Reliability Requirements Engineering in Socio-Cyber-Physical System Context

Ksenija Lace and Marite Kirikova

Riga Technical University, Latvia
ksenija.lace@rtu.lv, marite.kirikova@rtu.lv

Abstract. One of the system quality characteristics is system reliability, which defines the degree to which a system, a product or a component performs specified functions under specified conditions for a specified period of time. System reliability significantly impacts also other system quality characteristics, such as performance and usability, and often is the key factor impacting overall quality of the system. With emerging new generation of cyber systems, where three dimensions – socio, cyber and physical are tightly linked together in order to achieve common goals or solve common problems, system reliability became even more important. This paper proposes the approach for reliability requirements engineering in the context of SCPS. The approach integrates Failure Mode Effects Analysis and Morphological Analysis in reliability requirements engineering, for the in-depth multi-dimensional analysis of potential failure scenarios.

Keywords: Socio-Cyber-Physical Systems, Reliability, Maintenance, Downtime, Deployment, Software Upgrading, Maintenance automation.

1 Introduction

Socio-cyber-physical systems (SCPS) are complex real-time systems of systems, which have very high expectations for reliability. At the same time, achieving reliability in SCPS is a very difficult task due to several reasons – high uncertainty, different nature of system elements, emergent system behavior, and many interdependencies between system components. This requires much more comprehensive analysis for reliability requirements then is performed in traditional requirements engineering [1].

Reliability research has dramatic importance, due to the following factors [1, 2, 3]:

- Reliability expectations dramatically increased during the latest decade.
- The complexity of developed systems is leading to the high level of uncertainty, meaning the potential risk of lower system reliability.
- Development projects usually have limited resources, including time and budget, that might again lead to decreased reliability.

Despite the fact that reliability engineering has evolved during last decades and has comprehensive research, and wide range of available techniques for reliability predic-

tion and failure analysis, there is still no common methodology for holistic reliability engineering process [2].

In both, reliability engineering and requirements engineering areas, not much research is focused on reliability requirements perspective specifically, and there is no research in the field of integration of reliability requirements engineering and reliability engineering disciplines [1, 3].

Much deeper integration of requirements engineering and reliability engineering activities can be a possible solution.

2 Research Method

In the scope of this research, the following activities were executed:

1. Investigation of existing research in the field of reliability requirements. The main focus was on research covering reliability for real-time systems, reliability of physical and social systems, and managing reliability in high uncertainty.
2. Identification of challenges for reliability requirements engineering in SCPS context, comparative analysis of existing reliability requirements engineering approaches from the perspective of their applicability in the context of SCPS challenges.
3. Investigation of existing research in the field of morphological analysis. The main focus was on approach applicability for process modeling, software requirements analysis, and complex systems analysis.
4. Proposing the approach for reliability requirements engineering, combining reliability engineering techniques with morphological analysis for multi-dimensional failure analysis.
5. Practical application of the proposed approach for the SCPS example.
6. Building conclusions about applicability of the proposed approach and defining possible further steps of research.

3 SCPS Reliability Requirements Engineering

The first official definition of reliability was stated in 1957, by Advisory Group on the reliability of Electronic Engineering (AGREE). According to this definition, reliability is the probability of a product performing a specified function without failure under given conditions for a specified period of time [4].

Two reliability standards, commonly used today – ISO and IEEE, have similar definitions of reliability:

- According to the IEEE standard, focusing on the software reliability, the reliability is the probability that software will not cause the failure of a system for a specified time under specified conditions.

- According to the ISO standard, focusing on more generic system level, the reliability is the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

These definitions can be used also for reliability of SCPS. Usually, required reliability is described in the format of reliability requirements [5]. From the stated reliability definitions, reliability requirements should provide the following:

- The definitions of system functions, for which reliability is defined.
- Conditions for function execution.
- Definitions of function successful execution or function failure, depending on the reliability metrics.

Reliability requirements should be based on the facts (real data) and should be focused on the most critical reliability aspects (real need) [1].

