Model-Based Systems Product Line Engineering of Physical Protection Systems
INCOSE International Symposium
Tekinerdogan, Bedir; Özcan, Murat Kaan; Yakın, İskender; Yağız, Sevil
https://doi.org/10.1002/j.2334-5837.2021.00822.x

This publication is made publicly available in the institutional repository of Wageningen University and Research, under the terms of article 25fa of the Dutch Copyright Act, also known as the Amendment Taverne. This has been done with explicit consent by the author.

Article 25fa states that the author of a short scientific work funded either wholly or partially by Dutch public funds is entitled to make that work publicly available for no consideration following a reasonable period of time after the work was first published, provided that clear reference is made to the source of the first publication of the work.

This publication is distributed under The Association of Universities in the Netherlands (VSNU) 'Article 25fa implementation' project. In this project research outputs of researchers employed by Dutch Universities that comply with the legal requirements of Article 25fa of the Dutch Copyright Act are distributed online and free of cost or other barriers in institutional repositories. Research outputs are distributed six months after their first online publication in the original published version and with proper attribution to the source of the original publication.

You are permitted to download and use the publication for personal purposes. All rights remain with the author(s) and / or copyright owner(s) of this work. Any use of the publication or parts of it other than authorised under article 25fa of the Dutch Copyright act is prohibited. Wageningen University & Research and the author(s) of this publication shall not be held responsible or liable for any damages resulting from your (re)use of this publication.

For questions regarding the public availability of this publication please contact openscience.library@wur.nl

31st Annual INCOSE international symposium
virtual event
July 17 - 22, 2021

Model-Based Systems Product Line Engineering of Physical Protection Systems

Bedir Tekinerdogan
Wageningen University & Research, Information Technology
P.O. Box 8130, 6700 EW, Wageningen, The Netherlands
Phone: +31 (0)317 485477
bedir.tekinerdogan@wur.nl

Murat Kaan Özcan
Aselsan A.Ş.
Ankara, Turkey
mkozcan@aselsan.com.tr

İskender Yakın
Aselsan A.Ş.
Ankara, Turkey
iyakin@aselsan.com.tr

Sevil Yağız
Aselsan A.Ş.
Ankara, Turkey
syagiz@aselsan.com.tr

Copyright © 2021 by the authors. Permission granted to INCOSE to publish and use.

Abstract. A physical protection system (PPS) aims to protect a system against adversarial attacks by deterrence, detection, delay, and response. PPSs have targeted the protection of various systems, each of which has its specific requirements. On the other hand, PPSs also share a large set of features that are implemented for each system. To reduce the development cost, reduce time-to-market, and increase the quality of systems, a large-scale systematic reuse approach as defined by systems product line engineering (SPLE) can be applied. So far, PPS methods have mainly considered the systems engineering of single PPSs. In this paper, we report on our industrial experiences and lessons learned for adopting SPLE for PPSs. An explicit model-based systems engineering approach is adopted in which the focus is on the formalized application of models in the overall systems engineering life cycle. Hereby, the presented approach adopts and integrates model-based systems engineering, product line engineering, and the PPS method to provide a systematic large-scale reuse approach for systems engineering of PPSs. We discuss the detailed steps of the approach and report on the lessons learned for adopting model-based systems product line engineering for PPS.

Introduction

A physical protection system (PPS) aims to protect a system against adversarial attacks by deterrence, detection, delay, and response. Developing PPS is a systems engineering approach (SEBOK, 2016) to integrate people, procedures, and equipment (Garcia, 2006; 2008). PPSs have been developed to protect various systems, including highways, hospitals, airports, rail transport, bridges, dams, power plants, seaports, the electricity grid, oil refineries, and water systems. Engineering effective PPSs requires careful consideration of the requirements and the resources to provide the needed protection. Several PPS development methods have been proposed in the literature to support the systematic development of PPSs. These methods typically define the steps for deterrence, detection, delay, and response measures necessary to protect a physical system. Although the existing PPS methods have been successfully applied, we propose the enhancements of these from two perspectives.

First of all, it is necessary to adopt a formalization of models adopted by the model-based systems engineering paradigm to better support life cycle activities. So far, no explicit focus has been provided on developing model-based systems engineering methods for PPS. Existing PPS methods in the literature provide the well-defined steps to develop and analyze PPSs, but these are not at the level that is required by model-based systems engineering. Thus, the challenge is to develop a method with the proper modeling abstractions for supporting model-based systems engineering of PPS.

The second observation is that the current PPS methods seem to be focused on developing a single system and likewise do not consider systematic, planned, and large-scale reuse. While each PPS is unique concerning its features, we can observe that PPSs share a high percentage of common features. Obviously, there is a large potential for reuse in PPS systems development, but this is not fully exploited in current PPS development methods. Reuse has been an important goal in many industrial practices and is broadly addressed in the literature. While reuse was initially focused on a small scale, ad hoc reuse, currently, it is widely recognized that the broadest and the most valuable benefits are derived from a large-scale systematic reuse approach. This idea has culminated in the product line engineering (PLE) approach that focuses on exploiting reuse over the whole lifecycle process (Clements & Northrop 2002; Pohl et al. 2005; Linden et al., 2007). Traditionally, a product line is defined as a set of systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission developed from a common set of core assets in a prescribed way. Despite earlier reuse approaches, SPLE aims to provide pro-active, pre-planned reuse at a large granularity (domain and product level) to develop applications from a core, shared asset base.

Based on these observations, we report our experiences on a model-based systems product line engineering approach for PPSs. The research context is a systems engineering company (ASELSAN 2020) that is developing a broad range of PPSs for various industrial domains. On the one hand, the company wishes to adopt model-based systems engineering to better support communication of stakeholders' concerns, guide the design decision, and analyze the overall system. On the other hand, it aims at adopting a smarter reuse-based development approach for PPS using the product line engineering paradigm. We will report the required challenges and the adopted solutions of the research using an action research approach.

The remainder of the paper is organized as follows. Section 2 presents the background and preliminaries for the remaining part of the paper. Section 3 presents the adopted research approach. Section 4 presents the results of the feasibility analysis for PLE for PPS. Section 5 discusses the product line scoping for the PLE approach. Section 6 presents the integrated method. Finally, section 7 presents the discussions and lessons learned.

