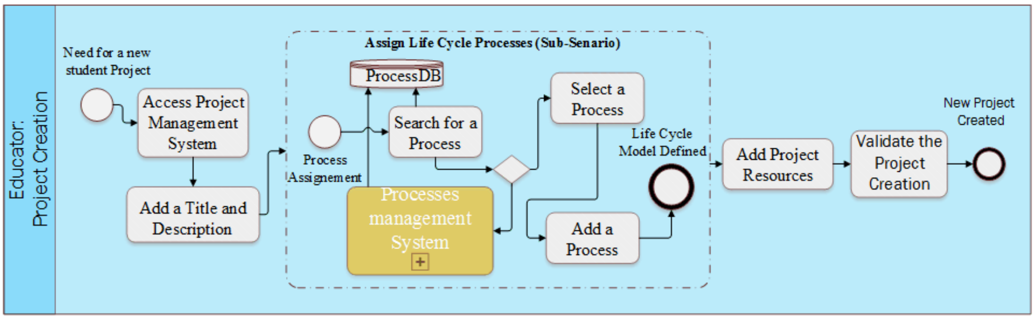

Figure 4: Project creation scenario

For the time being, the life-cycle model is defined by the educator. In the future, it may be possible that students with a solid background in SE are asked to define the life-cycle model by themselves.

- Educator: Processes management scenario

  This scenario allows educators to create, adapt or remove SE processes. The adopted architecture for a process is compliant to the 15288 standard. In addition to its purpose and its outcomes, a process is defined as a set of activities, and each activity is defined as a set of tasks to be performed.

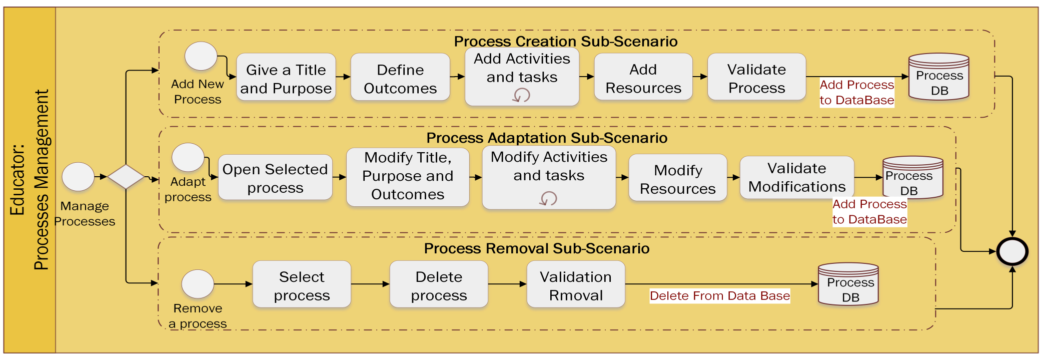

Figure 5: Processes management scenario

• Student: Project engineering scenario

This scenario, described in Figure 6, represents the high-level stages that students will follow to engineer the required system. After selecting an active project, and after managing their roles, students engineer the system by performing the tasks of each activity of the life-cycle model processes. The tasks are done using the adequate tools and methods, and their results are uploaded to the project work-space.

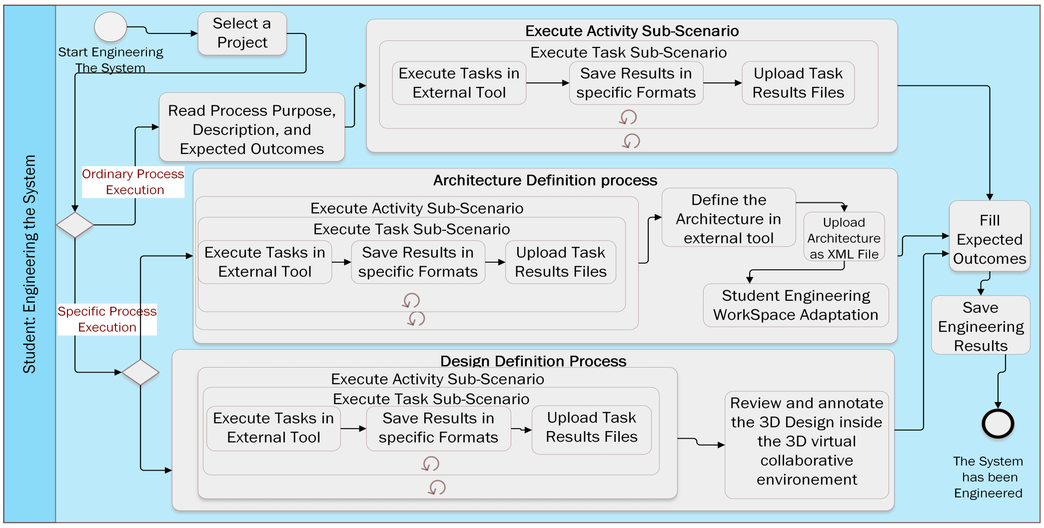

Figure 6: Project Execution scenario

All processes are executed in the same way, except for the *system architecture definition process* and the *system design definition process*, which include some specific tasks that can be executed inside the solution, as described next.

• Student: Environment Adaptation According to System Architecture Scenario

The system architecture is defined by students and uploaded to the project workspace in form of an xml file. Depending on the chosen architecture, the project workspace is automatically adapted to provide students sub-workspaces for each subsystem or system element. Consequently, students can engineer each subsystem and system element as a system, by passing through the entire life-cycle model.

• Virtual design scenario

Virtual 3D representation and the design of a tangible system are indispensable in any system design process. In our solution, we propose virtual system assembly in a collaborative mode, where students in different locations are able to visualize and interact with the system being assembled in real time. Currently, the virtual design is possible for a limited number of pieces. If users want to use the 3D features, they can only engineer systems based on the LEGO® MINDSTORMS® Education EV3 Bricks and their additional elements (Sensors, Motors...etc). However, if 3D virtual design is irrelevant for students and educators, the solution can be used to engineer any kind of systems. The 3D visualization can also be used for simple design review instead of system assembly.

- Collaboration

  Collaboration is enabled and encouraged in this solution, allowing students to work together on different tasks of the same project, while maintaining a global vision of the work all along the engineering scenarios. More features are planned to enhance collaboration, such as annotation of results, or chat and video-conferencing.

- Assessment

  Student assessment is out of scope of this paper. However, we profoundly believe that this solution will help implementing new students assessment methods, allowing educators to objectively evaluate students with respect to various teaching objectives. We already identified a number aspects that need to be considered for assessing students in a SE learning experience as: *result assessment, execution assessment, knowledge assessment, and skills assessment.*

Results and discussion

In this section, we discuss the acceptance of our solution by the SE academic community. We targeted a specific public within this community with a presentation of the solution and a survey.

Methodology

The targeted public were students and tutors participating in the 2016 Robafis challenge. Organized since 2006 by AFIS, the French chapter of INCOSE, the Robafis challenge is a yearly student competition for robot design [20], whose main goal is the promotion of SE. About ten student teams from French universities and engineering schools participate in this competition. Each team can consult a SE teacher, and they can also question AFIS experts. The road-map starts about eight months before the final stage competition, when AFIS communicates the general schedule, the regulations, specifications, and a reference development document. Three months before the final stage, the teams register and receive a LEGO Mindstorms Robotics kit, in order to physically implement their solution. Fifteen days before the final stage, the teams send their development document to systems engineering experts for evaluation. The competition concludes by a final stage where all teams meet and operationally validate their works, along with project and configuration audits. Few weeks after the competition, students receive a detailed debrief regarding their work.

In 2016, eight student teams participated in the Robafis challenge. Some mixed teams featured students from several engineering schools. We got in touch with four teams, totaling in 25 students and 9 tutors. The tutors are mainly educators in the field of SE. Since the solution was not shared for public, we produced a video tutorial showing the platform at work and explained its most relevant features. The video was shared with our targeted public, together with a questionnaire they had to answer. The video can be viewed here [32], and the questionnaire used for the educators can be found in appendix B. Note that there are not many differences between the two questionnaires, as we were mainly interested in feedback regarding the implemented/to-be-implemented features of the solution. The few differences will be apparent during the result analysis.

We received responses from ten students and six tutors, which represents a response rate of respectively 40% and 66%. The feedback from both students and tutors showed a high interest in the features of this solution.

Respondents profile

Nine out of ten of students who answered this questionnaire are undergraduate students, whereas the last one is a post-graduate student. 70 % of students estimate that they are beginners in SE, and the most usual way of learning SE were university lectures for 70%, and academic project-based learning for 60%. It appears from their responses that most of them have an overview of different topics of SE, with a focus on three topics, for which more than 90% think that their level is between medium and good: *"Design, Analysis, and Implementation"*, *"Operation and Management"*, and *"Technical Management"*. Also, 80% think they are good, or at least having a medium level in "*Requirements Management*", "*Architecting*", and "*Project-Enabling Management*".

Tutors respondents are mainly academic SE practitioners, exercising in this fields for more than two years for 67% among them, and more than five years for 17%. 50% used to teach and promote SE through *"university lectures"*, *"competitions and challenges organization"*, and *"academic or academy-industry project-based learning"*.

Results from the students perspective

The following list itemizes the received student feedback concerning

- The usefulness of the current features of the solution: The two features that appear not to be very useful, from a students perspective, are *"the ability to add resources to different processes"*, and *"reviewing the 3D design of the system"*, where only 50% estimate that they are useful or necessary. All the other features are considered useful by at least, 60 to 70% percent. For details, please refer to Table 1.

Table 1: Students appreciation about current features usefulness

Current features usefulness	**Manda-tory** %	**Very Useful** %	**Useful** %	**Not very inter-esting** %	**Needs Im-prove-ments** %	**Not useful at all** %
Enabling geo-distributed students to work together	10	10	50	10	20	00
Learning through SE processes	10	50	00	10	30	00
Reporting all tasks results in the same shared space	10	40	20	10	20	00
Virtual 3D design of the system	00	30	20	10	30	10
Processes presentation	10	30	30	10	20	00
Processes related resources	10	20	20	30	20	00

- The ease-of-use of the current features: Most features are considered simple or very simple to use. About 70% to 90% of students share the same opinion about all features, except for the *"virtual 3D design component"*, where only 50% think that it is simple to use, while the other 50% rate this feature to be hard or very hard. See table 2.

Table 2: Students appreciation of current features ease-of-use

Current features ease-of-use	**Very simple** %	**Simple** %	**Pretty hard** %	**Very hard** %
Enabling geo-distributed students to work together	10	60	30	00
Reporting all tasks results in the same shared space	30	60	10	00
Reviewing the 3D design of the system	00	50	40	10
Notification management system	10	60	30	00
Processes presentation	20	50	30	00

- The additional features to be implemented: Table 3 shows that the suggested features which we plan to add to this solution, will be very useful, if not obligatory, except maybe the chat and video-call systems.

