Measures and approach for modernization of existing systems

Rustam Akhtyamov
Skolkovo Institute of Science and Technology
Moscow, Russia
r.akhtyamov@skoltech.ru

Rob Vingerhoeds
Department of Complex Systems Engineering
ISAE-Supaero
Toulouse, France
rob.vingerhoeds@isae.fr

Alessandro Golkar
Skolkovo Institute of Science and Technology
Moscow, Russia
a.golkar@skoltech.ru

***Abstract*— Due to accelerating technological progress any engineered system eventually becomes obsolete and the question arises as to its modernization. Disciplines such as change and configuration management propose methods to maintain system at operational level by incorporating relatively small changes. When major rebuild is required to the system, existing literature proposes to follow the same System Engineering processes as during new system design and development. Systems Engineering, traditionally focused on design and development of new systems, paying limited attention to modernization of existing ones and its processes need to be specially tailored to the modernization of existing system. In the case of straightforward application of Systems Engineering processes, the step of new system formulation is often neglected and considered as something already granted. This paper gives an overview of approaches and methods in the field of system modernization. It proposes a metrics for classification of different types of system modernization. To facilitate system modernization this work introduces a framework based on proposed metrics, Design Structure Matrixes and functional analysis. The framework consists of decomposition and synthesis phases and aims to define new (modified) system, by providing a set of alternatives. The proposed framework anticipates application of traditional System Engineering approach. The paper is illustrated with an example in the field of car retrofitting.**

Keywords—systems engineering; modification; modernization; retrofit; reengineering

I. INTRODUCTION

With ever-increasing accelerating technological and socioeconomic progress, modern technical systems become more and more rapidly obsolete. The obsolescence, as opposed to physical deterioration, by definition, is caused not by performance decrease and system aging due to components degradation, but by innovations and the appearance of new technologies and new products on the market.

In today's reality, almost every system is to some extent obsolete, and systems of different generations co-exist simultaneously. Although this diversity increases maintenance costs, obsolete systems cannot be simply disposed due to economic and other reasons. This issue is gaining more and more importance for all types of industrial sectors. Different systems have different average lifetimes: from 10.7 years for passenger cars [1] to 20 years and more for airplanes [2]. It raises the question how to bring new functionality and concepts into existing systems, when does it make sense, which performance increase can be brought to the system, which constraints need to be considered, and which strategy needs to be selected.

The process of improving system performance and adding new functionality to existing systems is referred under different terms in the literature. System modification and system change are usually interchangeable and include other terms, such as re-engineering, retrofit, update and upgrade. By modification this paper understands any change in a system, including addition of new elements and functionalities. Modernization is a subset of modification aimed at system improvements. The paper refers retrofit as a subset of modernization, focused on addition new functionality to existing system. By reengineering the paper understands internal changes to the original architecture.

Systems engineering methodology covers design and development of new systems mostly from scratch, consider design options in the light of the issues associated with the acquisition phase, but pays less attention to through-life support aspects [3], while in reality, most of the systems are based on previously existing systems. For modification of existing systems INCOSE proposes to formulate a new system and apply the same processes and principles that are employed during the upfront design, development, integration, and testing [4]. However, little attention is paid to the early steps of re-formulation of this new system [5].

This paper aims to support decision-making by identifying and evaluating which new functionalities to bring into existing system. It analyzes connection between performance, functionality and physical structure of the existing and new system and proposes an approach for transition from existing system to formulation of a new system, to facilitate system modifications and eventually to complement existing systems engineering methodology. The paper is organized as follows: The next section gives an overview of the literature related to modification, modernization and change of existing systems. The third section proposes measures to classify existing terminology and approaches based on relation between original and modified system in terms of structure, functionality and performance. The fourth section proposes an approach to facilitate process of definition of a new system, required to proceed from pre-modified system to traditional systems engineering steps of new system design. The approach is illustrated by example of passenger car retrofit with cruise control. The last section summarizes the results and gives a conclusion.

II. LITERATURE OVERVIEW

The existing literature discusses modification and modernization of existing systems from different viewpoints and for different application areas (software systems [6], in-service systems [4], aerospace and defense systems [7], building renovation [8]), proposing very diverse terminology (Fig. 1) and methodology.