Reliability requirements should be a prerequisite to any reliability engineering activities. Quality of reliability requirements has direct impact on the reliability of the system. But unfortunately, there are several common reliability requirements problems, which may make it harder to use reliability requirements for reliability engineering activities later in the project [5 – 8]. Many of them can be addressed using traditional requirements engineering techniques. However, the problem that Reliability requirements are **too generic** and cannot be traced to the requirements implementation strategies, cannot be solved in the scope of traditional requirements engineering as it focuses on specifying reliability requirements, not the in-depth analysis of the potential issues for achieving desired reliability. This problem can be solved through integrating reliability engineering techniques into requirements engineering, which would link together requirements and implementation activities [1, 9].

However, using existing reliability engineering techniques for SCPS might have several challenges due to SCPS complexity, diversity and emergency [1, 4, 10]:

- Techniques do not take into account the **nature of system component** – SCPS socio, cyber and physical parts might need different approaches.
- Failure **impact assessment and failure prioritization** is not fully supported in the existing techniques; however, due to high complexity of the SCPS, addressing all possible failures is just unrealistic. This is why impact assessment and focus on the failures with highest negative impact should be the key point in SCPS reliability.
- **Historical failure data** is a prerequisite for reliability prediction – SCPS adapt and emerge, as well as human and software reliability cannot be so well predicted based only on the statistics, and historical data not always will be relevant for the future prediction.
- Approaches are more tended to focus on **separate system components** than on a system as a whole – a SCPS cannot be seen as just a list of components, as relationships and dependencies between components play significant role.
- Existing approaches often are based on the **difficult calculations** and are hard to use – SCPS complexity might lead to the need of assumptions and simplifications, which, in turn, will lead to lower accuracy.

4 Reliability Engineering Techniques

Based on the available research of specific reliability engineering techniques, the most popular techniques were reviewed for potential applicability for reliability requirements engineering [2, 8, 11].

Techniques for reliability analysis can be applied for reliability requirements in-depth failure analysis and connecting with the potential reliability engineering strategies.

In the scope of this research failure mode and effect analysis (FMEA) technique was selected for the integration into requirements engineering activities, specifically, for analyzing possible failure reasons and potential negative impact for specified reliability requirements. FMEA was preferred over fault tree analysis (FTA) technique as it is, in particular, efficient when a large number of different failure scenarios exist within a range of negative impact, not just a specific failure scenario that should be investigated.

FMEA usually consists of seven sequential phases [12 – 15]:

- Detection of possible failure modes for selected system components.
- Evaluate severity for each failure mode.
- Evaluate probability for each failure mode.
- Evaluate existing detection controls for each failure mode.
- Evaluate overall risk for each failure mode and select the ones with the highest risk.
- Determine actions to reduce risk for selected failure modes.
- Take appropriate actions and recalculate risk.

However, as stated before, there are several enhancements in existing reliability techniques, which are required for their successful application in SCPS domain [11, 16 – 19]:

- Technique should cover all three dimensions – socio, cyber and physical ones.
- Technique should be applicable in situations, when there is no historical data available.
- Technique should be able to review and evaluate many possible failure reasons from different possible failure aspects.
- Technique should support the selection of the most efficient strategy, based on the multi-dimensional evaluation.
- Technique should have a clear algorithm, which could be automated.

5 Morphological Analysis

One of the proven approaches for the analysis and modelling complex multi-dimensional systems is morphological analysis (MA). It was initially used for modelling relationships between structural components in different scientific fields like botany, linguistics, geology and mathematics [20]. More abstract version of MA was

proposed by Swiss-American physicist and astronomer Fritz Zwicky. He also started to use it for social problem solving and technological engineering. Nowadays MA is applied in many different cases, and has proved to be efficient for the following purposes [21, 22]:

- Developing alternative strategies.
- Assessing organizational readiness for different goals.
- Developing possible scenarios.
- Assessing possible risks.
- Analyzing cause-sequence relationships.
- Presenting complex situations in a more easy-to-understand format.