Table 3: Students appreciation of additional features usefulness

Additional features usefulness	**Mandatory**	**Useful**	**Not Useful**
Chat system	40	20	40
Video-call system	10	40	50
Assisting student through annotations	30	60	10
Engineering at both system and subsystems level	30	70	00
Tasks management	50	40	10
Direct web-access to SE tools	20	70	10
Direct web-access to learning resources and platforms	10	70	20
Downloading a project synthesis, at any moment	50	50	00

- Students evaluation methods: Students think that they should be evaluated in the context of learning SE, using different methods at the same time. 50% of students agree on evaluating them by the educator throughout the project (evaluation of processes execution quality), and regarding the acquired skills and knowledge (using questionnaires, for example). However, only 40 % of the respondents approve of self evaluation methods and final results evaluation. Regarding peer evaluation techniques, only 30% beleive that this is a good way of evaluation.

- Advantages of this solution compared to their traditional way to learn SE: The ease of use, the implementation of the project-based learning approach, and the use of SE standard processes are the most appreciated features of this solution (respectively by 50%, 50%, and 40% of the responses), followed by the ability to evaluate students regarding different metrics (30%).

Results from the educators perspective

- Current features usefulness and ease-of-use: Educators think that all features are useful without any exception. However, they showed a special interest in the processes, life cycle and projects management systems, the ability to learn through real SE processes, and the ability to supervise students performing tasks in one shared space. See table 4. Educators are also unanimous about the ease-of-use of the current features, see Table 5.

Table 4: Educators appreciation about current features usefulness

Current features usefulness	**Manda-tory** %	**Very Useful** %	**Useful** %	**Moder-ately useful** %	**Not useful at all** %	**Need Im-prove-ments** %
Processes management system	16.66	33.33	50.00	00.00	00.00	00.00
Adding resources to processes	00.00	50.00	50.00	00.00	00.00	00.00
Projects creation and management	16.66	50.00	33.33	00.00	00.00	00.00
Life-cycle model definition	00.00	50.00	50.00	00.00	00.00	00.00
Notification management system	00.00	00.00	83.33	16.66	00.00	00.00
Enabling the use of SE processes	00.00	50.00	33.33	16.66	00.00	00.00
Reviewing the 3D design	00.00	16.66	66.66	16.66	00.00	00.00
Supervising one shared space	16.66	33.33	50.00	00.00	00.00	00.00
Enabling distributed engineering	00.00	33.33	66.66	00.00	00.00	00.00

Table 5: Educators appreciation of curent features ease-of-use

Current features ease-of-use	**Very simple** %	**Simple** %	**Pretty hard** %	**Very hard** %
Processes management system	16.66	50	33.33	00.00
Adding related resources to processes	16.66	50.00	33.33	00.00
Projects creation and management	16.66	83.33	00.00	00.00
Life-cycle model definition	16.66	50	33.33	00.00
Notification management system	16.66	50.00	33.33	00.00
Reviewing the 3D design of the system	33.33	33.33	33.33	00.00
Supervising one shared space	33.33	33.33	33.33	00.00
Enabling distributed engineering	66.66	16.66	16.66	00.00

- Additional features usefulness: Regarding the proposed additional features, educators mostly agreed on their utility. According to their responses, the most important extentions are assisting students throughout the execution of SE processes, the ability to consider and engineer subsystems as a system, and students managing their tasks.

Table 6: Educators appreciation of additional features usefulness

Additional features usefulness	**Mandatory** %	**Useful** %	**Not Useful** %
Chat system	00.00	83.33	16.66
Video-call system	00.00	83.33	16.66
Assisting student through annotations	33.33	66.66	00.00
Engineering at both system and subsystems level	33.33	66.66	00.00
Tasks management	16.66	83.33	00.00
Direct web-access to SE tools	33.33	50.00	16.66
Direct web-access to learning resources and platforms	33.33	33.33	33.33
Downloading a project synthesis, at any moment	16.66	50.00	33.33

- Students evaluation methods: Educators also beleive that students should be evaluated in the context of learning SE, using different methods and metrics at the same time. The most expected method by educators is the ability to assess students throughout the project execution process (the engineering of the system), recommended by about 83% of them. This mainly represents the evaluation of SE processes execution quality. The second method favored by about 67% is student evaluation regarding the acquired knowledge and skills, by using surveys at the end of the project, but also by extracting useful information from the learning process. The third method recommended by 50% of educators, was self-evaluation and evaluation of the final results in order to see if they match with the starting requirements. Just as students, only 33% of educators approved of peer evaluation.
- The advantages of this solution compared to their traditional way to teach SE: Unlike students, educators did not think that the ease-of-use and the project-based learning approach were the most important aspect that differentiate this solution from traditional SE teaching. However, 50% of educators appreciate the use of SE standard processes and the ability to evaluate students using different metrics. More importantly, about 67% declare that the ability to manage the systems life-cycle model is a good idea, along with the ability to manage geo-distributed students.

Conclusion

In this paper, we presented a solution for SE education, using international standards in a project-based-learning approach. Thanks to this concept, students will learn to not only engineer the requested system, but also to engineer it the right way, using real-life SE practices conveyed by standardized processes, together with communication, team management, collaboration and related soft skills. The main advantages of our solution are the processes, life-cycle, and projects adaptation and management components, as well as the shared workspace for students engineering tasks during all the life cycle. Another advantage of the solution resides in its ability to help in meeting the challenges of a project-based-learning approach, in particular by opening a way of assessing students by different metrics, including: the final results, the execution quality, the acquired knowledge, and the acquired skills. In addition, the conducted survey highlights that both educators and students appreciate the usefulness and the ease of use of the current features of the solution. They also approved the proposed additional features, except for the chat and video-call systems.

These additional features will soon be added to the solution. After their implementation, we intend to conduct two new experimentations. The first one will consist of a survey targeting a larger amount of potential users. The second one will be the application of the solution to actual SE teaching. For this purpose, we aim to propose a SE course for a large group of students. Students will first assist to theoretical lectures on SE, and will then be asked to engineer a system in small groups. Half of these groups will be using our solution, and the other groups will work in a traditional fashion. At the end of the course, their results will be compared, with a special focus on the quality of the final product and its conformance to the project requirements, the acquired knowledge and skills, and the quality of SE processes execution.

Acknowledgment

This work is done under PACIS project, a project driven by SUPMECA Paris mechanical engineering school, in partnership with two other engineering schools EISTI and ENSEA. The project is funded by the French National Agency of Research-ANR-. Authors would like to thank Mr. Ismail Mansour, for the web development tasks of this solution and Mr. Sylvain Cerny for the 3D virtual environment development.

References

[1] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer, “Engineering design thinking, teaching, and learning,” *Journal of Engineering Education*, vol. 94, no. 1, pp. 103–120, 2005.

[2] D. D. Walden and G. J. Roedler, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. no: Wiley, 4 ed., 2015.

[3] C. S. Wasson, “Ac 2012-3389: System engineering competency: The missing course in engineering education,” *INCOSE International Symposium Journal*, vol. 20, no. 1, p. 21–36, 2010.

[4] G.-S. Chang, H.-L. Perng, and J.-N. Juang, “A review of systems engineering standards and processes,” *Journal of Biomechatronics Engineering*, pp. 71–85, 2008.

[5] “Systems and software engineering – lifecycle profiles for very small entities (vses) – part 5-6-1: Systems engineering – management and engineering guide: Generic profile group: Entry profile,” *ISO/IEC TR 29110-5-6-1:2015*, pp. 1–50, Jan 2015.

[6] BKCASE, “Guide to the systems engineering body of knowledge (sebok).” `http://sebokwiki.org/wiki/`. [Online;Accessed: 07-February-2017].

[7] W. Clifford, W. Corina, and K. Rabia, “Development of a system engineering competency career model: verification of the model using opm method.” `https://calhoun.nps.edu/handle/10945/47492`. [Online;Accessed: 07-February-2017].

[8] INCOSE, “Incose systems engineering competency framework, international council on systems engineering (incose) united kingdom (uk).” `http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_6.aspx?CatID=Publications`. [Online;Accessed: 07-February-2017].

[9] DAU, “Defense acquisition university (dau) eng competency model.” `http://www.hci.mil/docs/compentency_assessment/ENG_Competency_Model_20130612_Final.pdf`. [Online;Accessed: 07-February-2017].

[10] NASA, “Nasa project management and systems engineering competency model, academy of program / project engineering leadership (appel).” `https://appel.nasa.gov/competency-model/`. [Online;Accessed: 07-February-2017].

[11] “Mitre systems engineering competency model,” in *The Mitre Institute*, (Bedford, MA), Mitre Corporation, September 2007.

[12] INCOSE, “multi-level professional sep certification program.” `http://www.incose.org/certification`. [Online;Accessed: 07-February-2017].

[13] “Iso/iec/ieee international standard - systems and software engineering – system life cycle processes,” *ISO/IEC/IEEE 15288 First edition 2015-05-15*, pp. 1–118, May 2015.

[14] K. Kinda, B. Shadi, H. G. WESLEY, and A. Radaideh, “Engineering design education: Where, when and how,” *Journal of American Education*, 2013.

[15] P. John, W. Dennis, and W. Erica, “Systems & Design Thinking: A Conceptual Framework for Their Integration,” in *Proceedings of the 55th Annual Meeting of the ISSS-2011, Hull, UK*, vol. 55, 2011.

[16] M. Bougaa, S. Bornhofen, and A. Rivière, “A survey of current practice in systems engineering education,” in *44th SEFI Annual Conference*, (Tampere, Finland), September 2016.

[17] K. Craig and P. Voglewede, “Multidisciplinary engineering systems graduate education: Master of engineering in mechatronics,” in *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, 2010 IEEE*, pp. 1–14, April 2010.

[18] S. J. Seymour and R. R. Luman, “Academic perspectives of systems engineering,” *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 29, no. 4, 2011.

[19] “Placis: Collaborative platform for systems engineering.” `http://www.supmeca.fr/fr/international/placis-plateforme-collaborative-d-ingenierie-systemes`. [Online;Accessed: 01-February-2017].

[20] J.-C. Tucoulou, D. Gouyon, and E. Bonjour, “A student challenge in systems engineering: RobAFIS 2010,” *INSIGHT - International Council on Systems Engineering (INCOSE)*, vol. 14, pp. 9–11, Dec. 2011.

[21] G. M. Bonnema, I. F. Lutters-Weustink, and F. J. van Houten, “Introducing systems engineering to industrial design engineering students with hands-on experience,” in *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on*, pp. 408–413, Aug 2005.