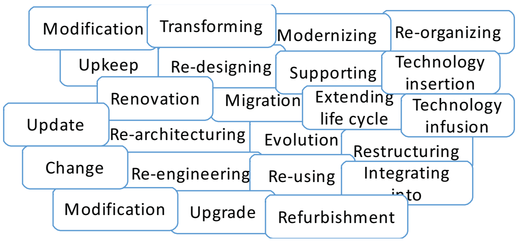

Fig. 1. Diversity of existing terminology

Currently, there is no global agreement on terminology: such terms as modification, modernization, retrofit, upgrade, update, enhancement and change are used mainly based on areas of knowledge (field-specific jargon) or personal author preferences. Lack of coordination in terminology eventually limits also mutual use of methodology in different fields, when new methods and approaches do not circulate among different fields. The problem of loose connection between different fields is common to many design papers [9].

The existing approaches for system modernization can be classified depending on the factor either modernization is conducted as part of operations and maintenance phase of existing system lifecycle or as an independent process of new system design. In particular, [7] states that *a major rebuild should be treated as a new system development*, while dealing with relatively small changes is a part of the system lifecycle.

A. Modification as a part of existing system lifecycle

There is considerable overlap between change management, change control and configuration management [10]. INCOSE considers change management as a part of configuration management [11].

Change management is the process of requesting, determining attainability, planning, implementing, and evaluating of changes to a system. Important part of change management process is a change request document [10].

Change control is the process to ensure that changes to a system are introduced in a controlled and coordinated manner [12].

Configuration management is the discipline of identifying and formalizing the functional and physical characteristics of a configuration item at discrete points in the product evolution for the purpose of maintaining the integrity of the product system and controlling changes to its baseline [13]. Engineering Change Proposals (ECPs) [13], [14], Engineering Change Notes (ECNs) and Engineering Change Orders (ECOs) [15] are used to describe and initiate updates and modifications to the original system, as elements of configuration management of system lifecycle. ECPs handle both minor changes (Class II), and significant modifications (Class I), affecting cost, warranties, deliveries and data requirements [7].

In the field of software engineering, one the most common modernization approaches is incremental, illustrated in Fig. 2 [16], [17]. [16] considers modernization of large software systems, implemented not in a single deployment, but using a sequence of steps and proposes 12 different strategies, ranging from different combinations of software restructure and componentization. Agile methodologies also support incremental development [18].

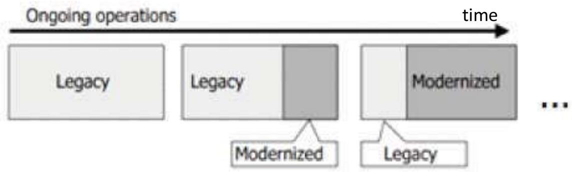

Fig. 2. Incremental approach for system modernization [16]

B. Modification with new system formulation

For major modification of product or service INCOSE proposes to use the same systems engineering processes and principles that are employed during the original design, development, integration, and testing, which require a formulation of a new system and a new system lifecycle [4]. Many authors notice that some of the issues of sustaining existing systems are more problematic than when new systems are realized [3]. Application of systems engineering processes to system modification requires additional tailoring, attention to constraints imposed by the existing system architecture during conceptual phase, and focus on interfaces during synthesis stage [7].

The “Systems Engineering Fundamentals” [7] focuses on large systems and states that upgrading or modifying of existing system is a matter of following the systems engineering process, with emphasis on configuration and interface management, and proposes the list of activities (Fig. 3).

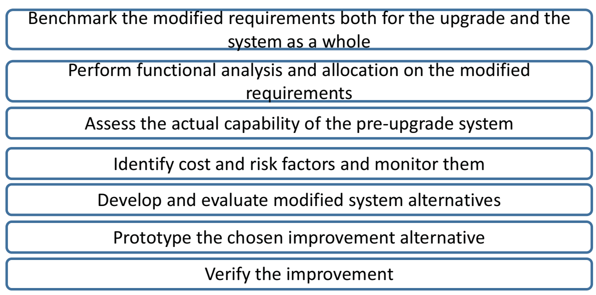

Fig. 3. DoD Framework for system modernization, adopted from [7]

A more general approach for modernization planning for software was proposed by R. C. Seacord and et al. in [6] (Fig. 4). The proposed approach starts with portfolio analysis based on technical quality and business value and concludes with a defined modernization plan or a decision to terminate modernization [6].