Background

As stated in the introduction section, the aim is to develop a (1) model-based systems engineering method for (2) product line engineering of (3) physical production systems. In the following subsections, we will discuss these domains in more detail.

Model-Based Systems Engineering

A model is typically defined as an abstraction of reality in the general sense. Modeling plays an important role in systems engineering and is used to communicate, analyze, or guide the production process. Models can be distinguished based on several criteria, one of which is the level of precision. Likewise, a model can be considered a Sketch, a Blueprint, or an *Executable*. A sketch is an informal model often used in the initial stages of the systems development process. A blueprint is a well-defined and precise model used as a guideline by a human engineer to develop a system. Executable models are more precise than sketches or blueprints and can be used by machines (computers) to automatically develop a more refined model. Thus, an executable model has everything required to produce the desired functionality of a single domain. In software engineering, a distinction is made between *model-based software engineering* in which primarily blueprints are adopted. In contrast, *model-driven software engineering* uses executable models often defined by the corresponding domain-specific languages.

In the systems engineering community, the notion of model-based systems engineering (MBSE) was introduced and popularized by INCOSE when it kicked off its MBSE Initiative in January 2007. The primary reasons for adopting MBSE include increased productivity by minimizing unnecessary manual transcription of concepts when coordinating large teams' work. Several definitions have been provided for MBSE. INCOSE defines it as "Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." (INCOSE, 2015)

For realizing MBSE, the proper model abstractions must be defined in the phases of the overall life cycle process (Gurbuz & Tekinerdogan, 2018; Tekinerdogan et al., 2005). For example, an important aspect of model-based systems engineering is model-based systems architecting. Architecture design is basically about *modeling* the system from different perspectives. The concepts related to architectural description are formalized and standardized and require the explicit usage of *views* that conform to well-defined formalized *viewpoints.*

Systems Product Line Engineering

The benefits of adopting a product line approach have been analyzed and discussed before by several authors (Clements et al., 2002; McGregor et al., 2002; Schmid & Verlage, 2002). Several studies show remarkable benefits of the organizations that are aligned with commonly held business goals including large-scale productivity gains, decreased time to market, increased product quality, decreased product risk, increased market agility, increased customer satisfaction, more efficient use of human resources, ability to effect mass customization, ability to maintain a market presence, and ability to sustain unprecedented growth.

Compared to single system development, applying a product line engineering approach requires additional investments. The initial investment will result in a so-called *return on investment* (ROI). Adopting the product line engineering approach will usually pay off after the development of more than one product. This point is denoted as the *break-even point*. Although different PLE processes have been proposed, they share the same concepts of *domain engineering* in which a reusable platform and product line architecture is developed, and *application engineering*, in which the results of the domain engineering process are used to develop the products. The overall development process is further controlled by a management process that consists of technical and organizational management. The typical common PLE process is shown in Figure 1.

The PLE process is agnostic to the domain and can be applied to developing a product line for any domain. Yet, for the development of PPSs, we also need to focus on the domain-specific PPS aspects. Hence an integrated PLE process for PPS is needed (Tekinerdogan et al., 2020c).

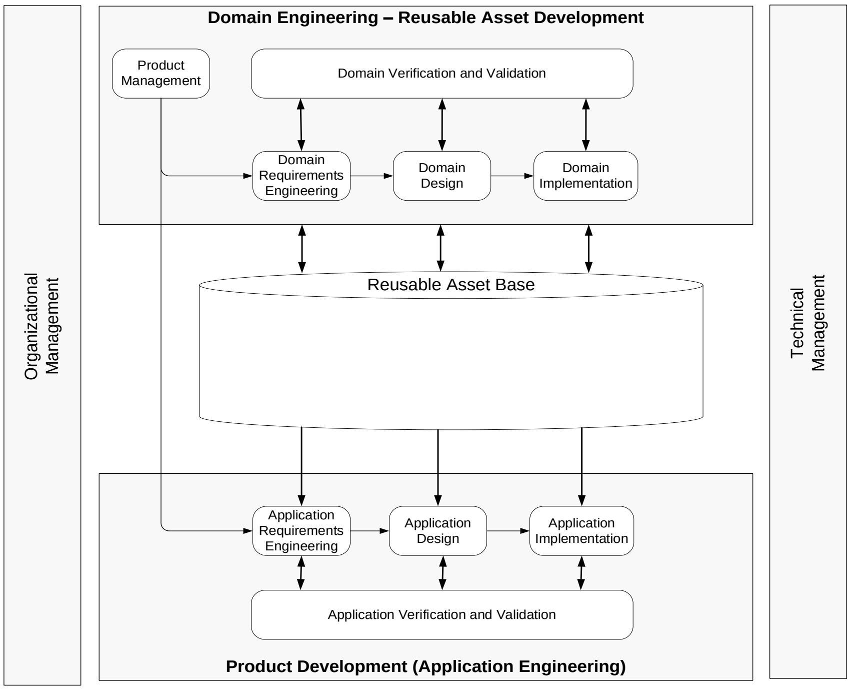

Figure 1. PLE Process

Physical Protection Systems

Developing PPSs requires the understanding and application of multiple disciplines and, as such, requires a systems engineering approach. The traditional systems engineering lifecycle process is often presented as a V-model, a general purpose-process that can be applied to develop various systems. Since it does not directly consider the domain-specific concerns, several domain-specific methods have been proposed for developing PPSs. Figure 2 shows the top-level activities for a typical PPS design process that we have modeled in our earlier study (Tekinerdogan et al. 2020c; 2020d). In essence, the PPS process consists of three key activities: determining the PPS objectives, designing the PPS, and evaluating the PPS. Each of these activities can be further refined; the top-level activities of these processes are also shown in Figure 2.

Determining the PPS objectives includes the facility characterization, the threat definition, and the target definition that needs to be protected. Designing PPS focuses on three activities, detection, delay, and response. The resulting PPS design should meet the facility's defined objectives and operational, safety, legal, and economic constraints. The final step in the PPS lifecycle is the evaluation of the design PPS. Several techniques can be distinguished here, including Path Analysis, Scenario Analysis, and System Effectiveness Analysis. For a detailed analysis of these process activities, we refer to (Garcia, 2006; 2008).