[22] M. K. Yurtseven, “Teaching systems thinking to industrial engineering students.”

[23] W. Bauer, W. Biedermann, B. Helms, and M. Maurer, “A student laboratory for systems engineering: Teaching systems engineering to students without previous se-knowledeg based on an industry-oriented example,” in *Systems Conference (SysCon), 2012 IEEE International*, pp. 1–6, March 2012.

[24] C. B. Nielsen and P. Adams, “Active learning via lego mindstorms in systems engineering education,” in *Systems Engineering (ISSE), 2015 IEEE International Symposium on*, pp. 489–495, Sept 2015.

[25] L. E. Whitman and T. L. Witherspoon, “Using legos to interest high school students and imtrove k12 stem education,” in *Frontiers in Education, 2003. FIE 2003 33rd Annual*, vol. 2, Nov 2003.

[26] M. Beth, S. Jason, L. Susan, and L. Peiyi, “Underwater lego robotics as the vehicle to engage students in stem: The build it project’s first year of classroom implementation,” *American Society for Engineering Education Mid-Atlantic, Hoboken, NJ*, 2008.

[27] D. B. Williams, “A freshman ’introduction to ece design’ course based on the lego mindstorms nxt,” in *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pp. 435–439, Jan 2009.

[28] S. Patel, D. Schaefer, and D. Schrage, “A pedagogical model to educate tomorrow?s engineers through a cloud-based design and manufacturing infrastructure ? motivation, infrastructure, pedagogy, and applications,” in *ASEE 2013 Annual Conference and Exposition*, 2013.

[29] M. C. Brannon, Z. Szajnfarber, and T. A. Mazzuchi, “Introducing the fundamentals of systems engineering to freshman through various interactive group activities,” in *2014 ASEE Annual Conference*, (Indianapolis, Indiana), ASEE Conferences, June 2014. https://peer.asee.org/20705.

[30] INCOSE, “Worldwide directory of se and ie academic programs.” `http://www.incose.org/AboutSE/SEEducation/SEProgramDirectory`. [Online;Accessed: 07-February-2017].

[31] M. Bougaa, S. Bornhofen, R. V. O’Connor, and A. Rivière, “Standard-based adaptive path to teach systems engineering: 15288 and 29110 use cases,” in *Accepted-for-Publication at the 11th Annual IEEE International Systems Conference*, (Montreal, Canada), IEEE, April 2017.

[32] “Video tutorial: A process-centred and project-based learning solution for teaching systems engineering.” `https://www.youtube.com/watch?v=ANy67Jxy3to&feature=youtu.be`. [Online;Accessed: 01-February-2017].

Appendices

A Appendix A: An illustration of some features from the proposed solution

- Educator: Processes activities and tasks definition

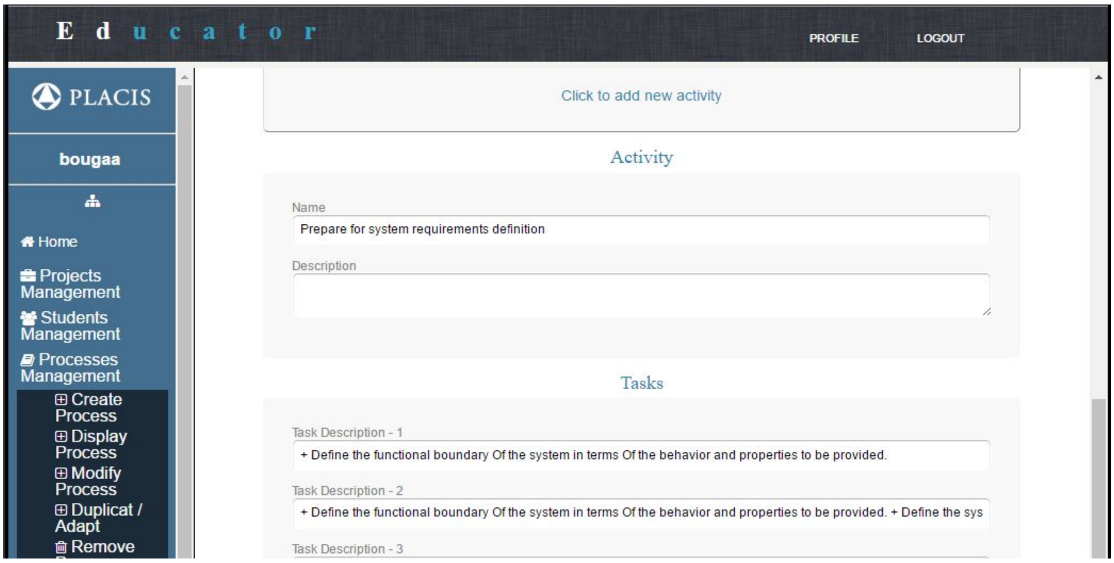

Figure A.1: Adding activities and their related tasks to a specific process

- Educator: New project definition

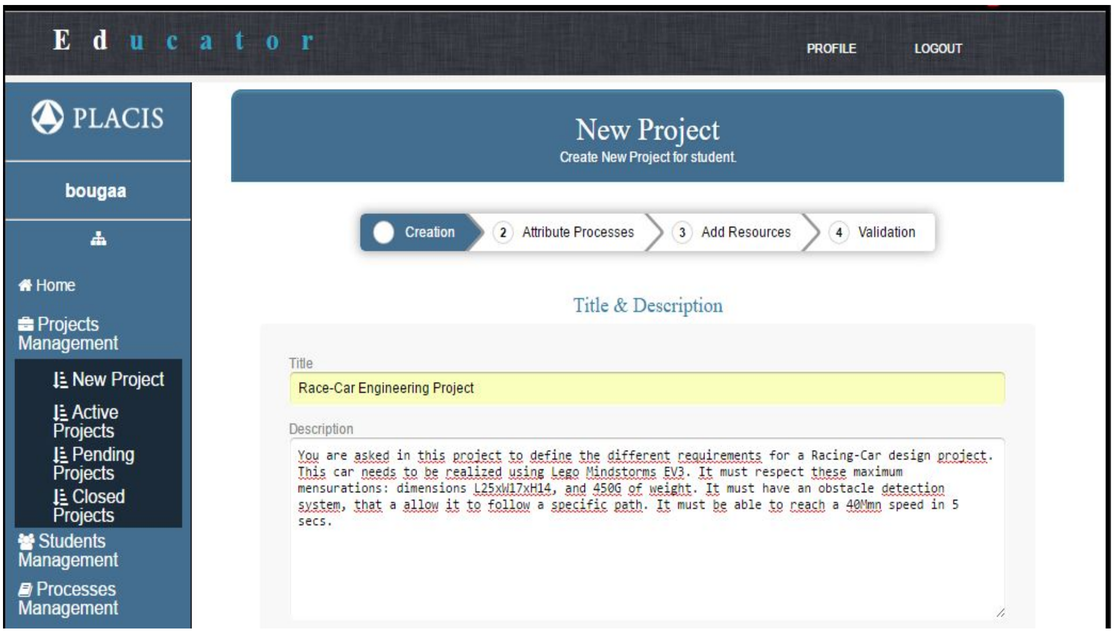

Figure A.2: Adding a title and a description of a new project

- Educator: Life cycle model definition (Processes attribution)

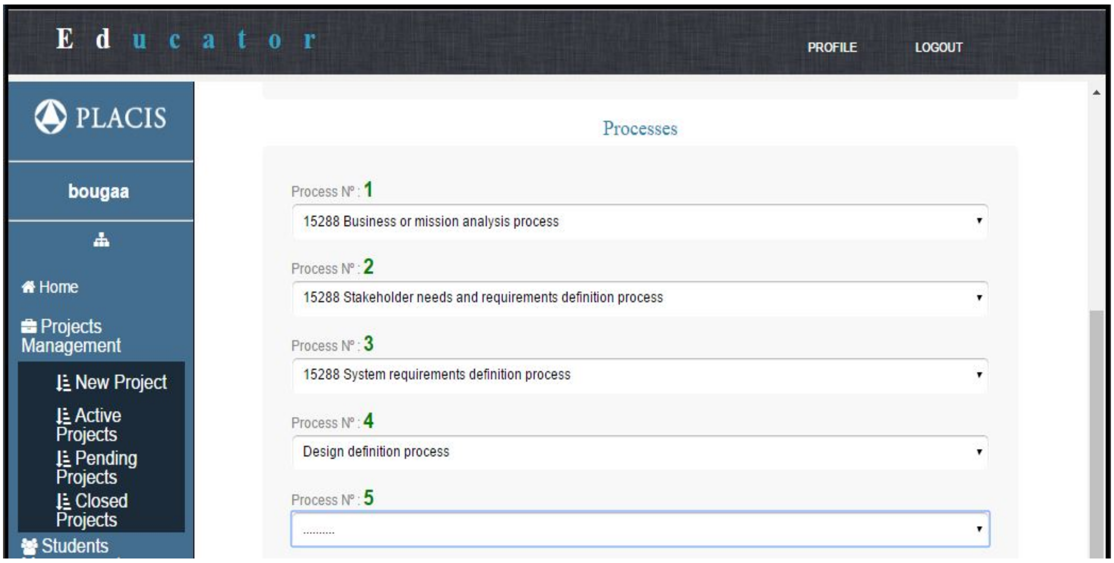

Figure A.3: Defining the adequate Life-Cycle Model for this project

- Student: A project workspace for a specific team

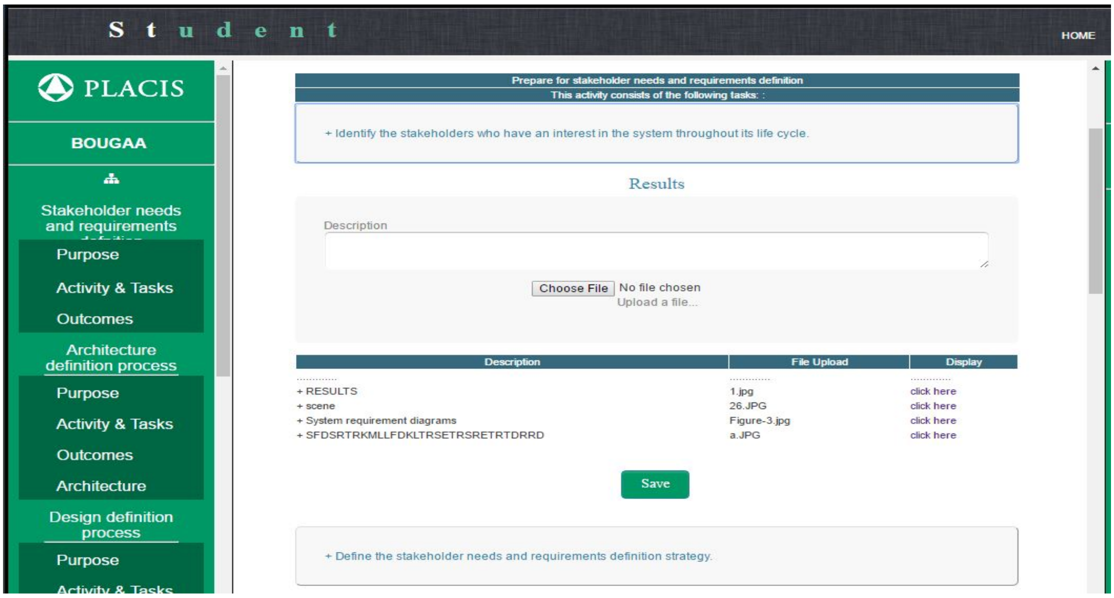

Figure A.4: A task execution results upload, relative to the first activity of the *Stakeholders needs and requirements definition* process