MA is very effective method for analyzing complex problems from different possible aspects. It also supports generation of various possible solutions for these problems [23].

MA can be used in system engineering during two main activities – (1) Building general model of a problem space and (2) Generating possible system solutions for addressing the problem. MA also supports innovation engineering, as, through this method, new previously unknown aspects can become visible, which stimulates new ideas for the solution space [24].

In general, morphological analysis is the process of sequential analysis and synthesis activities with a purpose of exploring different aspects of the specific complex, non-quantified problem and identifying all possible solutions [20]. During analysis step, the problem is structured using problem describing parameters and possible parameter values. During synthesis step, the values of different parameters are grouped into possible configurations, and resulting configurations are assessed from the aspect of probability. Configurations, that are not realistic, are excluded [21, 25].

In the scope of engineering, morphological analysis should include also some additional steps – (i) As a first activity – the problem should be formulated as precisely as possible and (ii) As a final activity – remaining realistic configurations can be then assessed from the selected perspective and the best configuration(-s) selected (for instance, all configurations supporting specific value for specific parameter, are identified) [21, 26].

But like any other method, MA has its own disadvantages. The main constraint for the application of this method is the workshop format, where highly motivated, knowledgeable system-thinking participants are working together. Depending on the complexity of the problem studied, MA also can be quite time consuming and requiring decent automation level [21].

In scope of this research, MA is integrated into FMEA approach for in-depth multi-dimensional analysis of possible failure reasons for system components which can affect defined reliability requirements. Additionally, MA is used on the possible failure negative impact minimization analysis level – for the definition of possible impact minimization strategies. This is not the traditional application of MA, when potential solutions are generated based on the possible configurations, but rather might be interpreted as the structuring method for failure negative impact problem space and assessing efficiency of potential strategies for improving the negative impact.

6 SCPS Reliability Requirements Elicitation Process

Reliability requirements elicitation process is organized as four related phases [1, 5, 14]:

1. Reliability requirements initial definition – during this phase functional requirements and appropriate reliability requirements are elicited and documented using standard requirements engineering activities. After that, for selected functional requirements, supporting system components are identified. At the end, the most critical components are selected for further analysis.
2. Failure mode analysis – in this phase, for each selected component, possible failure reasons are analyzed. Failure mode with the highest potential negative impact is selected for further analysis. Selected reason is divided into possible specific sub-reasons and for each sub-reason; the most possible negative failure impact is evaluated. The sub-reason with the highest possible negative impact is selected for the next phase.
3. Failure minimization strategy – during this phase possible solution strategies are generated for previously defined and assessed failure sub-reason with highest negative impact. The strategies improving failure mode parameters are selected for the next phase.
4. Reliability requirements detailed definition – based on the selected solution, initial reliability requirements are detailed with possible failure scenarios and selected strategies. If initially defined reliability cannot be achieved, reliability requirements are redefined.

In the following sub-sections each phase is described in more detail. In Fig.1 can be seen how exactly proposed method is organized by adjusting traditional FMEA and MA, and then combining together adjusted methods. MA is integrated into FMEA on the failure mode analysis phase, which is not very strictly defined for FMEA. MA however helps to evaluate failure modes through different perspectives and generate all possible failure mode configurations we should consider.

Proposed methodology is illustrated by the example study – Live Intelligent Tutoring System. The main goal of intelligent tutoring system is to create intelligent agents in cyber space that can teach students like live tutors – be autonomous and adapt to each student needs. [27]. But Intelligent Tutoring System for corporate training should be considered as the socio-cyber-physical System, having several additional characteristics:

- Socio space – training organization and organization where employees will use gained knowledge and skills should be considered as parts of a system.
- Physical space – physical equipment is used for training or training is aimed to teach how to use specific physical equipment.
- Cyber space – should have representations of both socio and physical spaces.