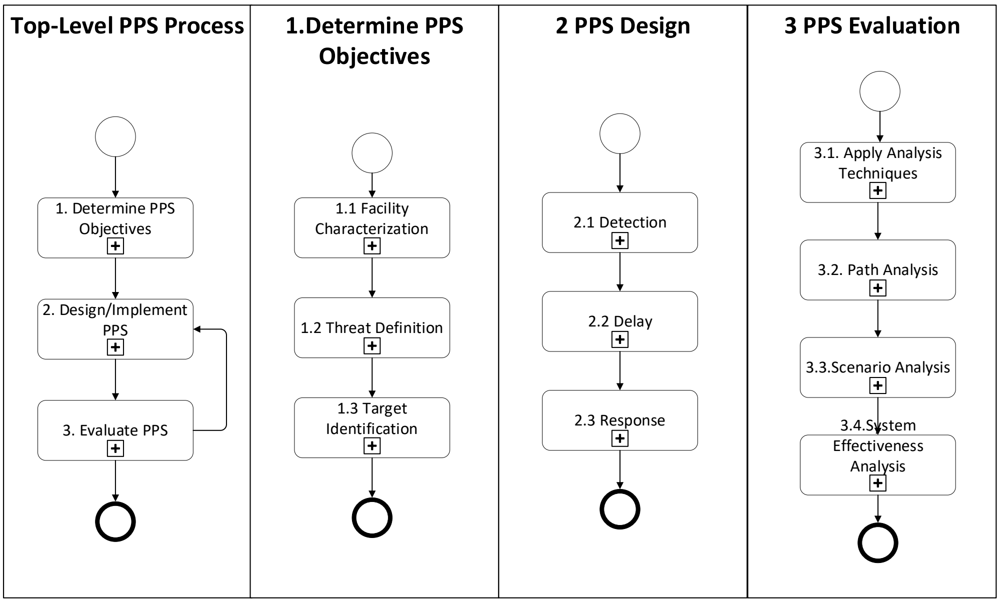

Figure 2. PPS Design Process activities (adopted from: (Tekinerdogan et al., 2020))

Problem Statement and Research Approach

Developing PPS can be done using a single systems engineering or product line engineering approach. In the traditional single systems engineering approach in which no PLE is adopted, a product portfolio usually can exist, but systems are developed separately. PLE practices such as explicit commonality variability modeling, a product family architecture, and a shared asset base are not adopted. The traditional way of developing PPS fails to see and exploit the potential for reuse. Although PPSs are different, they still share a common structure and features. The larger the commonality is, the more reuse potential we can identify. This means that a company that targets the development of multiple PPS can adopt a smarter, reuse-based approach. Hence, a PPS product line can be anticipated and developed from a common set of core assets in a prescribed way.

To investigate the feasibility, adoption, and development of the model-based systems PLE for PPS, we have adopted an empirical research approach based on so-called *action research* (Baskerville, 1999) an essential and valid instrument for solving research and development problems within an industrial context. Action research is an empirical research methodology whereby researchers attempt to solve a real-world problem while simultaneously studying the experience of solving the problem.

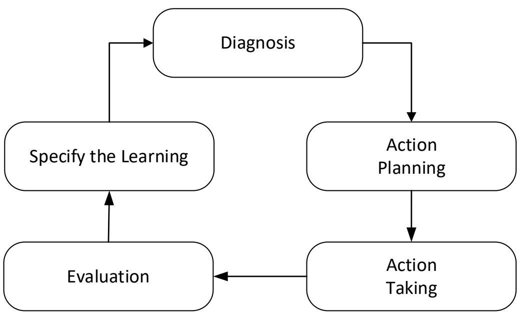

Figure 3. Typical Action Research Cycle

Action Research has been widely used as a research approach in social science, and it has been adopted for information systems and software engineering research during the last two decades.

According to Baskerville (1999), action research studies share four common characteristics: (1) An action and change orientation (2) A problem focus (3) An "organic" process involving systematic and sometimes iterative stages (4) collaboration among participants. In general, action research consists of five necessary steps, as shown in Figure 3.

Diagnosis corresponds to the identification of primary problems triggering the desire for a change in an organization. In this stage, theoretical assumptions about the nature of the organization and its problem domain are formulated. *Action planning* is the phase where you plan the actions to address the identified problems in the diagnosing phase. In this phase, the desired future state is formulated and the actions to achieve this desired state are listed. *Action taking* is the phase where the actions that are planned in the action planning phase are executed. *Evaluating* is this phase in which the researchers evaluate whether the theoretical effects of the actions are realized or not. *Specify the learning* is the identification of what has been learned from the cycle, regardless of implementation having been successful or not. These learning outcomes will be used to decide how to proceed for possible further cycles.

Very often, action research is carried out as one complete cycle of the above activities. However, due to the action research subject's unpredictable nature, the action research cycle can be iterated several times. Typically, action research is conducted by a team, including the target system participants, members of the project group that is initiating the change, and the external researchers.

In this paper, we aim to provide a model-based SPLE for PPS process. The steps of action research have been applied to various steps, including:

1. *Feasibility Analysis of PLE for PPS*
   Before adopting PLE the feasibility should be analyzed for PPS. This will be done using existing feasibility analysis methods in the literature.

2. *Identification and development of PLE Method*
   This implies identifying the PLE methods in the literature and the explicit process modeling of the selected method.

3. *Identification and development of PPS Method*
   This implies identifying the PLE methods in the literature and the explicit process modeling of the selected method.

4. *Integrated Design Process for PLE for PPS*
   This implies the integration of the developed PLE method with the PPS method.

5. *Development of the modeling abstractions for PLE for PPS method*
   To support model-based systems engineering the missing and necessary modeling abstractions will be developed. In particular we have focused on the development of the systems architecture framework.

6. *Application and enhancement of the method*
   Once the method is ready, it will be applied to case studies and if necessary, the above steps will be iterated and enhanced.

The causal connection and the workflow of these steps is shown in Figure 4. The adopted design method is shown in Figure 4. In the following sections, we will describe these steps in more detail.