- Student: Architecture definition process

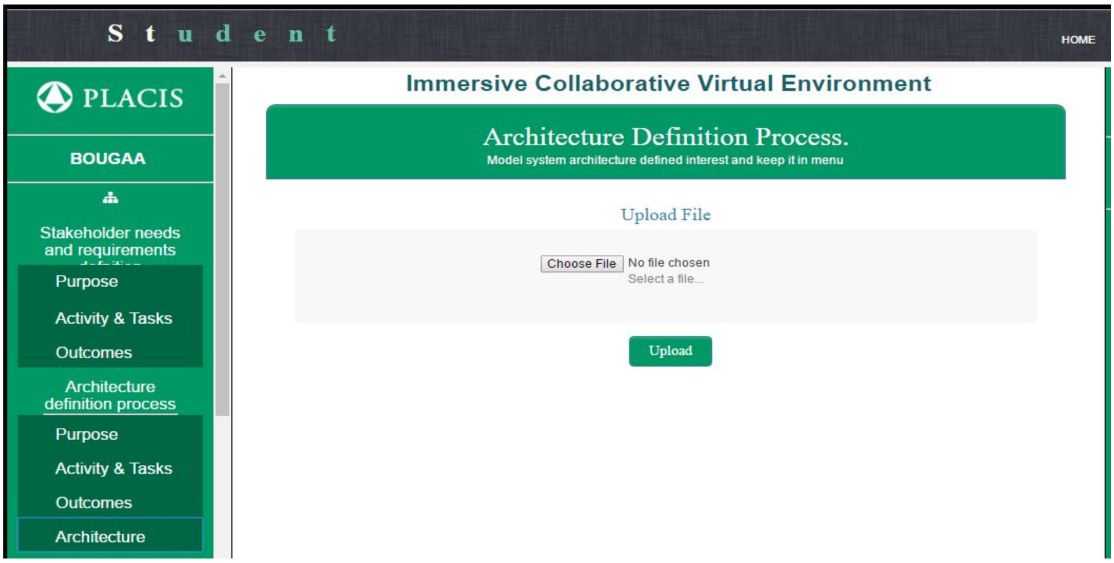

Figure A.5: System-of-interest architecture upload, as an XML file

- Student: 3D Design review and assembly feature, only supporting Lego Bricks for now.

Figure A.6: a collaborative virtual 3D environment for virtual design review and assembly

B Appendix B: Educators Questionnaire

Evaluation of a new solution for Systems Engineering Education

*Obligatoire

1. **Adresse e-mail** *


http://youtube.com/watch?v=ANy67Jxy3to

Some information about you

These information will only be used for statistical goals, and will be treated anonymously.

2. **You are:** *

Une seule réponse possible.

- ( ) Industrial Systems Engineering Practicioner
- ( ) Academic Systems Engineering Practicioner
- ( ) Both Industrial and Academic Systems Engineering Practicioner
- ( ) Autre :

3. **How much experience you have in Systems Engineering field** *

Une seule réponse possible.

- ( ) Less than 1 year
- ( ) 2 to 5 years
- ( ) More than 5 years

4. **How you used to transfer Systems Engineering knowledge and skills** *

Plusieurs réponses possibles.

- [ ] I dont teach Systems Engineering
- [ ] Systems Engineering university lectures
- [ ] Systems Engineering Books
- [ ] Using MOOCs (Massive Open Online Courses, ex.Coursera, Edx...etc)
- [ ] E-learning plateforms (Other than MOOCs)
- [ ] Academic Project Based learning
- [ ] Industry-Academy Project Based learning
- [ ] Competitions and challenges organization
- [ ] Autre :

5. **What are the topics/processes you deal with the most in Systems Engineering** *

Une seule réponse possible par ligne.

	All the time	Occasionally	Rarely	Never
System & Stakeholder Requirements	( )	( )	( )	( )
Architecturing	( )	( )	( )	( )
Design, analysis and implementation	( )	( )	( )	( )
Integration	( )	( )	( )	( )
Test, verification and Validation	( )	( )	( )	( )
Operation and Maintenance	( )	( )	( )	( )
Technical management (project planning & assessment, decision & risk management ...etc)	( )	( )	( )	( )
Acquisition and Supply	( )	( )	( )	( )
Project-Enabling Management (Life Cycle Model, Infrastructure, Portfolio, Knowledge...etc)	( )	( )	( )	( )

What do you think about the actual features of the proposed solution

6. **Please give your opinion about usefulness of these features** *
Une seule réponse possible par ligne.

	Obligatory	Very Useful	Useful	Moderately Useful	Not useful at all	Need Improvements
Processes Management System						
Adding related resources to processes						
Projects Creation and Management						
Life Cycle Model Definition						
Notification Management System						
Enabling learning by executing the real Systems Engineering Processes						
Reviewing the 3D Design of the System						
Supervising all tasks results in the same shared space						
Enabling Geodestributed students to work together						

7. **Please give your opinion regarding the ease-of-use (the ergonomy) of these features** *
Une seule réponse possible par ligne.

	Very simple to use	simple to use	Pretty hard to use	Very hard to use
Processes Management System				
Adding related resources to processes				
Projects Creation and Management				
Life Cycle Model Definition				
Notification Management System				
Reviewing the 3D Design of the System				
Supervising all tasks results in the same shared space				
Enabling Geodestributed students to work together				

What do you think about the following additional features:

These are some additional features we think useful to improve this solution, please tell us what do you think

8. **Enabling students in the same team to communicate in real time, by adding a chat system** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

9. **Enabling students in the same team to communicate in real time, by adding a video-calling system** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

10. **Allowing educators to assist students by making notes about the tasks being realized durring their execution.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

11. **Allowing students to Manage and execute the processes of the Life Cycle Model on the full system, but also on its sub-systems and systems elements. In other terms, adding a feature that allows students to switch between the different sub-systems and systems elements.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

12. **What about adding a features that enables students to manage and assign the different tasks to the different students of the team.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre :

13. **Would it be better if it's for the Educator himself to manage and assign the different tasks to the different students of the team.** *
Une seule réponse possible.

- Yes
- No
- Autre : ____________

14. **What about adding a feature that enables students getting direct web-based access to systems engineering tools (such as SysML)** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ____________

15. **What about adding a feature that enables students getting direct web-based access to documentation and learning plateforms such as EDX.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ____________

16. **What about giving the the Educator the ability to download at any time, a synthesis of the project as a Pdf File.** *
Une seule réponse possible.

- Necessary
- Useful
- Not Useful
- Autre : ____________

17. **What do you think will be the best way to evaluate the work done under this solution?** *
Plusieurs réponses possibles.

- [ ] Self Evaluation (By taking a survey)
- [ ] Peer Evaluation (Evaluation of the results by other students)
- [ ] Educator evaluation regarding the final Results
- [ ] Educator evaluation throughout the project execution
- [ ] Educator evaluation regarding the aquired skills and Knowledge (using questionnaires and projects execution data)
- [ ] A mixture of some of the previous methods (please check the ones you think are the best)
- [ ] Autre : ____________

Comparing this solution

18. **In which way do you think this solution is BETTER than the solutions you used to know or to teach systems engineering with (especially after adding the additional features).** *
Plusieurs réponses possibles.

- [ ] The ease of use of the solution
- [ ] The management of Geodistributed students
- [ ] The management of a Project Based Learning approach
- [ ] The use of Systems Engineering Standard Processes
- [ ] The ability to manage the Life Cycle Model
- [ ] The ability to use any Systems Engineering Standard (Standard choice independant)
- [ ] The ability to use any Systems Engineering Tools (Tools choice independant)
- [ ] The ablity to evaluate students regarding different Metrics
- [ ] Autre : ____________

19. **Do you have any additional remarks or suggestions?**

____
____
____
____
____

Une copie de vos réponses sera envoyée par e-mail à l'adresse indiquée.

Third Paper: (SYSCON) A Standard Based Adaptive Path to Teach Systems Engineering: 15288 and 29110 Standards Use Cases

- **Paper title:** A Standard Based Adaptive Path to Teach Systems Engineering: 15288 and 29110 Standards Use Cases
- **Authors:** Mohammed Bougaa, Stefan Bornhofen, Rory V. O'CONNOR, Alain Rivière.
- **Conference:** 11th IEEE International Systems Conference 2017.
- **Location:** Montreal, Canada.
- **Date:** April 24-27, 2017
- **Keywords:** : Systems Engineering , Systems Engineering Education, Systems Engineering Standards, Project-Based Learning, ISO/IEC/IEEE 15288 Standard, ISO/IEC 29110 Standard, Life Cycle Model.

A Standard Based Adaptive Path to Teach Systems Engineering:
15288 and 29110 Standards Use Cases

Mohammed BOUGAA
Quartz Laboratory EA7393
Eisti Cergy, CentraleSupelec
Cergy, France
Email: mba@eisti.eu

Stefan BORNHOFEN
Quartz Laboratory EA7393
Eisti Cergy
Cergy, France
Email: sb@eisti.eu

Rory V. O'CONNOR
Dublin City University
Dublin, Irlande
Email: rory.oconnor@dcu.ie

Alain RIVIERE
Quartz Laboratory EA7393
SupMeca Paris
Saint-Ouen, France
alain.riviere@supmeca.fr

***Abstract*—This paper discusses the use of two different standards for teaching Systems Engineering (SE): ISO/IEC/IEEE 15288 and ISO/IEC 29110. The first one is a general and widely-used standard describing the lifecycle processes of the entire system, whereas the second one is a relatively new standard based on a reduced set of standards elements focused on lifecycle profiles for Very Small Entities (VSEs). We are especially interested in the impact that SE standards can have on teaching this discipline to engineering students. We consider the teaching of fundamental principles of systems engineering. In this paper we illustrate how our, previously developed, standard based solution for systems engineering education can be used as a framework to support these standard-based teaching paths. We mainly focus on illustrating how adapting standard processes can be done, considering not only the learning goals, but also projects size and complexity, in a project-based learning environment.**

This paper shows that, thanks to it's adaptation from the ISO/IEC/IEEE 15288, and to it's reduced size, the ISO/IEC 29110 standard is particularly suitable for teaching systems engineering fundamental knowledge to undergraduate students, new to the discipline. While the ISO IEC/IEEE 15288 might be more suited for students that already have a good grounding in systems engineering fundamentals, especially thanks to the ability to use some from its various processes to separately teach different topics of systems engineering.