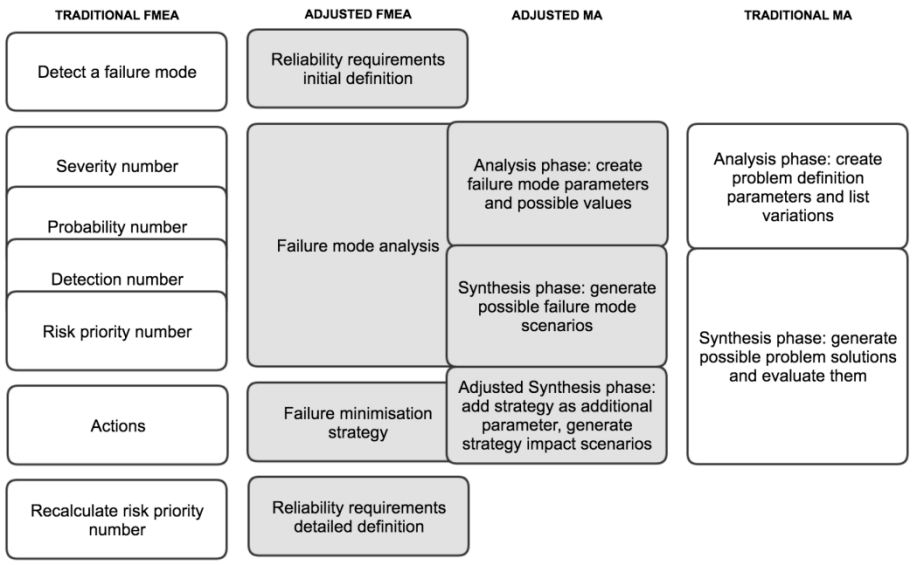

Fig. 1. Combination of FMEA and MA processes.

6.1 Reliability Requirements Initial Definition

In this phase reliability requirements should be defined for the system as reliability requirements for critical system functions.

After reliability is defined on function level, system should be structured into sub-systems, which support specific system functions. Each sub-system should be presented as a group of different cyber, physical and socio elements:

- Cyber element – can be software system, system module or specific service.
- Physical element – can be computer hardware, different equipment or equipment parts.
- Socio element – can be humans operating with physical elements, using or supervising cyber elements.

Table 1 can be used for describing solution space. In each column all system functions, socio, cyber and physical components are listed. After all components and functions are listed, for each function supporting components are identified, and for each supporting component importance (I) is defined using 1 (required very rare) ... 10 (component is critical) scale – 1 to 10.

Overall component significance (S) is calculated using the following formula (n – number of functions supported by component):

$$S = \sum_{i=1}^{n}(I_i) \quad (1)$$

Table 1. Solution space

Function	Socio components					Cyber components								Physical components			
	Tutors	Customer support	Video engineers	System administrators	IT administrators	System student interface	Lecture schedules	Course materials	Student profile	System tutor interface	Customer Support System	Video management system	Data exchange mechanisms	Devices used by students	Hardware equipment used for cyber systems	Physical tutoring equipment	Video equipment
Watch broadcasted video	10	3		3	3	9	8			9	3	9	9	9		5	9
Pass a test		3		3	3	9	6	8	9		3		9	9	9		
Ask questions	10	3	3	3	3	9	6	9	9		3		9	9	9		
Significance	20	9	3	9	9	27	20	17	18	9	9	9	27	27	18	5	9

As the next step, for each user requirement supporting system components are identified.

6.2 Failure Analysis

In this phase, for each selected component, the possible failure reasons are defined. For this MA approach is applied, using the following parameters for the definition of different possible failures:

- Reason – different types of failure reasons for specified cyber/physical/socio elements. For instance – software/hardware failure, maintenance or upgrade, human related accident.
- Scale – amount of other components that failure will affect. Some generic values can be one, some, many, and all. Can be also a specific number.
- Length – different possible failure affect length intervals. For instance: <5, 5-10, 10-30, 30-60, >60 minutes.
- Frequency – failure incidents frequency in specific time interval. For instance – x times in year, month, week, day, and hour.
- Recovery effort – different levels of required recovery effort during and after the failure - almost none, low, medium, high, extremely high.
- Impact level – different levels of failure impact on key system efficiency metrics – almost none, low, medium, high, extremely high.
- Control level – different levels of how well this failure reason can be controlled by system internal mechanisms; or this reason is external and cannot be impacted. Some possible generic values can be % of control.