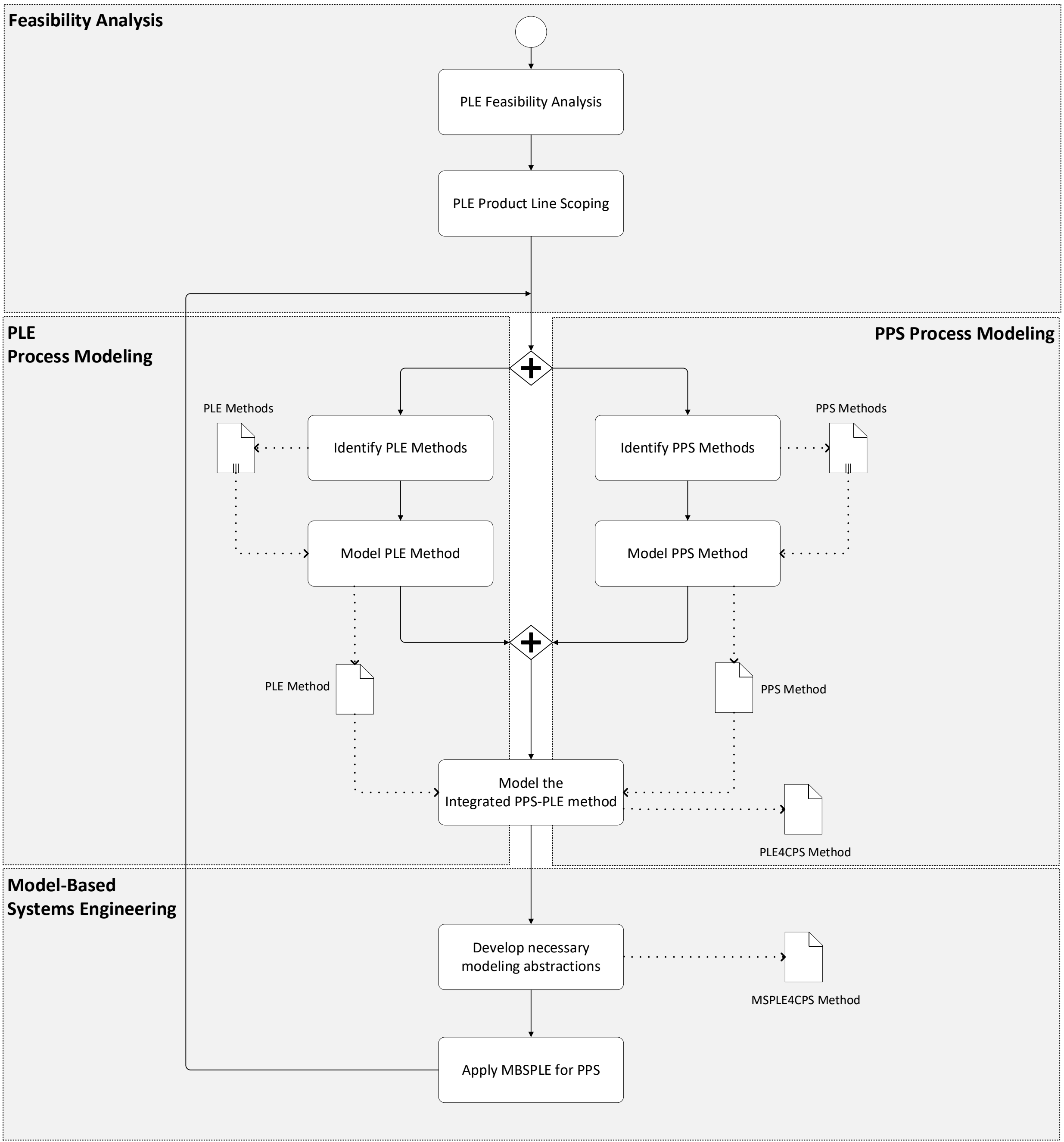

Figure 4. Adopted Design Method for Deriving Integrated PPS-PLE process

Feasibility Analysis

Adopting product line engineering is not trivial and involves many different risks, which could easily impede the expected benefits and the higher-level business goals. Hence before initiating and adopting a product line engineering approach, it is important to apply a feasibility analysis that depicts the current situation of the organization regarding the readiness for PLE and also indicates the potential risks.

Obviously, different PLE feasibility analysis methods can be identified in the literature. For our purposes, we have applied three feasibility analysis approaches, including the Product Line Potential Analysis (PLPA) (Fritsch & Hahn, 2004), Product Line Technical Probe (PLTB) (Clements, 2002), and Transit-PL (Tüzün et al., 2015).

PLPA aims to quickly assess whether PLE is suitable for a given set of products and market. The Product Line Potential Analysis is executed in a half-day workshop with a structured interview based on a questionnaire to examine products, software, markets, and customers. The PLPA aims to answer the following question: *Would the PLE approach be suitable for a given set of products and target market?* Since this question is too hard to answer as a whole, a set of criteria relevant to PLE's applicability are defined.

PLTP includes the processes *Product Line Quick look* (PLQL) and *Product Line Technical probe* (PLTP) for examining the organization's readiness to transition to PLE. PLQL is the initial gathering of information about the organization, while PLTP is a more thorough analysis of the organization for SPL readiness. Both PLQL and PLTP use the Framework Software Product Line Practice and Product Line Adoption Map as reference models. PLQL consists of a one-day session where experts interview organization product line sponsors, primary technical leads, and architects. PLTP consists of structured interviews of small groups that are selected from product line stakeholders. The results of the interviews are analyzed based on the 29 practice areas as specified in the framework.

Transit-PL is a comprehensive PLE feasibility analysis process that adopts 25 different aspects: business, architecture, process, and organization aspects. The feasibility is analyzed by an extensive questionnaire that results in radar charts to visualize the readiness level for each aspect. Figure 6 shows a randomly configured radar chart to illustrate the outcome of the Transit-PL. Note that this does not reflect the real outcome but is shown for illustration purposes only.