Keywords: Systems Engineering , Systems Engineering Education, Systems Engineering Standards, Project-Based Learning, ISO/IEC/IEEE 15288 Standard, ISO/IEC 29110 Standard, Life Cycle Model.

I. Introduction

Systems Engineering (SE) is a structured approach focusing on the design and the management of complex engineering projects over their entire life cycle. It is presented by the International Council on Systems Engineering (INCOSE), as *"an interdisciplinary approach and means to enable the realization of successful systems. It focuses, not only on defining customer needs and required functions early in the development cycle, but also, documenting requirements, proceeding with design synthesis and system validation, and considering the complete problem are in the scope of Systems Engineering"* [1]. This discipline has known much efforts trying to promote its adoption by different industries, especially when the complexity of nowadays systems has increased to an unprecedented level. This complexity is caused at the first place, by the multidisciplinary aspects of modern systems along with, their multiple involved stakeholders, and their geo-located engineering context. These efforts are primarily, development of multiple international standards, guides and methodologies, to help the industry adopting this approach [2]. However, many academic institutions have not coped with systems engineering challenges, and have not sufficiently considered its specifications and requirements when implementing their engineering curriculum. So, while the engineering industries may better cope with the complexity problems, thanks to the different efforts done in this domain, they face a significant lack of well-trained human resources who master the fundamental and domain-specific systems engineering principles and their corresponding standard processes. Standards and their related processes and methods are one of the most sought after skills, but academic institution, as stated in our review study [3], have not kept up with this demand. From an industry perspective, in order to be effective as systems engineers, in addition to the necessary knowledge in their traditional engineering disciplines, engineering students need practical and real world experiences, acquired in reality-like geo-distributed and multidisciplinary context. It needs to be a skills focusing and challenging context[4], more than being a knowledge focusing one, what presents real challenges from academic perspective. To cope with these challenges, we proposed in [5] a new technology-based solution to teach systems engineering.

Our proposed solution, presented in section (IV), enables educators to work on an adaptive path for teaching this discipline, by giving them the ability to adapt their learning scenarios. Educators are able to select and adapt standard processes to make them more suited for each learning situation. We illustrate two use cases using the 15288 and the 29110 standards in section (V) of this paper. In section (II) and (III) we respectively, highlight the specifications and requirements of systems engineering education and the leading standards in this discipline, with a focus on the previously cited ones.

II. Systems Engineering Education

According to Muller [6], systems engineering education differs from traditional mono-disciplinary engineering courses, since the training needs to focus more on skills and less on transferable facts. The author gave a set of recommendations to consider for a good systems engineering education program, including interaction with students, soft skills development, media use and students feedback. In the same context, Dym [7] believes that "*a good engineering education is about process, about learning how to think like an engineer; its much more than a prescription of content*".

Dym et al. [8] recommend the following three activities for a powerful learning environment for systems engineering and similar disciplines:

- *Instrumenting the learning process to obtain quantitative and qualitative data that support metrics consistent with quality control.*
- *Teaching design engineering and other disciplines such as systems engineering across geographically dispersed, culturally diverse, international networks*
- *Engage design coaches to help manage the contextualization of engineering design theory and practice.*

Finally, in a broader context, Herrington and Kervin [9] specify nine main characteristics that any learning environment, technology-based or not, should feature:

- *Provide authentic context that reflects the way the knowledge will be used in real life*
- *Provide authentic activities*
- *Provide access to expert performances and the modeling of processes*
- *Provide multiple roles and perspectives*
- *Support collaborative construction of knowledge*
- *Promote reflection to enable abstractions to be formed*
- *Promote articulation to enable tacit knowledge to be made explicit*
- *Provide coaching by the teacher at critical times, and scaffolding and fading of teacher support*
- *Provide for authentic, integrated assessment of learning within the tasks*

It appears clearly, from the work we presented in [3], that most of systems engineering current practices don't take these considerations and recommendations into account when designing their curriculum, and none of them considered the use of SE standards during the learning scenario. However, this survey [3] helped us defining the perimeter and the features of our solution, that will be presented later. This helps us especially deciding to focus our efforts in incorporating systems engineering standards in the learning scenarios, and to select the Project-Based Learning (PBL) as a pedagogical model.

III. Systems Engineering Leading Standards

A. SE Standards Overview

Systems engineering addresses the complexity of systems, in order to be able to transfer user needs into operational systems via an interdisciplinary processes. The early standard for systems engineering was the US Military Standard MIL-STD-499, Engineering management from 1969 [2], produced by the US Department of Defense (DoD) for the defense industry. It has been adapted twice after that, the MIL-STD-499A release on May 1st , 1974, and the MIL-STD-499B draft on 1992. By 1994, a group of organizations called Electronic Industries Alliance (EIA) collaborate to develop a commercial systems engineering standard to replace the military one. This group included representatives from the DoD, the Aircraft Industry Association (AIA), the National Security Industries Association (NDIA), the Institute of Electrical and Electronics Engineers (IEEE), and INCOSE [2]. By December 1994, they released the EIA Interim Standard 632 (EIA/IS 632) Systems Engineering. This Standard became later the ANSI/EIA 632-1998, Processes for Engineering a System [10], which has been approved on January 7th, 1999.

The IEEE , the International Organization for Standardization (ISO) along with the International Electrotechnical Commission (IEC), have also worked on developing systems engineering standards. By 1998, and after a trial-use version by 1995 (IEEE Std 1220-1994) [11], IEEE produced the IEEE Std 1220-1998, standard for application and management of the systems engineering process [12], and by 2002, ISO and IEC released the ISO/IEC 15288 standard, systems engineering-system life-cycle processes [13], which has been created by the same group that created the ISO/IEC 12207 software life-cycle standard, in collaboration with systems engineering experts [2].

Each of these three different Commercial standards, EIA 632, IEEE 1220 and the ISO/IEC 15288 addressed various level in the systems engineering processes. While the last active version of the EIA 632, processes for engineering a system, still the one approved on 1999 and reaffirmed on 2003, and the IEEE 1220 that has been revised once on 2005, which still be the actual active version of this standard [14], the ISO/IEC 15288-2002 after its adoption by IEEE in 2004, has been revised by the ISO/IEC 15288:2008 [15], before it has been canceled and replaced by its final revision, the ISO/IEC/IEEE 15288:2015 [16] which was prepared by Joint Technical Committee ISO/IEC JTC 1 (Information technology), Subcommittee SC 7 Software and systems Engineering), in cooperation with the IEEE computer society systems and software engineering standards committee [Ref152882015standard]. In 2004, ISO/IEC 15288-2002 has been adopted by IEEE, in the IEEE Std 15288-2004 [17], Also, INCOSE adopted this standard and aligns upon it, the process and life cycle content in their 4th Version of the SE Handbook [18]. ISO/IEC/IEEE 15288, by gathering all the important normalization institutions and big industrial around it, is becoming the most revised and the most complete standard for systems engineering. It's becoming the starting point of many derivative products, different institutions and researchers, in order to support it, are producing content, guidance, reports, use cases...etc. Others are creating completely new products, such as the ISO/IEC 29110 standard [19] , which is a Systems and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs). ISO/IEC 29110 is mainly based on ISO/IEC/IEEE 15288 for systems engineering part, and on the ISO/IEC/IEEE 12207 for software engineering part. This was just a sample set of standards, to show their importance in the field of systems engineering, for more information about systems engineering standards, especially in some specific disciplines, you can take a look at Incose website [20]

B. Illustrative Standards

Based on the previous overview of the different systems engineering processes standards, and as our goal is to illustrate our adaptive learning path, we consider the use of ISO/IEC/IEEE 15288 and ISO/IEC 29110 as the best choice. Especially because one of these standards covers the entire life cycle of a system, with a large number of different individual processes, while the other one, is a smaller set of processes focusing on the need of VSEs, some of their advantages are listed here.

*1) **The ISO/IEC/IEEE 15288 Standard**:* The most significant characteristics of the ISO/IEC/IEEE 15288 are:

- It is the systems engineering reference standard, and it is promoted by mainly all the standardization organizations including IEEE and INCOSE. It is up-to-date and based on proven practices.
- Our aim is to let the students learn the fundamentals of systems engineering, but we don't want to bother them with high level of details relative to deeper processes application. However we want educators to be able to select most suited topics to learn for specific students, and for which kind of systems. This standard provides them with the ability to do that.
- Its processes can be applied in different manners: concurrently, iteratively and recursively to a system, and incrementally to its elements. It can be applied to a an element of a system, considered as a system itself, as it can be applied at any level in the hierarchy of a system across its life cycle.
- It applies to man-made systems configured with one or more of the following, hardware, software, humans, or processes.
- When defining the life cycle model and its different stages, educators can choose which of this standard processes to consider in order to be in conformance with. But also, its processes can be tailored to fit a specific learning goal for example.
- It can be used alone or jointly with the ISO/IEC/IEEE 12207, for software engineering, which has the same terminology and concepts.

In its last revision of 2015, it includes 30 processes grouped into four categories.

- Agreement processes: 2 processes
- Organizational Project-Enabling Processes: 6 processes
- Technical management processes: 8 processes
- Technical processes: 14 processes

For each process, this standard provides us by:

- Its Purpose: a paragraph that describes at a high level overall goal of performing the process.
- Its Outcomes: Outcomes express the observable results expected from the successful execution of the process.
- Its Activities : Activities provide the first level of structural decomposition of a process, they generally provide a set of the related lower-level elements called Tasks.
- Its activities tasks: Tasks are requirements, recommendations, or permissible actions intended to support the achievement of the outcomes.
- and some Notes:

This standard provides a common processes framework for describing the life cycle of systems created by humans, adopting a systems engineering approach[16]. It does not describe a specific system life cycle model, neither a development methodology, method, model or technique.