- Process maturity – different maturity levels of existing failure handling process – automatic, semi-automatic, fully manual, not formalized, not existing.

Defining specific values for each of these parameters the space of all possible failure aspects is created – each column in the Table 2 contains all possible values for the specific parameter. Note, that some of parameters can have more possible values and some less; at this point each parameter values are completely independent from other parameters.

Table 2. The space of failure aspects

Nr	Reason (R)	Occurrence			Severity		Detection	
		Scale, users (S)	Length, min (L)	Frequency (F)	Recovery effort (R)	Impact level (I)	Control level (C)	Process maturity (M)
1	Software error	One	<5	Monthly	Almost none	Almost none	100%	Auto
2	Viruses	Some	5-30	Weekly	Low	Low	50%	Semi auto
3	Maintenance	Many	>30	Daily	Medium	Medium	0%	Manual
4					High	High		

As the next step a cross-consistency check for all defined failure aspects is performed. In Table 3 all possible parameter values are listed both in columns and in rows, and then each possible pair of parameter values is evaluated from the two perspectives:

- Definition of interrelated parameters - several parameters are always directly interconnected, that means that changing values for one parameter will directly influence value of another parameter – for instance, failure length affects impact level. Related parameter values should be highlighted using specific color in the table.
- Definition of not consistent values – for interconnected parameters several values are not consistent, that means that these values cannot coexist and are exclusive – for instance, long failure period and small failure impact. Inconsistent parameter values should be marked in the table using "X" or similar symbol.

Note, that parameter "Reason" is not included in rows and parameter "Process maturity" is not listed in columns; – this is not needed, as we are not evaluating pairs of values for the same parameter, only for different parameters. Due to space limitations, instead of full names of parameters and values in two first columns just first letters of parameters and numbers of values are listed.

Based on the cross-consistency check results, for each reason, number of immutable impact parameter values with highest negative impact is counted (this means that a specific reason cannot lead to the highest negative impact). The reason with the lowest number is the one with highest possible impact.

For the selected reason different possible failure sub-reasons are identified and for each of them the appropriate failure mode is generated as a configurations of different parameter values, where this reason can be a failure cause.

Table 3. A matrix of parameter values

		Reason			**Scale**			**Length**			**Frequency**			**Recovery**				**Impact**				**Control**		
		1.User error	2.Software bug	3.Maintenance	1.One	2.Some	3.Many	1.<5	2.5-30	3.>30	1.Monthly	2.Weekly	3.Daily	1.Almost none	2.Low	3.Medium	4.High	1.Almost none	2.Low	3.Medium	4.High	1.100%	2.50%	3.0%
S	1																							
	2																							
	3	X		X																				
L	1		X	X																				
	2			X																				
	3	X																						
F	1	X																						
	2			X																				
	3		X	X																				
R	1		X				X			X			X											
	2									X			X											
	3			X																				
	4	X		X	X																			
I	1		X				X			X			X			X	X							
	2																							
	3			X																				
	4	X		X	X																			
C	1	X														X	X			X	X			
	2			X																				
	3		X	X																				
M	1	X	X							X	X						X				X			X
	2																							
	3	X		X																		X		

Each configuration is evaluated using the following criteria:

- Occurrence (O) as a probability of failure occurring to the specific reason. Occurrence is rated on a scale from 1 to 10, based on the scale, length and frequency parameters. 1 is extremely unlikely, 10 is inevitable.
- Severity (S) defines how serious the impact of the failure is. Severity is rated on the scale from 1 to 10, based on the recovery and impact parameters. 1 is insignificant, 10 is catastrophic.
- Detection (D) defines how well the failure can be detected before any impact. Detection is rated on a scale 1 to 10, based on the control and maturity parameters. 1 means the control is absolutely certain to detect the failure, 10 no control exists.
- Overall failure risk (R) is calculated using the following formula:

$$R = O \times S \times D \tag{2}$$

Evaluation is made manually, based on the forming parameter values. Configuration with the highest possible negative impact is selected for further analysis. In our example, in the Table 3 all immutable parameters are highlighted in red, and the smallest number of such "red" cells has the reason "Software bug". It is important to mention, that the process should be iterative, and after selected configuration is analyzed and possible impact minimization strategies selected, the negative impact should be recalculated and next configuration with the highest negative impact should be selected. The process continues until the configuration with the highest impact will be the one, which was already analyzed.

Next step is to identify and evaluate all possible failure modes in case of the reason "Software bug", taking into account immutable values for other parameters (for instance, in the example "Low Impact" is immutable with "Many affected components" and "High recovery"). As there can be too many possible combinations, we need to divide "Software bug" reason into more specifics reasons; and for each of them to evaluate negative impact. The following "Software bug" more specific reasons are selected:

1. Software is not compatible with user device.
2. Software is not tested.
3. Change in one software part broke another software part.
4. Not compatible authentication service.

6.3 Failure Minimization Strategy

In this phase, for the selected failure reason with maximal possible negative impact minimization strategies are generated. Some possible approaches for choosing the strategy can be:

1. Prediction approach – failure prediction and addressing in preventive manner.
2. Quality approach – quality of system operation and produced output. Minimization of the risk that it will make system failures more frequent and more difficult to recover from.
3. Proactive maintenance approach – system health monitoring and maintenance in advance.
4. Recovery efficiency approach – efficiency of recovery process after failure occurred, minimization of over processing and not required actions and motions.
5. Backup utilization approach – different backup resources for usage during system recovery after failure.
6. External services and vendors availability approach – efficiency of external services handling, external resources reliability.
7. Inventory approach - waiting/wasted resources during recovery, resources utilization during recovery to minimize related costs and effort.

After that each possible strategy should be assessed from the point of the efficiency in the selected failure mode. Strategy is added as one additional parameter for failure mode space definition. Using cross-consistency matrix (see Table 4), the impact of

each strategy on the each failure mode parameter is evaluated. The combination of strategies affecting as many as possible parameters, is selected for the next phase.

Table 4. Failure mode parameter – strategy cross-consistency matrix

		Scale			**Length**			**Frequency**			**Recovery**				**Impact**				**Control**			**Process Maturity**		
		1.One	2.Some	3.Many	1.<5	2.5-30	3.>30	1.Monthly	2.Weekly	3.Daily	1.Almost	2.Low	3.Medium	4.High	1.Almost	2.Low	3.Medium	4.High	1.100%	2.50%	3.0%	1.Auto	2.Semi-auto	3.Manual
S	1	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X		X
	2	X	X	X	X		X		X	X	X	X	X	X	X	X	X	X	X		X	X		X
	3	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X		X
	4	X	X	X	X		X	X	X	X	X	X		X	X	X		X	X		X	X		X
	5	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X		X	X		X
	6	X	X	X	X		X		X	X	X	X	X	X	X	X	X	X	X		X	X		X
	7	X	X	X	X		X	X	X	X	X	X		X	X	X		X	X		X	X		X
S	1													X				X						
	2																							
	3										X				X									
L	1																							
	2																							
	3										X	X			X							X		
F	1																					X		
	2																							
	3										X	X			X									
R	1																							
	2																							
	3														X				X					
	4														X				X			X		
I	1																							
	2																							
	3																		X					
	4																		X			X		X
C	1																							
	2																							
	3																					X		

As in our example we want to select strategies for improvements in all failure mode parameters, two strategies would be sufficient – quality and recovery efficiency. We also can define some specific activities in the scope of selected strategies (for instance, define how exactly quality will be achieved – additional manual testing, automated tests, etc.)