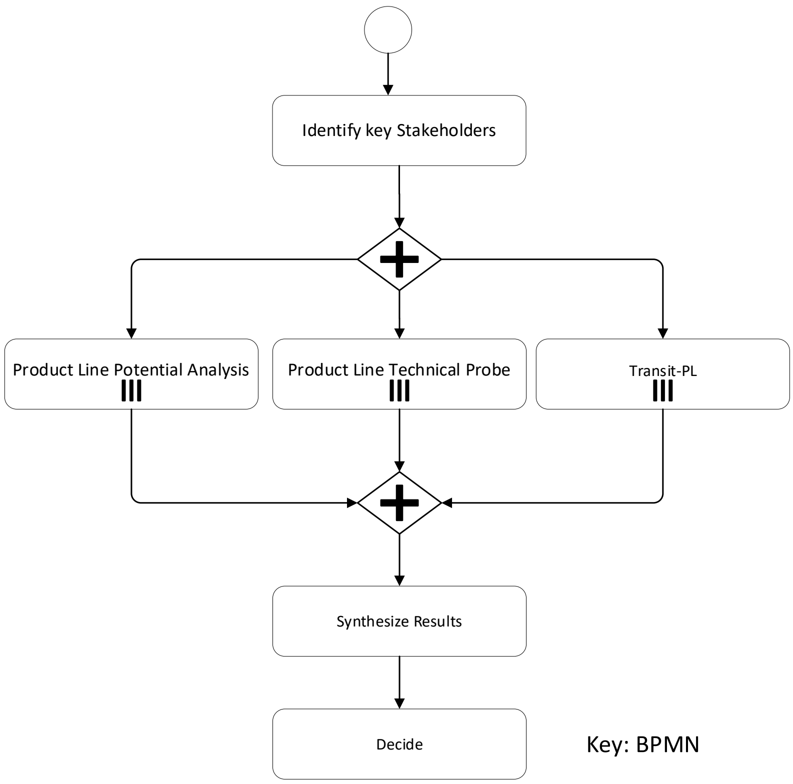

Figure 5. Integrated Feasibility Analysis Process

We have applied all these three methods for the PLE feasibility analysis within the industrial context. The workflow diagram for this is shown in Figure 5. The process starts with the identification of the stakeholders that are necessary for performing the feasibility analysis. For each method, a separate questionnaire has been prepared that is implemented using Google Forms. The answers have been collected and synthesized. All these results are then synthesized and used to provide the current situation of the industrial context as well as the PLE feasibility. The feasibility analysis's typical output is a radar chart (example shown in Figure 6) that visualizes the readiness levels for various feasibility parameters. For a detailed discussion on the feasibility analysis parameters, we refer to our earlier work on Transit-PL (Tüzün et al., 2015).

Figure 6. Example (hypothetical random) output of the feasibility analysis using a radar chart

Product Line Scoping

The systems in the product line are not randomly selected. Typically, systems are targeted that are likely to achieve the most economic benefit; and can be efficiently developed with the core assets either planned or in hand. An essential activity in PLE is the product management step that aims to control the development, production, and marketing of the product line and its applications. The input for product management consists of the company goals defined by top management. The purpose of product management is to contribute to entrepreneurial success by integrating the development, production, and marketing of products that meet customer needs. An essential task of product management is thus managing a company's product portfolio, which is defined as the product types provided by the product line organization. Portfolio management is a dynamic decision process that continually checks and updates the portfolio according to the market and business requirements. The product management sub-process specifies a *product roadmap*, which outlines the estimated product line and defines the significant common and variable features of all product line applications. The product roadmap is, in its turn, provided to the domain requirements engineering, which defines the product requirements based on which the product line architecture will be designed.

Product management employs *scoping* techniques to define what is within the product line's scope and what is outside. The success of the product line depends largely on the appropriate product line scope. If the scope is too large, the different product members will typically vary too much, and likewise, it will be more difficult to realize commonality and variability. The risk is then that the product line will collapse into the one-at-a-time product development effort. On the other hand, if the scope is too small, then the core assets might not accommodate future growth, and the product line will stagnate. As a result, it will be challenging to realize economies of scope and achieve the expected return on investment. Scoping should be done carefully to mitigate these risks.

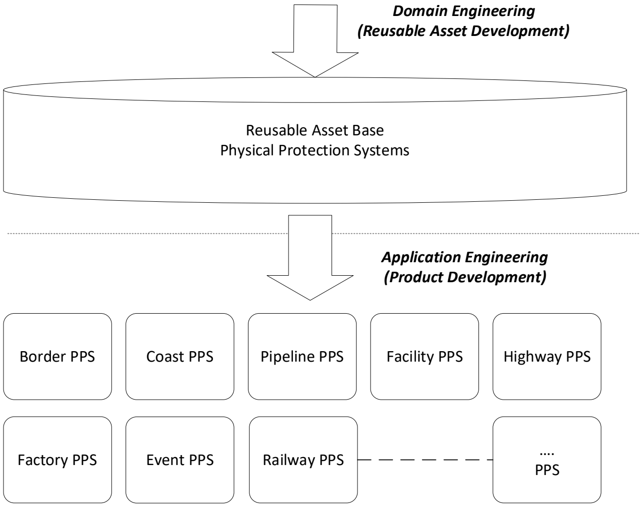

Figure 7. Overall Scope for PPS

The term PPS by itself is also very broad and includes several sub-domains. Further, the relationship between domains and systems is often many-to-many. Systems do not necessarily cover a whole domain and may belong to several domains. Also, a domain can be used in several systems, and several domains may be scattered under one system. Figure 7 depicts the overall domain with the identified sub-domains that are considered for our project. In the figure, several PPS domains have been identified, but the adopted global scope is not only limited to these domains.

Integration of PLE and PPS Processes

In the previous sections, we provided an explicit process model for both PLE (Figure 1) and PPS (Figure 2). To develop a PLE method for PPS we first considered each of the methods separately by studying the literature. For identifying the PLE methods, we have used the traditional methods as published in the PLE community. In this paper, we do not elaborate on the modeling of the PLE method since we have reported on these already in our earlier studies (Tekinerdogan et al. 2020). For the PPS methods, we have primarily consulted the methods of the following sources: (Garcia, 2006; 2008)(Fennelly, 2016)(Williams, 1997)(Axelsson, 2000)(IAEA 2000)(NIST 2007).