*2) **The ISO/IEC 29110 Standard**:* The most significant characteristics/advantages of this standard and its differentiation regarding the first one are:

- The recently published set of ISO/IEC 29110 international standards (IS) and Technical Reports (TR) [21] are specifically aimed at addressing the specific needs of VSEs (Very Small Entities), i.e., enterprises, organizations, departments or projects with up to 25 people.
- The engineering standards and guides developed by an ISO working group, Working Group 24 (WG24), are targeting VSEs which do not have experience or expertise in selecting, for a specific project, the appropriate processes from life cycle standards such as ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288, and tailor them to the needs of a specific project [22].
- Building upon the success of ISO/IEC 29110 for software, in 2009, an INCOSE working group was established to evaluate the possibility of developing a standard using the generic profile group scheme of the ISO/IEC 29110 series and the systems engineering life cycle processes standard ISO/IEC/IEEE 15288 (2008), for organizations developing systems instead of just softwares. This new ISO/IEC 29110 standard is targeted for VSEs that do not have experience or expertise in tailoring ISO/IEC/IEEE 15288 to their needs. The result is the publication of a Systems Management and Engineering Guide Entry profile (ISO/IEC TR 29110-6-5-1:2015), i.e. for VSEs working on small projects (e.g. at most six person-months effort) and for start-up VSEs and Basic Profile (ISO/IEC TR 29110-6-5-2:2014) [23].

- The systems engineering basic profile is composed of two processes: Project Management (PM) and System Definition and Realization (SR). An acquirer provides a Statement of Work (SoW) as an input to the PM process and receives a product as a result of SR process execution, see Figure 1.

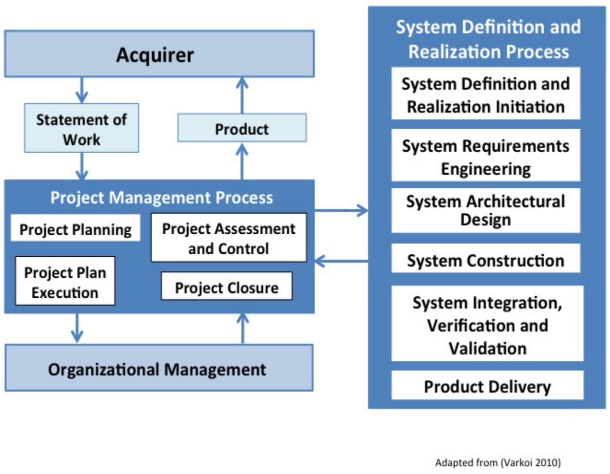

Fig. 1. ISO/IEC 29110-6-5-2:2014 architecture

IV. Solution Description

In this section we describe our solution and highlight its main features, more details about the full solution can be found in [5]. At this stage, the developed solution only allows us to work with one kind of processes: the technical processes. These processes allow users, students in this case, to engineer their systems, without dealing with other kinds of processes, such as the management, agreement, and project-enabling processes.

One of our main goals is to help systems engineering education organizations improve their SE teaching experience, and as stated before, we are focusing on the teaching of the fundamental principles of systems engineering. This fundamental knowledge, when correctly acquired by students allows them to easily adapt it to meet the specific industries needs in terms of systems engineering skills. For this main reason, we decided not to build this solution upon an existing systems engineering methodologies, which may limit us by imposing some kind of tools, methods, or processes to follow even if they are not fully adapted to the system-of-interest being engineered by students, or simply not fulfilling the learning experience requirements. Rather, our solution is process-centered, hopefully standards ones, while still being independent of the specific standard choice. It provides the learner and educator with the ability to work on the different stages of a life cycle model using any systems engineering standard processes. We illustrate in the next section, two different use cases using two different standards.

From the adopted systems engineering standard, the educator can easily register in the solution the different processes he's interested in, while personalizing them or not. See Figure 2 bellow, for an example of a process registered in the solution. Then, for each new project regarding the engineering or re-engineering of a system, educators are able, through a specific interface, created for that effect, to define the project life cycle model and share it with students. They can choose to make students passing through all the proposed processes in the standard, or just through few of them, and also adapt standard processes depending on what type of conformance with the standard they want to claim for their system-of-interest, but also depending on the project characteristics and the learning goals.

The defined life cycle model, will be then followed by students to engineer the requested system, by performing the different activities and tasks. They will be working on a collaborative project based approach, using the recommended tools, methods, and resources, while producing the expected outputs. An example of students workspace, including the system life cycle at the left, is illustrated in Figure 3 below.

In order to illustrate how different standards, with different adaptations can and should be implemented in this solution for teaching purposes, we only need to use the following three features from our solution:

- **Processes Management System:** allows educators, to register systems engineering processes in the solution. Educators are able to create new processes, or adapt existing ones.
- **Life Cycle Management System:** Here, the Life cycle model is defined by the educator, using the systems engineering processes, defined inside the processes management system.
- **Students Engineering Workspace:**
  This represents the workspace where students work in collaboration mode to engineer the requested system. In short words, it allows students teams as producer of the system-of-interest,to follow the project life cycle model, in order engineer the system. They can, depending on the role of each one, start performing their tasks and activities to get the expected outcomes, and reporting them in the adequate process element in the solution menu, so, everyone from the group, including the educator, can be aware of the progress in the project. This help them, very early detecting if there is a system requirement definition gap, a misconception, a validation problem, or any other problem, and report it.

V. Illustrations

- **Case No.01:** Introduction to requirements engineering using the 15288 Standard
  Learning Goal: The goal of this use case is to make students learn the different aspects of the needs and requirements engineering, through the realization of a project proposed by the educator. Figure 4, show the

Fig. 2. 15288 stakeholder needs and requirements definition process, implemented in our solution

definition of the project "*Requirements for a 3D Racing-Car Design Project*". This will enable them to go from defining the problem and solution spaces, until defining the technical view of the solution that meets operational needs of the user, passing through stakeholder identification with their needs. In shorter words, they should pass through the defined system life-cycle model illustrated in Figure 3 and 4.

Used processes: Using the 15288 standard, students needs to pass at least through three processes: the *Business or mission Analysis Process*, the *Stakeholder Needs and Requirements Definition Process* and the *System Requirements Definition Process*, as shown in Figure 3, that represent students workspace containing the three processes forming the system life-cycle model.

Note that, this is just an illustration use case, we can do the same thing about learning the system design, system verification, validation, and other topics of systems engineering using the 15288 standard processes. The main significant thing that must be considered at this cases, is that educators, need not to only define the project and life cycle, but also to provide the necessary inputs, such as the systems requirements. educators can even make students pass through the entire life cycle technical processes of the 15288 or another standard. Next, we'll show how this can happens using the 29110 standard this time.

- **Case No.02:** Introduction to SE through the entire technical system life cycle model, using 29110 standard

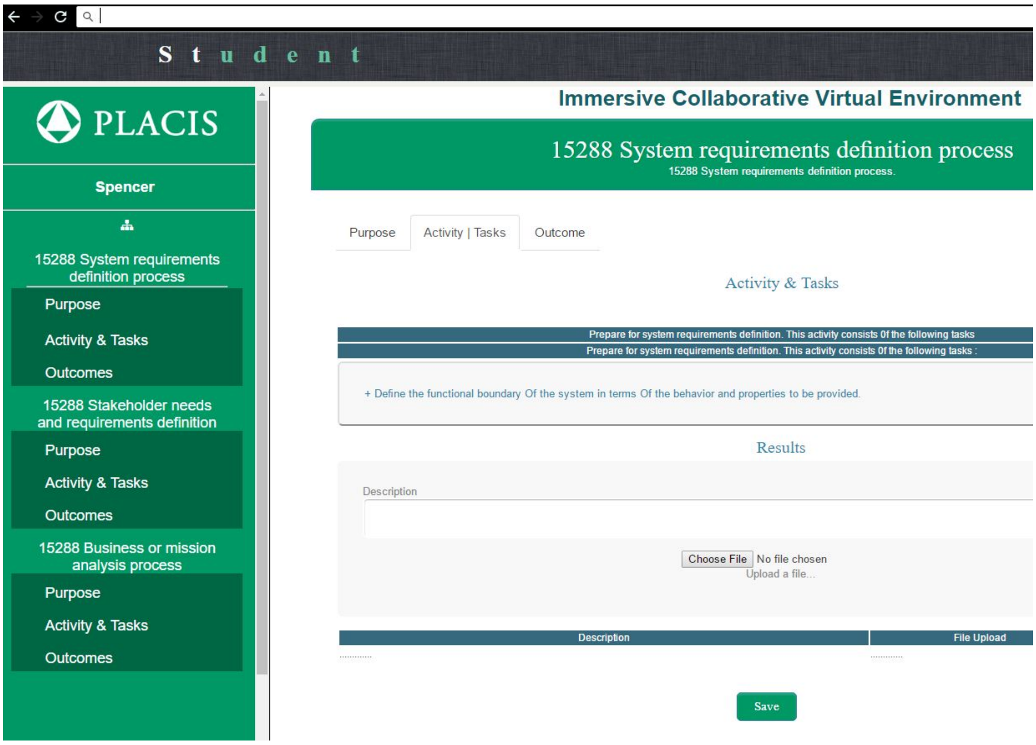

Fig. 3. A example of students workspace, showing the system-of-interest life cycle model at the left, and the different tasks and activities execution space, for a specific process ”15288 System requirements definition process”

Fig. 4. Project description and related system life cycle model

Learning Goal: This time, we are interested in teaching more aspects of systems engineering, those conveyed by all the technical processes of the entire system life cycle processes. We also aim to teach students these aspects using a PBL approach, using the previously defined project, with some adaptations, like the fact that, this time students have to fully engineer the requested Racing-Car, going from system definition to systems delivery.

Used Processes:

This time, we think that using the 29110 standard will be more appropriate, especially for small teams of students, dealing with simple pedagogical systems, and who are new in systems engineering. The educator defines the system life cycle model inside this solution, according to the 29110. More specifically, the system life cycle model is defined based on the generic profile group: Entry profile, from systems engineering management and engineering guide ISO/IEC TR 29110-5-6-1 [24]. It was intended to be used by VSEs to establish processes to be implemented using any development approach or methodology, based on the specific VSE or project needs. What we consider as technical processes, are defined in this guide, as a single process called *System Definition and Realization (SR) process*, see Figure 1. This global process has six activities: SR.1 System Definition and Realization Initiation, SR.2 System Requirements Engineering, SR.3 System Architectural Design, SR.4 System Construction, SR.5 System Integration, Verification and Validation, SR.6 Product Delivery [24]. These activities can be considered at the same level as the 15288 technical processes.