6.4 Reliability Requirements Final Definition

In this phase initially stated reliability requirements are elaborated, and the following information is added to each of them:

- List of components supporting appropriate function.
- For each component the failure reason, sub-reason and failure mode with the highest overall risk is described.
- For each failure the strategy with the highest benefit is described.

If selected strategy cannot guarantee initially stated reliability requirement, requirement should be adjusted.

For instance, first component – **system student interface** had the following methodology application results:

- The failure reason with the most possible negative impact is the software bug, specifically – not compatible authentication service.
- The combination of strategies that can be used for minimization of failure impact are quality and recovery efficiency.

7 Conclusions

System reliability is one of the key quality characteristics of any system, meaning that without the ability to perform system's functions in specific conditions for specific period of time, system cannot be used efficiently. At the same time reliability is very hard to achieve due to high uncertainty about potential system failures during system development. Reliability requirements should form the basis for system reliability engineering, stating exactly what level reliability is required. However often reliability requirements are stated based only on the stakeholders' opinion and are too generic to be transformed into specific system functions or reliability engineering activities.

In the scope of this research the new approach of reliability requirements engineering was proposed, as an integration of existing reliability engineering technique for analysis of potential failure reasons and related negative impact – failure mode and effect analysis (FMEA). This approach allows not only stating reliability requirement as a number of successfully executed system functions per period of time or number of usage, but also defines critical system components for supporting related functions, possible failure reasons and impact, as well as the most efficient strategies for addressing this impact.

Morphological analysis is incorporated for multi-dimensional analysis of failure impacts, as well as for evaluating possible strategies for addressing the impact.

The approach was applied for the example of live tutoring system; and based on the application results the following opportunities of the further research were identified:

- The approach should support evaluation of the initially stated reliability requirement, based on the defined failure mode and risk strategy, so that in the final stage

"Reliability requirements final definition" it is possible to evaluate realistic reliability that can be achieved and agree on the new requirement or reiterate failure analysis and come up with new strategies for achieving desired reliability level.

- Different evaluations in the approach that now are based on the human assessments (for instance, functions supporting components, failure modes impact) should be reworked into automated evaluation that can be performed by the machine. This is needed if we want to integrate this approach into SCPS.

References

1. Woo, S.: Introduction to Reliability Design of Mechanical/Civil System. In Reliability Design of Mechanical Systems: A Guide for Mechanical and Civil Engineers, Cham: Springer International Publishing, pp. 1–6 (2017).
2. Jiang, R.: Design Techniques for Reliability. In Introduction to Quality and Reliability Engineering, Berlin, Heidelberg: Springer Berlin Heidelberg pp. 147–168 (2015).
3. Malkawi, M. I.: The art of software systems development: Reliability, Availability, Maintainability, Performance (RAMP). Human-centric Comput. Inf. Sci., 3(1), 22, Dec. (2013).
4. Tan, C. M., Carlo, M.: Overview of Reliability Engineering. In Theory and Practice of Quality and Reliability Engineering in Asia Industry, pp. 3–23 (2017).
5. Alho, P., Mattila, J.: Breaking down the requirements: Reliability in remote handling software. Fusion Eng. Des., 88(9), 1912–1915 (2013).
6. Immonen, A., Pakkala, D.: A survey of methods and approaches for reliable dynamic service compositions. Serv. Oriented Comput. Appl., 8(2), 129–158, Jun. (2014).
7. Kusters, R. I., Van Solingen, R., Trienekens, J. J. M., Wijnands, H.: User-perceptions Of Embedded Software Reliability. In: Gritzalis D. (eds) Reliability, Quality and Safety of Software-Intensive Systems. IFIP — The International Federation for Information Processing. Springer, Boston, MA, pp 67–82 (1997).
8. Ahuja, A.: Reliability Requirements, Risk Management, and Associated Building Systems Engineering. In Integration of Nature and Technology for Smart Cities, Cham: Springer International Publishing, pp. 203–222 (2016).
9. Lawrence, R., Dunn, D., Pe, R. L., Dunn, D., South, W. L.: Optimal system reliability: The way forward. In 2008 55th IEEE Petroleum and Chemical Industry Technical Conference, pp. 1–7 (2008).
10. Xie, M., Models, S. R., Applications, P.: Software Reliability Models for Practical Applications. In Software Quality and Productivity: Theory, practice, education and training, M. Lee, B.-Z. Barta, and P. Juliff, Eds. Boston, MA: Springer US, pp. 211–214 (1995).
11. Kravets, R., Calvert, K., Krishnan, P., Schwan, K.: Adaptive Variation of Reliability. In High Performance Networking VII: IFIP TC6 Seventh International Conference on High Performance Networks (HPN `97), 28th April -- 2nd May 1997, White Plains, New York, USA, A. Tantawy, Ed. Boston, MA: Springer US, pp. 202–216 (1997).
12. Peeters, J. F. W., Basten, R. J. I., Tinga, T.: Improving failure analysis efficiency by combining FTA and FMEA in a recursive manner. Reliab. Eng. Syst. Saf., vol. 172, no. December 2017, pp. 36–44 (2018).
13. Khaiyum, S., Kumaraswamy, Y. S.: An Effective Method for the Identification of Potential Failure Modes of a System by Integrating FTA and FMEA. In ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I, vol. I, pp. 679–686 (2014).