In this paper, we aim to model the PPS process and integrate this with the PLE process. In general, a process is defined as a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular set of customers. A business process model (BPM) is an abstract representation of a business process. In this paper, the primary goal of process modeling includes communication of the PPS process to different stakeholders, the guidance of the PPS process activities, analyzing the progress, and aligning the PPS process with the product line engineering process. Our focus is not on automated process guidance and/or automated execution of the process, although this could be indeed considered as a follow-up study

As stated before, the SPLE process is, in general, domain agnostic and can thus be applied to multiple application domains, and as such, fails to address domain-specific process concerns. On the other hand, the PPS process focuses on developing a single PPS and does not explicitly consider reuse. To provide a systematic product line engineering for PPS, we have integrated both processes, which is shown in Figure 8.

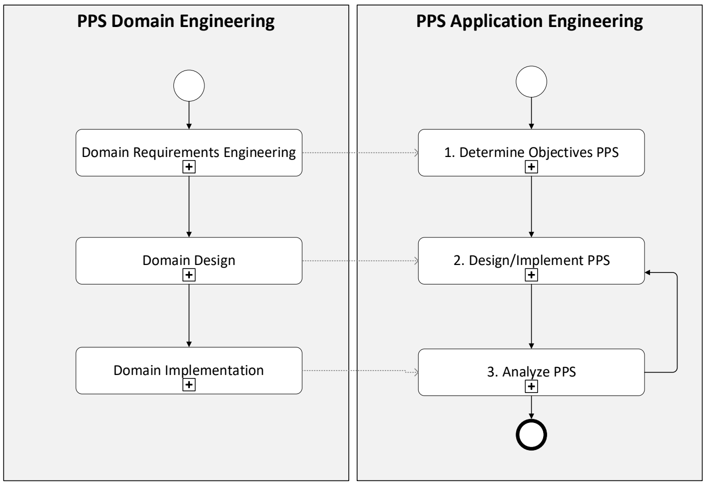

Figure 8. PPS Design Process integrated within the PLE Process

In essence, the dominant process model is the two-life cycle process of the PLE process consisting of a domain engineering process and an application engineering process. In the domain engineering process, the core assets for the PPSs that are envisioned can be developed. In a sense, this does not change the conventional domain engineering process steps. The domain requirements engineering will result in a family requirements specification, the domain design will provide the product line architecture, and the domain implementation will provide the necessary implementation of the identified core assets for the PPSs.

The PPS method, shown on the right part of the figure, is integrated with the application engineering process, which starts by identifying a particular PPS's objectives and requirements. However, these objectives and requirements are not developed from scratch but reused from the reusable asset base that is the result of the domain engineering process. Similarly, the design process follows the exact same process steps that we have described in the previous section but will primarily reuse the assets and the design that is needed for developing the PPS. Finally, the analysis of the PPS is based on the evaluation of the designed PPS.

Before, we stated that we wish to provide large scale systematic reuse for developing PPS. In particular, this is necessary if a company is developing multiple different PPSs, which are based on a common product line architecture and a substantially large part of commonality. One could state that this could be just developed using conventional PLE methods. However, in particular, PPS design requires very domain-specific specific steps regarding the design of deterrence, detection, delay, and response actions. As such, it is needed to represent not only the artifacts but also the PPS process steps in the PPS product line engineering. The method shown in Figure 8 accomplishes both goals. On the one hand, it ensures that PPS's key concern that is protection is properly addressed. On the other hand, it helps to develop PPSs faster, with lower cost and higher quality.

Modeling Abstractions

The integration of the PLE and PPS methods support the reuse-based development of PPS systems. What remains was to provide a model-based systems engineering approach. For this, we largely relied on the existing modeling abstractions as provided by SysML and UML. However, although developing a PPS is in essence, a systems engineering approach the existing systems engineering processes are limited to address the domain-specific aspects of PPSs. As such dedicated processes have been proposed in the literature to develop PPSs. An important artefact of the design process is the systems architecture representing the PPS's fundamental structure. To design a proper PPS it is necessary to adopt a well-defined architecture framework with the corresponding viewpoints addressing the required concerns. Unfortunately, neither in systems engineering nor in the dedicated PPS life cycle processes, a suitable architecture framework has been proposed yet to model PPS architectures. Hence, in alignment with model-based systems engineering vision, we have provided an architecture framework (PPSAF) for designing physical protection system architectures. We have discussed this in detail in our earlier published paper (Tekinerdogan et al. 2020). To develop the required viewpoints, we have performed a thorough domain analysis to PPS and provided a meta-model that defines the PPS key concepts as shown in Figure 9. Subsequently, based on the meta-model we have derived a coherent set of six architecture viewpoints including facility viewpoint, threats and vulnerabilities viewpoint, deterrence viewpoint, detection viewpoint, delay viewpoint and response viewpoint. Each viewpoint can be used to model the systems architecture from the perspective of a particular concern held by the corresponding stakeholders..