Since the technical processes are considered as activities forming one global process, the SR Process, they will be implemented such as in the platform. So, the system life

Fig. 5. A example of students workspace, showing the system-of-interest life cycle model at the left, and the different tasks and activities execution space, for a specific process ”29110 System definition and realization process”

cycle model in this case, even if it represent the entire technical life cycle model, consists of only the SR Process. This process is defined in the platform by it's name, purpose, and its Seven objectives as outcomes. Then, the different activities, representing each, the equivalent of a technical process in the 15288, are added as activities, and their related tasks as tasks. At this level of maturity of our solution, we don't take the *Roles* described in this guide into consideration. *Input Products* and *Output Products* can be considered as the results students will be uploading in each task space, even if their links with other activities are not managed in the actual solution. At the end, students will have the workspace illustrated in Figure 5.

VI. Conclusion and Further Work

Systems engineering international standards, encompass from one side the fundamental knowledge of systems engineering, and from the other side, they have been used as the main source of competencies used in different systems engineering competency models, including, the Incose, Nasa, and other competency models. We showed in this paper, how systems engineering teaching and learning will be improved by adopting these standards when developing our solution. The greatest impact is especially remarkable regarding the learning outcomes compared to the systems engineering competencies, described in different competency models. The most significant advantage of the proposed solution reside in the fact that educators, starting from which competencies they want students to learn, they are able to tailor different systems engineering standards and their processes to teach them students. In addition, the ability of this solution to get students working together using a technological solution in a PBL-Based approach from different locations, to engineer the same system, by passing through the entire life cycle model defined by the educator, enables students to learn the other part of systems engineering competencies, including the soft skills, the team management, ...etc, which needs to be learned by practice. We highlighted in this paper the ability of this solution to be used with different standards, using different adaptations, depending on the project type and learning goals.

We are still working on improving this solution, by enhancing its outcomes for both students and educators. We are working on a new way to evaluate students, and improving by the way PBL assessment challenges. We are doing that by sharing with educators a vision of all what is happening during the project engineering time, but also by putting some KIs (Key Indicators) in the students work-space, and sending the extracted information to the educators. This will enhance students evaluation regarding: their results, execution quality, along with their acquired Knowledge and Skills, and enable educators to set which systems engineering competencies are learned and what still need to be learned, for each student.

We can go further in the possibilities of adapting the learning path, so it can fit systems engineering wanted skills, such as the communication and time/cost management. We can, for instance imagine a new learning path where different independent groups of students are working on the same project. In this case, group A is responsible of the needs and requirements engineering, group B on the system design and architecture, group C deals with system construction and finally group D with system integration, verification, and validation. At this case, the repartition of students among the different groups should be done based on their systems engineering competencies, and the competencies they had to learn. However, in order to make this really happening, some additional features are needed in this solutions, such as adding new communication channels, to enable teams to communicates together, and making it possible to use standardized management and agreement processes in the solution, in addition to the technical processes. We can also ensure that, in some situations, students will define and manage their system life cycle model by themselves. This can make students get a larger picture of what systems engineering is about, and making them able to speak the systems engineering language, at different levels.

Acknowledgement

This work is done under PLACIS project, driven by Institut Polytechnique du Grand Paris (formed by SUPMECA, EISTI and ENSEA engineering schools). PLACIS project is funded by the French Ministry of Higher Education and Research, under the future investments program with the reference ANR-11-IDFI-0029

References

[1] Cecilia Haskins. *Systems Engineering Handbook, a guide for system life cycle processes and activities*. INCOSE, no, 3.2.2 edition, 2012.

[2] Guey shin Chang, Horng linn Perng, and Jer nan Juang. A review of systems engineering standards and processes. *Journal of Biomechatronics Engineering*, pages 71–85, 2008.

[3] Mohammed Bougaa, Stefan Bornhofen, and Alain Riviere. A survey of current practice in systems engineering education. In *44th SEFI Annual Conference*, Tampere, Finland, September 2016.

[4] T. S. S. Subramanian, M. Mukundlal, P. Prempyari, and P. Dubey. Systems engineering: A new approach to engineering education in india. In *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*, pages 1–3, July 2012.

[5] Mohammed Bougaa, Stefan Bornhofen, Alain Riviere, and Jean-Claude Tucoulou. [in review process] standard processes centered and project based framework for teaching systems engineering. In *124th ASEE Annual Conference*, Columbus Ohio, United States, June 2017.

[6] Gerrit Muller. Didactic recommendations for education in systems engineering, 2010.

[7] Clive L Dym. Design, systems, and engineering education. *International Journal of Engineering Education*, 20(3):305–312, 2004.

[8] C.L. Dym, A.M. Agogino, O. Eris, D.D. Frey, and L.J. Leifer. Engineering design thinking, teaching, and learning. *Journal of Engineering Education*, 94(1):103–120, 2005.

[9] Jan Herrington and Lisa Kervin. Authentic learning supported by technology: Ten suggestions and cases of integration in classrooms. *Educational Media International*, 44(3):219–236, 2007.

[10] A. Ei. *ANSI/EIA-632 Standard, Processes for Engineering a System*. EIA, 1998.

[11] Ieee trial-use standard for application and management of the system engineering process. *IEEE Std 1220-1994*, pages i–, 1995.

[12] Ieee standard for application and management of the systems engineering process. *IEEE Std 1220-1998*, pages i–, 1999.

[13] Iso/iec/ieee international standard – systems engineering – system life cycle processes. *ISO/IEC 15288 First edition 2002-11-01*, pages 1–70, Jan 2002.

[14] Ieee standard for application and management of the systems engineering process. *IEEE Std 1220-2005 (Revision of IEEE Std 1220-1998)*, pages 1–87, 2005.

[15] Iso/iec/ieee systems and software engineering - system life cycle processes. *IEEE STD 15288-2008*, pages c1–84, Jan 2008.

[16] Iso/iec/ieee international standard - systems and software engineering – system life cycle processes. *ISO/IEC/IEEE 15288 First edition 2015-05-15*, pages 1–118, May 2015.

[17] Adoption of iso/iec 15288:2002 systems engineering-system life cycle processes. *IEEE Std 15288-2004 (Adoption of ISO/IEC Std 15288:2002)*, pages 1–67, 2005.

[18] David D. Walden and Garry J. Roedler. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Wiley, no, 4 edition, 2015.

[19] Claude Y Laporte, Ronald Houde, and Joseph Marvin. Systems engineering international standards and support tools for very small enterprises. In *INCOSE International Symposium*, volume 24, pages 551–569, 2014.

[20] Incose Website. Systems engineering standards, 2017. Accessed: 02605-2017.

[21] Rory V. O'Connor and Claude Y. Laporte. The evolution of the iso/iec 29110 set of standards and guides. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 10(1):1–21, 2017.

[22] Rory V. O'Connor and Claude Y. Laporte. Tan innovative approach to the development of an international software process lifecycle standard for very small entities. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 7(1):1–22, 2014.

[23] C. Y. Laporte and R. V. O'Connor. Systems and software engineering standards for very small entities: Accomplishments and overview. *Computer*, 49(8):84–87, Aug 2016.

[24] Systems and software engineering – lifecycle profiles for very small entities (vses) – part 5-6-1: Systems engineering – management and engineering guide: Generic profile group: Entry profile. *ISO/IEC TR 29110-5-6-1:2015*, pages 1–50, Jan 2015.

1st Additional paper: Virtual Reality for Manufacturing Engineering in the Factories of the Future

- **Paper title:** Virtual Reality for Manufacturing Engineering in the Factories of the Future
- **Authors:** Mohammed Bougaa, Stefan Bornhofen, Hubert Kadima, Alain Rivière.
- **Journal:** Applied Mechanics and Materials
- **Volume:** Vols. 789-790, pp 1275-1282
- **Date:** 2015
- **Keywords:** Virtual Reality, Manufacturing Engineering, Systems Engineering, Factoris of the Future.

- **Abstract/Introduction:**

This paper discusses the possibilities of applying Virtual Reality (VR) technologies to Manufacturing Engineering, and in particular assesses its role in the Factory of the Future (FoF). We review, classify and compare the recommendations given by four major European reports on the challenges that have to be met for a successful deployment of the FoF, and we identify the potential contributions of VR to this vision in terms of new technologies, worker-factory relationship, modular infrastructure and production efficiency. We argue that VR can be a key technology to support the FoF at all levels of the Systems Engineering approach, either directly by applying it in standard engineering processes, or indirectly by leveraging other useful technologies.

Introduction

From manufacturing to services companies in military, aerospace, medical or automotive industries, the engineering of physical systems hasgrown more complex than ever. Modern systems often have large architectures and require interdisciplinary competences. They must not only be functional, but also reliable, maintainable and safe. These challenges require new methodologies and have profound consequences on systems modeling, analysis, validation, safety, decision making, and skills learning [1] [2]. Systems Engineering, which appears as a necessary approach in this case, is a "GoalIndependent methodological approach", whose first concern is not a specific design objective, but the optimization of all involved engineering processes. As defined by the International Council on Systems Engineering (INCOSE): "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs". [3] Much research effort has focused on proposing appropriate tools for Systems Engineering, yet not all technologies that might support this approach had full attention of the industrial and academic communities so far, either because they are recentlike Big Data or the Internet of Things, or because, up until recently, they were immature and unaffordable, like Virtual Reality (VR). In this paper, we are especially interested in the question of how VR technologies can be successfully applied to Systems Engineering in the context of the Manufacturing Industry (Manufacturing Engineering). The paper is organized as follows. Section tworeviews the concept of the Factory of the Future (FoF) anddiscusses its challenges and recommendations given by four different reports on the subject. Section three presents a number of VR initiatives in the context of a traditional factory, and subsequently compiles the most promising ways for an efficient use in the FoF. Finally, section four concludes by discussing the potential role of VR in the FoF.

- **Abstract/Introduction References:**

[1] M. Maurer, H. Winner (eds.), Automotive Systems Engineering, Springer, Heidelberg, 2013, ch 1 pp.3-35.

[2] M. Mihelj, D. Novak, S. Bagus, Interaction with a Virtual Environment, in Virtual Reality Technology and Applications, Springer Netherlands, 2014, pp.205-212.

[3] Systems Engineering Handbook, a guide for system life cycle processes and activities, v. 3. 2. 2, INCOSE, October 2011, ch1, pp.7-8.