14. Gigante, G., Gargiulo, F., Ficco, M., Pascarella, D.: For Consistency Verification Between Requirements and FMEA. pp. 403–413.
15. Spreafico, C., Russo, D., Rizzi, C.: A state-of-the-art review of FMEA/FMECA including patents. Comput. Sci. Rev., vol. 25, pp. 19–28 (2017).
16. Fleischmann, H., Kohl, J., Franke, J.: A modular web framework for socio-CPS-based condition monitoring. In 2016 IEEE World Conference on Factory Communication Systems (WFCS), pp. 1–8 (2016).
17. Lenzini,G., Mauw, S., Ouchani, S.: Security analysis of socio-technical physical systems. Comput. Electr. Eng., vol. 47, pp. 258–274 (2015).
18. Ritchey, T.: Modelling Complex Socio-Technical Systems Using Morphological Analysis Adapted from an address to the Swedish Parliamentary IT Morphological Analysis: What is MA used for ? Mess. Russell J. Bertrand Russell Arch. (2002).
19. Harvey, P. L.: Toward a Discovery and Strategic Alignment Matrices for Socio-technical Systems ' Design. Community Informatics Design Applied to Digital Social Systems, vol. 12, pp 279–312 (2017).
20. Ritchey, T., Society, S. M.: Wicked Problems: Modelling Social Messes with Morphological Analysis. Acta Morphol. Gen., 2(1), pp. 1–8 (2013).
21. Duczynski, G.: Morphological analysis as an aid to organisational design and transformation. Futures, vol. 86, pp. 36–43 (2017).
22. Johansen, I.: Technological Forecasting & Social Change Scenario modelling with morphological analysis. Technol. Forecast. Soc. Chang., vol. 126, no. February 2017, pp. 116–125 (2018).
23. Levina, O., Kranich, L.: Mobility and the Internet of People: A Morphological Analysis. In 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), pp. 915–920 (2016).
24. For, T., Creative, P.: Morphological approach to non-quantified modeling Morpological approach to problem solution (2013).
25. Ritchey, T.: Principles of Cross-Consistency Assessment in General Morphological Modelling. 4(2), pp. 1–20 (2015).
26. Araújo, R. D. A., Soares, S., Oliveira, A. L. I.: Expert Systems with Applications Hybrid morphological methodology for software development cost estimation. Vol. 39, pp. 6129–6139 (2012).
27. Nwana, H. S.: Intelligent tutoring systems: an overview. Artif. Intell. Rev., 4(4), pp. 251–277, Dec. (1990).