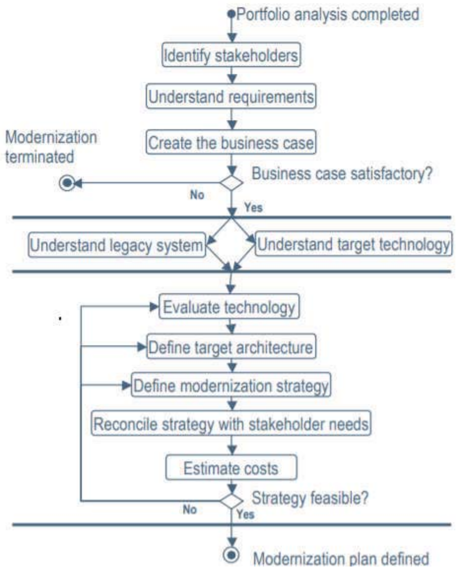

Fig. 4. A systematic framework for legacy program module modernization, adopted from [6]

[6] defines two types of modernization in software:

- White-box that changes the application architecture and the technological stack without losing business value and quality attributes;
- Black-box approach requires knowledge only about system external interfaces;

Besides these two types, [6] and [19] point to the differences in terminology and consider reengineering as a separate type of modernization, including such techniques as retargeting, rewarping, use of commercial off-the-shelf components, and various code optimization techniques (source code translation, reduction, functional transformation).

C. Criteria for modification strategy selection

The above-mentioned strategies for system modernization for every particular situation require consideration and clear understanding of the existing system and various criteria. The following major non-exhaustive criteria can be formulated:

1) The main driver for the modification:

a) routine changes,

b) obsolescence-driven changes,

c) system enhancements for better performance or maintainability,

d) system repurposing

2) Type of the system: product, in-service system (is interruption of operations possible), enterprise;

3) Accessibility to the system (some system e.g. satellites on the orbit have limited physical access);

4) Stage of system life cycle when modification is needed e.g. before or after deployment (modification of the system design, modification of the existing system before deployment or already in-service);

5) The level of available knowledge about the system (white-box or black-box system);

6) Was modification anticipated during the system design;

7) Size of the system (number of elements or interconnections);

8) Size of required modification;

9) Level of complexity of the system;

10) Nature of the system (hardware, software, both);

11) Precedented and unprecedented system (does the system with desired functionality already exist on the market).

D. Transition to a new system definition

A significant amount of the literature is available on the topic of system modification, however even generalized approaches consider very limited options and criteria among those listed above. Meanwhile most of the articles describe particular examples, applications, or phases.

Existing approaches for system modification utilize various paradigms and concepts, have different types of input/output parameters, sometimes not defined clearly, and cover different phases of the system lifecycle. Eventually, it is difficult to formalize, compare and map frameworks on a system lifecycle because of the lack of quantitative characteristics and not clearly stated prerequisites.

The straightforward application of Systems Engineering methods requires a re-formulation of the system and clear understanding how other inputs, including such instances as stakeholders and their needs might evolve. In general, little attempts are made in the literature to give a recipe for the first steps of formulation of a new system – inputs for systems engineering acquisition phase. Richard Wise and et al. in [5] states this problem, calling this process architecture re-definition, and proposing a methodology to extract requirements from the old system to construct a new system, using SysML.

The process of new system formulation also includes formulation of modification strategy, including identification which functionality and elements need to be added or replaced.

This paper proposes measures to characterize process of system modification and a framework for the acquisition phase of new system lifecycle and helps to re-architecture existing system.

III. MEASURES TO CHARACTERIZE SYSTEM MODIFICATION PROCESS

The process of system modification can be formalized and classified using Set Theory, commonly used to describe and connect various fields [20]. This paper proposes to consider the original system and modified system in functional, performance and physical domains. The relation between different metrics can be used to give characterization. If none of these domains has changes, no modification was made. These metrics will eventually be used for the proposed framework to identify alternative design solutions and to facilitate formulation of the definition of the new modified system in the physical artifact domain.

A. System modification in physical (artifact) domain

In the physical domain, we can introduce a set of elements to describe our system of interest. The selection of abstraction level depends on modification objectives. Different tools, such as for example Design Structure Matrix (DSM) [21], [22], SysML [23] or OPM [24] can be used for system decomposition and identification of elements.