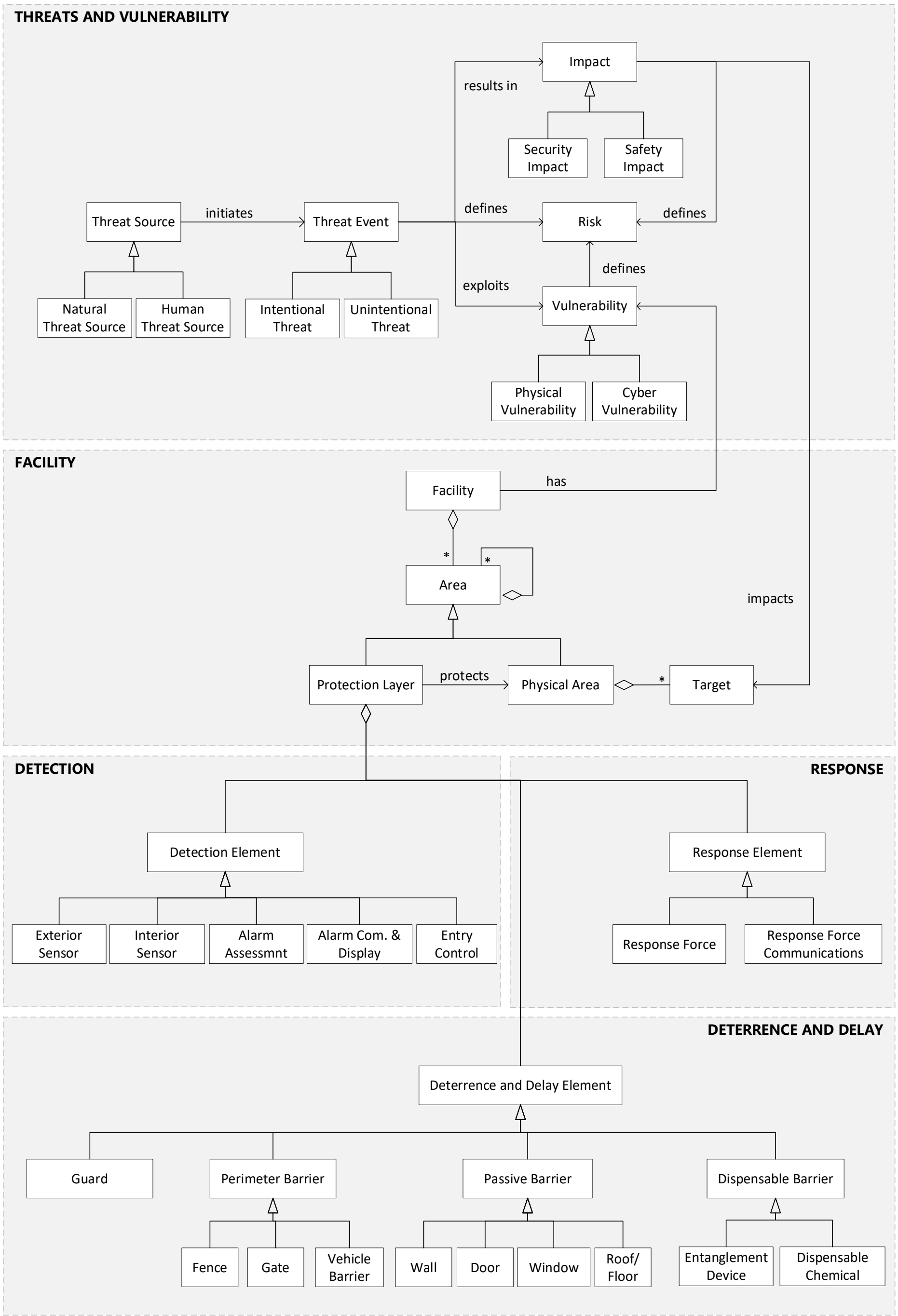

Figure 9. Metamodel for Physical Protection Systems (adapted from: (Tekinerdogan et al. 2020a))

Conclusion and Lessons Learned

This paper has provided the research results to product line engineering for physical protection systems (PPS). A PPS integrates people, procedures, and equipment to protect assets or facilities against theft, sabotage, or other malevolent intruder attacks. In essence, developing PPSs is thus a system engineering activity. However, with the increasing role of software designing PPSs requires an explicit consideration of software engineering aspects. In this context, we use the term software-intensive PPS to differentiate this from non-software related PPS.

In essence, whether it is software controlled or not, a PPS provides deterrence and a combination of detection, delay, and response measures to protect against an adversary's attempt to complete a malicious act. Many PPSs share the same set of features and as such there seems to be an obvious chance for exploiting the potential of reuse. The contributions of this study can be listed as follows:

Product Line Feasibility Analysis

We have provided a feasibility analysis approach for PLE for PPS within the industrial context of ASELSAN. From our detailed feasibility analysis study, we can conclude that PLE is indeed feasible for PPSs. To support triangulation and thus view the results from different perspectives, we have adopted three different methods. The overall analysis highlighted the current situation for the selected domains and the possible risks for PLE adoption. Overall, we can conclude that PLE seems applicable for PPS (Tekinerdogan et al. 2020b; 2020c; 2020d).

Product Management and Product Line Scoping

Product management is the sub-process of domain engineering for controlling the development, production, and marketing of the software product line and its applications. We have focused on different product line scoping techniques within the project, including domain scoping, product scoping, and asset scoping. Since the company focuses on a broad non-anticipated set of PPSs, the primary focus tended to be on domain scoping, in which we have defined the broad scoping of PPSs. The broad scope appeared to be feasible due to the large percentage of common features of the potential PPSs.

Explicit Modeling of the PPS and PLE Methods

The design of PPS is discussed in detail in the literature, and we have also benefited from these sources. However, none of these studies have provided a formal representation of the adopted methods. From the model-based systems engineering paradigm perspective, we have explicitly modeled both the PPS and PLE methods. This step paved the way for the integration of PLE with PPS method from a model-based systems engineering perspective (Tekinerdogan et al., 2020c).

Integration of PPS method with PLE method

Current PPS methods do not adopt a large product line or product family focus. On the other hand, existing PLE methods are agnostic to the domain of the products and as such lack the required focus on the specific process steps, such as that in PPS methods. With this observation, we have provided an approach that integrates the PPS method with the PLE method (Tekinerdogan et al., 2019; 2020c; 2020d). The PLE method has been considered the dominant decomposition of the process consisting of two life cycle processes, domain engineering and application engineering. The domain engineering process is largely the same as in conventional PLE, however the application engineering process has been adjusted with respect to the needs of the PPS process steps. Both in the PLE literature and PPS literature, this integration has not been discussed before.

Feature Modeling of PPS

Our earlier study has provided a comprehensive feature model for PPS that can be used to characterize many different PPSs (Tekinerdogan et al., 2020d). The feature model has been derived after a thorough domain analysis on PPSs and, as such, builds on existing literature. The distinguishing factor of this study is the explicit focus on the commonality and variability analysis. Feature modelling appeared to be a very useful approach for this purpose. Besides the ontological support and the focus on understanding the key features, the family feature model can also be used in the threats definition process, resulting in a design basis threat document. Further, the PPS family feature model can be embedded in product line engineering processes to support the design of a PPS architecture. We consider this also part of the future work.

Metamodeling of PPS

PPSs share important concepts, which we have explicitly described in a metamodel consisting of the key parts Threats and Vulnerability, Facility, Detection, Response, Deterrence and Delay. In essence, each PPS can be described using this metamodel. The metamodel further serves as a conceptual basis for defining the systems architecture framework for PPS (Tekinerdogan et al., 2020a).