2nd Additional paper:3D Interaction in Virtual Environments for Systems Engineering

- **Paper title:** 3D Interaction in Virtual Environments for Systems Engineering
- **Authors:** Mohammed Bougaa, Stefan Bornhofen, Hubert Kadima, Alain Rivière.
- **Journal:** International Journal of Computer Theory and Engineering.
- **Volume:** Vol. 8, No. 6.
- **Date:** December 2016
- **Keywords:** Virtual reality, virtual environment, systems engineering, 3d interaction, adaptive interaction.
- **Abstract/Introduction:**

  This paper discusses the potential benefits of applying Virtual Reality (VR) technology to the context of Systems Engineering (SE), for both educational and industrial purposes. After an introductive presentation of the two fields and their state of the art, we explore if and how VR can be of assistance to the processes involved in a typical SE approach. We especially focus on commonly used 3D interaction techniques in VR and argue that the design of appropriate 3D interactions is a key ingredient for the success of VR in SE. We suggest three research directions that may be considered for this design: interaction generality, context awareness and adaptability. The 3D interactions should adapt, manually or automatically, to the VR device, the virtual scene and the user context

  Introduction:

  Over the last decades, the engineering of physical systems and the management of their life cycle have grown more and more complex. Military, aerospace, automotive and medical industries are confronted with the challenge of building systems that have large architectures and require interdisciplinary competence. Moreover, the systems must not only be functional, but also reliable, maintainable and safe. As an example, Fig. 1 illustrates the multidisciplinary nature of the design process of an aerospace launch vehicle at the Onera research center [1]. The understanding and the design of such complex systems cannot be achieved any more by a simple Systems Design approach which is the concept, specification, implementation, verification and validation of a technical system for achieving a specific (and mostly functional) objective [2]. New goal-independent methodological approaches and modeling techniques are required, such as Systems Engineering (SE) whose first concern is not a specific design objective, but the optimization of all involved engineering processes. As defined by the International Council on Systems Engineering (INCOSE): "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs". [3]. Much research effort has focused on proposing appropriate tools for SE. Dedicated modeling languages have been introduced, such as

SysML [4], a visual modeling language, and LML (Lifecycle Modeling Language) [5], providing a simple way to understand and communicate cost, schedule and performance design information to all stakeholders in a standard manner. Other successful standardizations are ISO/IEC 15288 [6], defining a common framework for the life cycle of systems, EIA 632 [7] proposing the blueprint of all processes and requirements necessary for engineering a system, or IEEE 1220 [8] providing the next-level-of-detail description of SE processes defined in EIA 632. However, not all technologies that might support the SE approach had full attention of the industrial and academic communities so far, either because they are recent like Big Data, or because up until recently they were immature and unaffordable, like Virtual Reality (VR). As a matter of fact, only little research has been done on VR for SE, although its potential has already been discovered especially for educational needs [9]. This paper pursues the assessment of the benefits of VR in the context of SE. We intent to develop a collaborative VR based environment for using and learning SE, we are particularly interested in the study of 3D interaction techniques and explore if and how they should be improved or adapted, in order for systems engineers to take maximum advantages from a VR environment. The paper is organized as follows: Section II discusses the requirements to be met by a typical SE environment. Section III reviews the state of the art in VR, and distinguishes between the components sufficiently mature to be applied to SE and those that require improvements. Section IV focuses on immersive 3D interaction techniques and puts forward some directions to explore, in order to take VR to a higher level of maturity, in terms of precision and ease of use.

- **Abstract/Introduction References:**

[1] S. Defoort, M. Balesdent, P. Klotz, P. Schmollgruber, J. Morio, J. Hermetz, C. Blondeau, G. Carrier, and N. Bérend, "Multidisciplinary aerospace system design: principles, issues and onera experience," Onera, AerospaceLab Journal, no. 4, p. 11, May 2012.

[2] M. Maurer and H. Winner, Automotive Systems Engineering, Springer, Heidelberg, 2013, ch. 1, pp. 3-35.

[3] Systems Engineering Handbook, a Guide for System Life Cycle processes and Activities, v. 3.2.2, INCOSE, October 2011, ch. 1, pp. 7-8.

[4] L. Balmelli, "An overview of the systems modeling language for products and systems development," Journal of Object Technology (JOT), vol. 6, no. 6, pp. 149-177, July 2007.

[5] Lifecycle Modeling Language (LML) specification, v 1.0, Technical Report. (October 2013).

[6] Systems and Software Engineering -System Life Cycle Processes-, ISO/IEC International Standard, ISO/IEC 15288, 2 nd ed., 2008.

[7] Processes for Engineering a System, ANSI/EIA Standard, EIA-632-1998, Approved: January 1999.

[8] Standard for Application and Management of the Systems Engineering Process, IEEE Standard, IEEE 1220-1998, 1998.

[9] A. G. Abulrub, A. N. Attridge, and M. A. Williams, "Virtual reality in engineering education- the future of creative learning," presented at IEEE Global Engineering Education Conference, Amman, Jordan, April 4-6, 2011.

Résumé de thèse en Français

Faire face aux systèmes complexes d'aujourd'hui nécessite des ingénieurs systèmes hautement qualifiés. Mais pour être efficace en tant qu'ingénieur système, les étudiants en différentes disciplines d'ingénieries traditionnelles ont besoin d'une expérience pratique et réelle dans ce type de systèmes complexes, en plus des connaissances théoriques nécessaires dans leur discipline.

L'éducation dans le domaine de l'ingénierie des systèmes (IS) concerne aussi bien les compétences et les processus, que les faits transférables et les connaissances. Cela rend l'education en IS un sujet très difficile, avec un besoin de considérer à la fois la complexité des systèmes à concevoir, ainsi que la complexité et l'abstraction de l'IS dans certains de ses principes fondamentaux. Des efforts considérables ont été accomplis afin d'améliorer les pratiques de cette discipline et pour faire face à la complexité des systèmes actuels et à l'éducation en IS. Ces efforts sont fournis par les organisations utilisant l'IS et les établissements universitaires et collèges formant des ingénieurs. Ils ont conduit à la production de normes, de documentations pour ces normes, des modèles d'évaluation des processus, ainsi que des modèles de compétences, tout en améliorant le déploiement et la pratique de cette discipline dans l'industrie.

Au cours de cette thèse, nous avons examiné les pratiques actuelles en matière d'éducation en IS, et nous avons conclu qu'il n'existe pas de solution d'apprentissage formelle, unique, et adoptée par la communauté pour cette discipline. Une solution capable de répondre aux défis de l'industrie, et de produire des ingénieurs systèmes bien formés ou des ingénieurs spécialisés ayant des connaissances minimales viables sur les principes fondamenteaux de l'IS. Une solution qui devrait être facilement transférable d'une université à une autre, et d'une discipline d'ingénierie à une autre, et qui permet de former des ingénieurs système de différents niveaux d'expertise en largeur et/ou en profondeur sur les principes de l'IS. Nous avons ensuite essayé de prendre en compte quelques aspects de base du domaine de l'éducation, comme le modèle pédagogique qui devrait ètre utilisé et le rôle que la technologie et les formateurs devraient jouer dans un environnement idéal d'éducation en IS. Nous avons fini par proposer une approche novatrice pour former les étudiants sur les principes fondamenteaux de l'IS, indépendamment de leur discipline technique. Nous avons proposé une approche centrée sur l'utilisation des processus normalisés en IS, tout en faisant en sorte que le scénario d'apprentissage soit très adaptable et qu'il puisse être piloté par les compétences d'IS acquises ou à acquérir. Ensuite, une plateforme Web a été développée pour soutenir cette nouvelle approche d'apprentissage. Enfin, un échantillon d'étudiants de premier cycle et de leurs formateurs ont été interrogé sur l'utilité et l'efficacité de cette solution, et un autre échantillon de doctorants l'avait expérimenté dans un cours d'initiation à l'IS, dans le cadre d'une approche d'apprentissage par projet (APP).

En utilisant cette approche, les étudiants ne seront pas seulement en mesure de bien concevoir le système demandé de manière distante et collaborative, mais ils seront aussi capables de l'élaborer de manière appropriée. Cela leur permettra d'apprendre les principes et processus fondamentaux de l'IS, à mieux communiquer dans un environement de travail, la gestion d'équipe, la collaboration et les compétences techniques connexes. Les formateurs d'un autre côté pourront mieux gérer leur parcours d'apprentissage, les ressources pédagogiques, et les résultats escomptés. En utilisant cette solution, les organisations de ces formateurs et étudiants, c'est-à-dire les universités et les collèges, pourront gérer et normaliser les compétences acquises par leurs futurs ingénieurs systèmes à tous les niveaux.

Mots-clefs Ingénierie des Systèmes (IS), Education de l'IS, Normes de l'IS, Apprentissage Par Projet, Modèle de Cycle de Vie, Modèles de Compétences.

Titre : Une Approche basée sur les processus et dirigée par les Compétences pour l'Education en Ingenierie des Systèmes

Mots clefs : Ingénierie des Systèmes (IS), Education de l'IS, Normes de l'IS, Apprentissage Par Projet, Modèle de Cycle de Vie, Modèles de Compétences

Résumé : Au cours de cette thèse, nous avons examiné les pratiques actuelles en matière d'éducation en IS. Nous avons proposé une approche centrée sur l'utilisation des processus normalisés en IS, tout en faisant en sorte que le scénario d'apprentissage soit très adaptable et qu'il puisse être piloté par les compétences d'IS acquises ou à acquérir. Ensuite, une plateforme Web a été développée pour soutenir cette nouvelle approche d'apprentissage. En utilisant cette approche, les étudiants ne seront pas seulement en mesure de bien concevoir le système demandé de manière distante et collaborative, mais ils seront aussi capables de l'élaborer de manière appropriée. Cela leur permettra d'apprendre les principes et processus fondamentaux de l'IS, à mieux communiquer dans un environement de travail, la gestion d'équipe, la collaboration et les compétences techniques connexes. Les formateurs d'un autre côté pourront mieux gérer leur parcours d'apprentissage, les ressources pédagogiques, et les résultats escomptés. En utilisant cette solution, les organisations de ces formateurs et étudiants, c'est-à-dire les universités et collèges, pourront gérer et normaliser les compétences acquises par leurs futurs ingénieurs systèmes à tous les niveaux.

Title : A Process-Centered and Competency-Driven Approach for Systems Engineering Education

Keywords : Systems Engineering, Systems Engineering Education, Systems Engineering Standards, Project-Based Learning, Lifecycle Model, Competency Models

Abstract : This thesis surveys the current practices and advancements in SE education. We ended up proposing a novel solution for SE education (an approach with its supporting web-based platform). The proposed approach is based on the recommendations of academic and industrial communities. It is centered around the use of SE standardized processes and at the same time very adaptive, with learning scenarios that can be driven by the acquired or to-be-taught SE competencies. The proposed solution is a web based platform that has been developed to support this novel approach within a distant Project Based Learning (PBL) environment. The solution aims to ease the learning at the same time of fundamental principles and processes of systems engineering, along with communication, team management, collaboration, and related soft skills. On the other hand, educators will be able to better manage their learning scenarios, training resources, and the expected outcomes. Last, educators and students' organizations using this solution will be able to manage and normalize the competencies to be acquired by their future systems engineers at every level.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France