The set of system elements S_0 will correspond to the original systems:

$$S_0= \{S_1, S_2, \ldots, S_i, \ldots, S_n\} \quad (1)$$

In case of a retrofit process, the existing set of system elements S_0 is not affected by modification, instead new set system elements ΔS may be introduced allowing the realization of the targeted functionality. This approach is used due to different reasons, such as Intellectual Property protection requirements, or in case of infeasibility of physical access to the original system (in-service system or physically unreachable system). We can refer to the modified system as S_m and the elements:

$$S_m = \{S_1, S_2, \ldots, S_i, \ldots, S_n, S_{n+1}, .. S_{n+j}, \ldots S_{n+k}\} \quad (2)$$

$$\Delta S = \{ S_{n+1}, .. S_j, \ldots S_k\} \quad (3)$$

$$\Delta S, S_0 \in S_m \quad (4)$$

Reengineering includes transformation of the system and its elements into a new system, including displacement and replacement elements of set S_0. S_0* would represent the same set of elements as in the original system, but structured according to a different pattern. Compared to the retrofit, the system is considered as a white-box and this type of modification appears to be a pure example of reengineering.

$$S_m = S_0* = \{S_1, S_2*, S_3* \ldots, S_i, \ldots, S_n\} \quad (5)$$

Then we can describe the process of modification based on relation between S_m and S_0.

In reality, most of the cases of system modifications have elements of both reengineering and retrofit. Use of both approaches can help to define the new system description before applying systems engineering approach for new system design. Fig. 5 uses Venn diagram to visualize the difference between retrofit and reengineering scenarios formalized by equations (1-5).

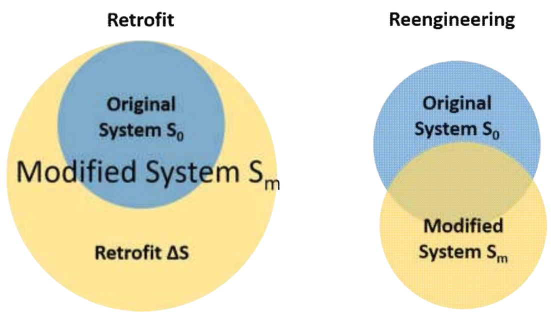

Fig. 5. Types of system transformation from structural point of view

B. System modification in the functional domain

In the functional domain we can define the sets F_{S0}, F_{Sm}, F_{So*}, $F_{\Delta S}$, which correspond to sets in physical domain S_0, S_m, S_0*, ΔS, and an additional set ΔF, which represents the emergence of functionality not directly related to particular elements [25]. Adding new components can bring new desired, undesired or neutral functionality, which needs to be considered during system modification.

Functional decomposition should follow criteria for functional independence, in particular, functions at the same level should have no interdependencies [26]. The level of decomposition should also correspond to structural decomposition.

Similar to the physical domain, the addition of new functionality with permanent original functionality can be defined as a retrofit:

$$F_{Sm} = \{F_1, F_2, F_3*, \ldots, F_n, F_{n+1}, \ldots F_{n+k}\} \quad (7)$$

Change of functionality, which corresponds to the original system, can be defined as system repurposing:

$$F_{Sm} = F_{S0*} \quad (8)$$

Using set theory the scale of modification can be evaluated in functionality domain via set cardinality (number of set elements), for example, $|F_{\Delta S}| > |F_{S0}|$ can be considered as a new system design, opposed to modification of existing system.

The relation between such terms as system restructuring, retrofit and repurposing can be visualized by its allocation in corresponding quadrants, using proposed measures (Fig. 6). In real applications, the modernization process usually is a combination of retrofit, reengineering and repurposing.

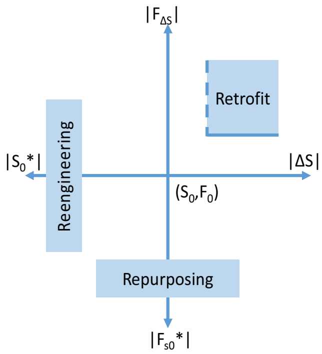

Fig. 6. Connection between functional and physical domain

C. *System modification in performance domain*

Another domain for consideration is system performance. A particular interest is to characterize relation between performances of the original system and the modified system, as well performances corresponding to retrofit and reengineering. In case of performance, numerical evaluation can be used.