Systems Engineering Architecture Framework for PPS

An important artefact of the design process is the systems architecture which represents the fundamental structure of the PPS. To design a proper PPS it is necessary to adopt a well-defined architecture framework with the corresponding viewpoints addressing the required concerns. Unfortunately, neither in systems engineering nor in the dedicated PPS life cycle processes, a suitable architecture framework has been proposed yet to model PPS architectures. Hence, in alignment with the vision of model-based systems engineering, we have provided an architecture framework PPSAF for designing physical protection system architectures (Tekinerdogan et al., 2020a). The PPSAF has been developed based on the metamodel that defines the PPS key concepts. Based on the metamodel we have derived a coherent set of six architecture viewpoints including facility viewpoint, threats and vulnerabilities viewpoint, deterrence viewpoint, delay viewpoint and response viewpoint. Each viewpoint can be used to model the systems architecture from the perspective of a particular concern that is held by the corresponding stakeholders.

Software Product Line Architecture

As stated before, most PPSs today are software-intensive and are controlled by software. The PPSs in the product line share a common product line architecture. Proper design and documentation of the architecture are crucial for the success of a product line. In alignment with model-based systems engineering, we have explicitly designed and documented the software-intensive product line architecture of PPS (Tekinerdogan et al., 2020b). For this, we have adopted the developed PPSAF architecture framework. To the best of our knowledge, this is the first time that a PPS architecture is documented using a model-based approach. In our future work, we will further develop the PPS product line using the presented product line architecture, thereby also addressing the other PLE lifecycle activities.

References

Axelsson, S 2000, Intrusion Detection Systems: A Survey and Taxonomy.

Baskerville, R.L., 1999. Investigating information systems with action research. Com- mun. AIS 2 (3), 4.

Clements, P, Northrop, L 2002. Software Product Lines: Practices and Patterns. Boston, MA: Addison-Wesley, 2002.

Drago, A 2015, Methods and Techniques for Enhancing Physical Security of Critical Infrastructures, PhD Thesis, University of Naples.

Fennelly, L 2016, Effective Physical Security, Fifth Edition (5th. ed.). Butterworth-Heinemann, USA, 2016.

Fritsch, C, Hahn, R 2004. Product line potential analysis, Software Product Lines, pp. 95–97, 2004.

Garcia, ML 2006, Vulnerability assessment of physical protection systems. Amsterdam: Elsevier Butterworth-Heinemann; 2006.

Garcia, ML 2008, The design and evaluation of physical protection systems. 2nd ed. Amsterdam: Elsevier Butterworth-Heinemann; 2008.

Gurbuz, HG, Tekinerdogan, B 2018, Analyzing Systems Engineering Concerns in Architecture Frameworks–A Survey Study, IEEE International Systems Engineering Symposium (ISSE), pp 1-8.
IAEA 2000, Handbook on the Physical Protection of Nuclear Material and Facilities, IAEA-TECDOC-127.
INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Ed. John Wiley & Sons, Inc. Published 2015.
McGregor, J. D., Northrop, L.M., Jarrad, S, and Pohl, K 2002, Initiating software product lines, IEEE Software, vol. 19, no. 4, pp. 24–27, Jul. 2002.
Van Der Linden, F, Schmid, K, and Rommes, E 2007, Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007, p. 334.
Mellor, S.J., Scott, K, Uhl, A, Weise, D, 2004. MDA Distilled: Principle of Model Driven Architecture, Addison Wesley, Reading.
NIST (National Institute of Standards and Technology) – Guide to Intrusion Detection and Prevention Systems (IDPS), February 2007.
Pohl, K, Böckle, G.F., van der Linden, F 2005. Software Product Line Engineering – Foundations, Principles, and Techniques, Springer.
Schmid, K, Verlage, M 2002. The Economic Impact of Product Line Adoption and Evolution. IEEE Software, Vol. 19, No. 4, 50-57.
SEBOK 2016, Guide to the Systems Engineering Body of Knowledge (SEBoK).
Tekinerdogan, B, Bilir, S, Abatlevi, C. 2005. Integrating Platform Selection Rules in the Model Driven Architecture Approach. In: Aßmann U., Aksit M., Rensink A. (eds) Model Driven Architecture. MDAFA 2004, MDAFA 2003. Lecture Notes in Computer Science, vol 3599. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11538097_11
Tekinerdogan, B & Aksit, M 2002, Classifying and evaluating architecture design methods, in: Software Architectures and Component Technology, Springer, pp3-27, 2002.
Tekinerdogan, B, Duman, S, Gümüşay, Ö, and Durak, B 2019. Devising Integrated Process Models for Systems Product Line Engineering. 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom.
Tekinerdogan, B, Ozkose Erdogan, O, Aktug, O 2014. Supporting Incremental Product Development using Multiple Product Line Architecture, International Journal of Knowledge and Systems Science (IJKSS) 5(4).
Tekinerdogan, B, Özcan, K, Yagiz, S, Yakin, I. 2020a. Systems Engineering Architecture Framework for Physical Protection Systems, International Symposium on Systems Engineering (ISSE), Vienna, Austria, October 12-14.
Tekinerdogan, B, Yakin, I, Yagiz, S, Özcan, K 2020b. Product Line Architecture Design of Software-Intensive Physical Protection Systems, International Symposium on Systems Engineering (ISSE), Vienna, Austria, October 12-14.
Tekinerdogan, B, Yagiz, S, Özcan, K, Yakin, I, 2020c. Integrated Process Model for Systems Product Line Engineering of Physical Protection Systems, In: Shishkov B. (eds) Business Modeling and Software Design. BMSD 2020. Lecture Notes in Business Information Processing, vol 391 (ISBN: 978-3-030-52305-3). Springer, 2020.
Tekinerdogan, B, Özcan, K, Yagiz, S, Yakin, I. 2020d. Feature-Driven Survey of Physical Protection Systems, 11th Complex Systems Design & Management (CSD&M) conference, Paris, Online.
Tüzün, E, Tekinerdogan, B, Kalender, M.E, Bilgen, S 2015. Empirical Evaluation of a Decision Support Model for Adopting Software Product Line Engineering, Information and Soft-ware Technology, Elsevier, Vol. 60, Pages 77–101.
Williams, J.D. 1997, Physical Protection System Design and Evaluation, IAEA-CN-68/29,
S.J. Mellor, K. Scott, A. Uhl, D. Weise. MDA Distilled: Principle of Model Driven Architecture, Addison Wesley, Reading , 2004.