For simplicity, different Figures of Merit (FoMs) can be described as an overall performance or customer value. The overall performance of the modified system P_{Sm} is connected to the functional domain and can be defined through nominal performance of the original system P_{S0}, increment in performance due to retrofit $P_{\Delta S}$, performance of the re-engineered original system P_{S0*} and emerging performance $P_{\Delta F}$, corresponding to F_{S0}, $F_{\Delta S}$, F_{S0*} and ΔF. Then for the retrofit case, overall performance can be defined as:

$$P_{Sm} = P_{S0} + P_{\Delta S} + P_{\Delta F} \qquad (9)$$

The quantitative evaluation of proposed performance characteristics might be conducted through surveys and interviews of stakeholders and experts.

Performance measures allow to distinguish between system modernization and system maintenance, aimed to deal with aging and degradation of system elements. Original or nominal performance P_0 and desired performance P_D, by definition, have following relations:

$$P_{Sm} > P_0;\ P_{Sm} <= P_D \qquad (10)$$

The difference between the supporting activities, aiming to maintain system performance during operations phase such as upkeep and refurbishment and modernization activities, aiming to increase performance and achieve desired performance P_D such, as upgrade and update, is visualized using proposed measures in Fig. 7.

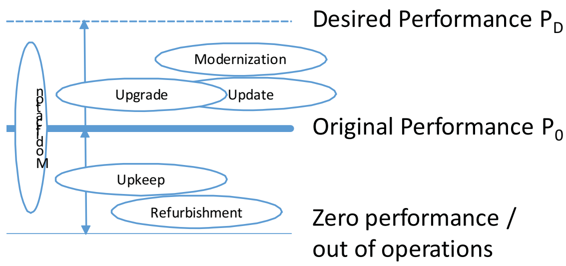

Fig. 7. System modification – relation to performance

Eventually we can allocate relations and connections between different terms based on performance and functionality metrics (Fig. 8). Directions of arrows give general indication on the impact of each process.

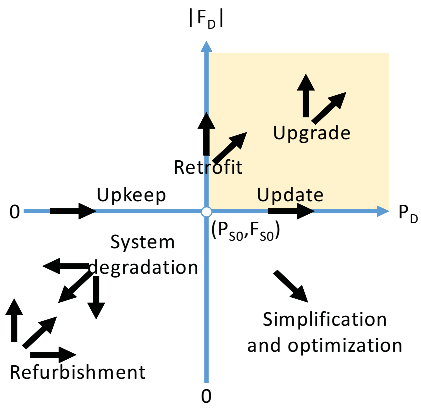

Fig. 8. Relation between different terms in function-performance domain

The above-proposed measures are used in the next section, where a formalized approach to choose the right scenario (reengineering or retrofit) is proposed.

IV. A Proposed Framework

The proposed framework corresponds to the pre-acquisition phase and partially the conceptual stage of acquisition phase of system lifecycle [3]. It aims to re-architecture existing systems and formulates alternatives of modified architectures. The framework consumes the definition of the existing system S_0, general business needs and figures of merit (FoMs).

The framework defines the possible architectures of system described as sets of elements ΔS or S_0*, with corresponding functionalities. The framework can also determine that it may not be feasible or cost-effective to proceed to acquisition phase of new system design, for which pre-acquisition phase is intended for [3].

The framework includes decomposition and analysis of the existing system in physical, functional and performance domains, identification and evaluation of possible alternatives and new system formulation (synthesis).

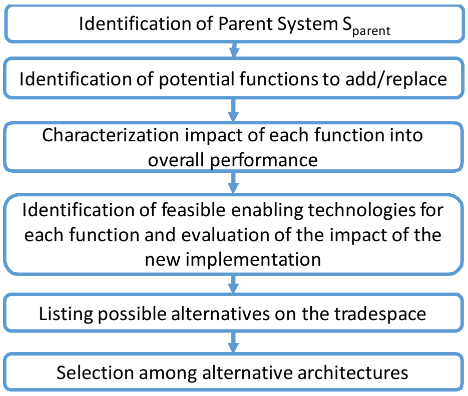

Fig. 9. A proposed Framework

To illustrate the framework a relatively simple case of car retrofit with cruise control functionality was chosen. After its invention in the 50s, classical cruise control has become a standard in the most of modern cars [27]. More advanced adaptive cruise control is common for cars since the 90s [27]. Retrofit kits for both types of cruise control are commercially available on the market today. A logical evolution of cruise control function is a fully autonomous car. Prototypes of cars with auto-piloting are widely tested nowadays. Several companies develop retrofit kits for Autonomous Driving [28].

A. Definition of a parent system

The clear understanding of the original system and its borders is crucial for modernization [29]. The first step of the framework aims to decompose existing system into elements, formulate set S_0, and define the corresponding parent system, sometimes called accompanying or using system. DSM is proposed as a tool for characterization and identification of the interactions with the environment. The selection of the abstraction level and borders for the parent system depends on stakeholder objectives.

In the case of car retrofitting, the parent system will include car, driver and elements of the environment. Figure 10 shows an example of DSM for parent system for car retrofitting case: some elements are not represented for simplicity.

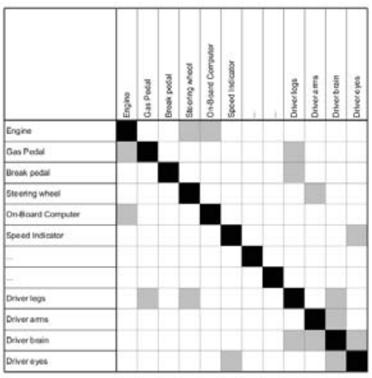

Fig. 10. Parent system for car

Using DSM also allows conducting a quantitative comparison of different systems. Different values might be assigned to interfaces to give a characterization of the system. For example, the ease of changes in the system might be characterized in a similar manner, as system re-configurability [30], using numeric DSM [21]. In particular, each interconnection of elements needs an assigned value, which depends on type of interface and ease of making changes. Then the DSM can be represented as numerical matrix and quantitative characteristics might be computed. Eventually, in this way, different systems might be compared in terms of "flexibility" for changes.

Next steps will define which elements of S_{parent} need to be replaced according to business needs and FoMs.

B. Functional decomposition of S_{parent} and analysis

The aim of this step is to decompose the parent system in functional domain. Such tool as functional tree [31] can provide required decomposition on different levels to identify original functionality F_{S0}, potential functionality $F_{\Delta S}$ and emerged functionality ΔF.

In particular for the car cruise control retrofit case such functionalities are performed by the driver as checking speedometer F_5, comparing speed with the limit F_6, taking decision and action to press gas pedal F_7 will form a potential retrofit set $F_{\Delta S}$. Fig. 11 shows an example of functional tree corresponding to S_{Parent} for car retrofit case. The lowest level is represented by functions F_1 … F_n. More advanced adaptive cruise control retrofit case would include several additional functions such as measurement of the distance between objects ahead F_3, emergency braking F_4 and etc. Even more advance case of retrofitting existing car with autonomous driving capabilities consist of replacement of other functions which driver performs today, such as navigation and identification of available space for road maneuvering.

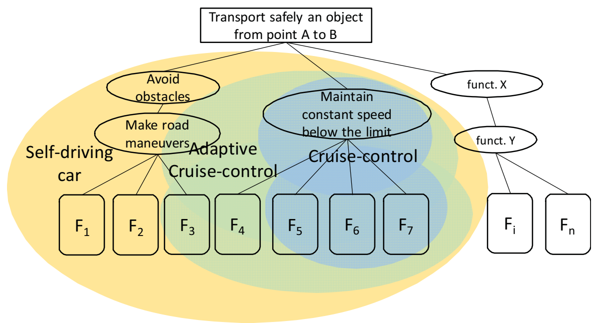

Fig. 11. Functional tree diagram for S_{parent}

The next step after decomposition of S_{parent} in functional domain is to evaluate impact of each identified function to overall performance.

C. Performance characterization

For performance characterization, different FoMs need to be correlated with overall performance. One of the approaches is to use a weighted-sum method [32]. This approach requires close cooperation with customer and experts, use of surveys and interviews to define weights. The impact of functions on the overall performance can be described:

$$P = w_1P_1 + w_2P_2 + \ldots + w_iP_3 + \ldots + w_nP_n, \quad (11)$$

where P_1 ... P_n – figures of merit and w_1 ... w_n are corresponding weights.

Then for each potential function, identified at the previous step, its impact on FoMs and overall performance needs to be evaluated. For example in the case of passenger car, functions of comparing the speed F_6 and making decision of pressing gas pedal F_7 affect such FoMs as safety of driving, driver satisfaction, fuel consumption and car emission rate.

The replacement of higher-level function of speed maintenance (F_5, F_6, F_7) performed by human to automatic for some models of cars can save from 7% to 30% of fuel consumption on hills [33]. More advanced functions as Eco-cruise control save even more fuel, but increase travel times [34] and eventually negatively affect driver satisfaction.

D. Identification and evaluation of enabling technologies

The aim of this step is to identify potential enabling technologies to replace functions of the parent system. Technology Readiness level (TRL) can be used to evaluate feasibility [35]. Each function-technology pair needs to get corresponding values of impact on FoMs, overall performance and cost of implementation.

The relation between physical, functional and performance domains allows identifying which particular elements eventually need to be replaced and which additional customer value can be brought to the original system.

Different methods, such as technology infusion framework [22] and Real Options Analysis [36], can be used to evaluate customer value of changing technology for particular function and its corresponding cost.

For example, historically, two major technological solutions for car cruise control were classical mechanical controls and fuzzy logic controls [27]. In case of retrofit this two options need to be characterized in terms of performance.

E. Formulation of alternatives for system modernization

At this step alternative modernization options, described as different combinations of replacement and changes of the existing elements and functionality and addition of new, can be formulated as new systems ΔS or S_0*. The identified modernization alternatives can be distinguished as reengineering and retrofit options.

Modernization alternatives can be visualized using tradespace exploration paradigm in bi-dimensional space [37] with customer value and cost of modernization on x- and y-axes.

The following process of selection of an alternative is comprehensively described in the literature dedicated to tradespace exploration [38], [39].

After the selection of an alternative architecture is conducted by decision-makers, system engineering processes of new system design and development can be applied.

V. Conclusion

The paper gives an overview of the literature in the field of modernization and obsolescence management of existing systems. The work emphasizes a problem of a lack of quantitative tools for description of system modernization and need of developing a formalized supporting methodology to tailor systems engineering processes for system modernization.

The paper presents measures to give a classification to existing terminology and to support a proposed framework for identification of a modernization strategy. The proposed measures and framework are based on characterization of existing and modified system in physical, functional and performance domains using set theory. Different systems engineering tools, such as DSM, functional tree and tradespace exploration were used to support the framework.

The framework was illustrated by the example of car retrofit with cruise control functionality. While the paper used a retrospective analysis for the case study and cruise control retrofit kits are available on today's market, the results of the case study have a potential to be expanded to autonomous vehicles case.

Next steps and future work include validation of the proposed framework on multiple case studies.

Acknowledgment

Rustam Akhtyamov thanks his colleagues Nicola Garzaniti, Armen Poghosyan, Yaroslav Menshenin and Ilya Yuskevich for constructive criticism and fruitful discussions.

References

[1] ACEA, "Vehicles in use - Europe 2017," 2017.

[2] H. Jiang, "Key Findings on Airplane Economic Life," 2013.

[3] I. Faulconbridge and J. M. Ryan, *Systems engineering practice*. 2018.

[4] INCOSE UK Chapter, "Applying Systems Engineering to In-Service Systems," 2010.

[5] R. Wise, L. Sheppard, J. Huggins, and D. Broadwell, "A methodology and heuristics for re-architecting a legacy system," in *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, 2015, pp. 382–389.

[6] R. C. Seacord, D. Plakosh, and G. Lewis, *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*. Addison-Wesley, 2003.

[7] B. Lightsey, *Systems Engineering Fundamentals*, no. January. Fort Belvoir, Virginia: DEFENSE ACQUISITION UNIVERSITY PRESS, 2001.

[8] E. Fregonara, V. Lo Verso, M. Lisa, and G. Callegari, "Retrofit Scenarios and Economic Sustainability. A Case-study in the Italian Context," *Energy Procedia*, vol. 111, pp. 245–255, Mar. 2017.

[9] L. Blessing and A. Chakrabarti, "DRM: A Design Reseach Methodology," in *Springer London*, no. September, London: Springer London, 2009, pp. 13–42.

[10] J. McCalman, R. Paton, and S. Siebert, *Change Management, A guide to effective implementation*. 1992.

[11] INCOSE, "Guide for the Application of Systems Engineering in Large Infrastructure Projects," no. June, pp. 1–55, 2012.

[12] P. Hall and J. Fernandez-Ramil, *Managing the software enterprise : software engineering and information systems in context*. Thomson Learning, 2007.

[13] M.-H. DoD, *Configuration Management*, no. 1. 2001.

[14] J. S. Lee and L. E. Miller, *INCOSE - Systems Engineering Handbook*. 2012.

[15] F. J. Buckley, *Implementing Configuration Managment, Hardware, Software and Firmware*. IEEE Computer Society Press, 1996.

[16] R. C. Seacord, S. Comella-Dorda, G. Lewis, P. Place, and D. Plakosh, "Legacy System Modernization Strategies," 2001.

[17] T. C. Fanelli, S. C. Simons, and S. Banerjee, "A Systematic Framework for Modernizing Legacy Application Systems," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 678–682.

[18] J. V. Sutherland, *Scrum : the art of doing twice the work in half the time*. 2014.

[19] K. Kowalczyk and A. Kwieci, "Model-Driven Software Modernization," no. November, 2009.

[20] J. Ferreirós, *Labyringth of Thought: A History of Set Theory and its Role in Modern Mathematics*. Birkhäuser Verlag, 2007.

[21] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. MIT Press, 2012.

[22] E. S. Suh, M. R. Furst, K. J. Mihalyov, and O. De Weck, "Technology infusion for complex systems: A framework and case study," *Systems Engineering*, vol. 13, no. 2, pp. 186–203, 2010.

[23] S. Friedenthal, A. Moore, and R. Steiner, "A Practical Guide To SysML: The Systems Modeling Language," *Systems Engineering*, 2015.

[24] D. Dori, *Object-Process Methodology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[25] E. Crawley, B. Cameron, and D. Selva, *System architecture : strategy and product development for complex systems*. .

[26] N. P. Suh, "Axiomatic Design Theory for Systems," *Research in Engineering Design*, vol. 10, no. 4, pp. 189–209, Dec. 1998.

[27] A. K. Shaout and M. A. Jarrah, "Cruise control technology review," *Computers and Electrical Engineering*, vol. 23, no. 4, pp. 259–271, Jul. 1997.

[28] M. Herger, "Overview of Retrofit Kits for Autonomous Driving." [Online]. Available: https://thelastdriverlicenseholder.com/2017/10/21/overview-of-retrofit-kits-for-autonomous-driving/. [Accessed: 28-May-2018].

[29] M. W. Maier and E. Rechtin, *The Art of Systems Architecting*. CRC Press, 2000.

[30] A. M. Farid and D. C. McFarlane, "A Design Structure Matrix Based Method for Reconfigurability Measurement of Distributed Manufacturing Systems," *International Journal of Intelligent Control and Systems*, vol. 1, no. 1, pp. 118–129, 2007.

[31] N. Viola, S. Corpino, M. Fioriti, and F. Stesi, "Functional Analysis in Systems Engineering: Methodology and Applications," in *Systems Engineering - Practice and Theory*, InTech, 2012.

[32] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, Jun. 2010.

[33] J. N. Hooker, "Optimal driving for single-vehicle fuel economy," *Transportation Research Part A: General*, vol. 22, no. 3, pp. 183–201, May 1988.

[34] S. Park, H. A. Rakha, K. Ahn, and K. Moran, "Fuel Economy Impacts of Manual, Conventional Cruise Control, and Predictive Eco-Cruise Control Driving," *International Journal of Transportation Science and Technology ·*, vol. 2, no. 3, pp. 227–242, Sep. 2013.

[35] J. C. Mankins, "Technology Readiness Levels," *White Paper April*, p. 5, 1995.

[36] R. De Neufville, S. Scholtes, and T. Wang, "Real Options by Spreadsheet: Parking Garage Case Example," *Journal of Infrastructure Systems*, vol. 12, no. 2, pp. 107–111, Jun. 2006.

[37] A. M. Ross and D. E. Hastings, "The tradespace exploration paradigm," *INCOSE International Symposium 2005*, vol. 15, no. 1, p. 13, Jul. 2005.

[38] C. A. Mattson and A. Messac, "Concept Selection using s-Pareto Frontiers," *AIAA Journal*, vol. 41, no. 6, pp. 1190–1198, 2003.

[39] G. S. Parnell, *Trade-off Analytics: Creating and Exploring the System Tradespace.* Wiley, 2016.