COMPLEXITY AND ERROR POTENTIAL OF DIGITAL I&C

R. J. Heigl and A. Lindner
TÜV Rheinland ISTec GmbH – Institut für Sicherheitstechnologie
Boltzmannstraße 14
85748 Garching near Munich, Germany
Raimund.Heigl@de.tuv.com; Arndt.Lindner@de.tuv.com

ABSTRACT

Utilization of digital I&C for application in nuclear power plants led to a growing discussion on how to create methods for a probabilistic evaluation on such systems. Issues arise on the one hand due to a lack of commonly accepted models for quantitative evaluation of availability and reliability and on the other hand due to the integration of such models into the evaluation of reliability of an overall system. Nevertheless programmable and even more complex systems are implemented to satisfy growing functionality requirements. Therefore the question for a method to evaluate the reliability of systems based on software or on programmable logic became even more urgent. The approach described is based on the commonly accepted assumption that more complex designs are less reliable.

Key Words: complexity, PLC function diagram, function block, FPGA netlist, interconnection

1 INTRODUCTION

The average age of NPPs currently operated in Europe exceeds 20 years. Thus the need for backfitting and modernization will inescapably grow in the next future. Most of the installed I&C systems are based on analogue technologies such as electromagnetic relays and analogue electronic modules. The technical solutions available on the market mainly rely on digital technologies, applying microcontrollers or Field Programmable Gate Arrays (FPGA) [9]. Those technologies still raise many technical and procedural issues. One of them is the quantification of software reliability.

Since the time that computer-based systems have been used in other safety relevant applications (e.g. Aviation, Telecommunication) there is an increasing discussion concerning the probabilistic safety assessment of software. The problems are due to the lack of widely accepted models for determining the reliability of software systems.

As known from practical experience it is the complexity of a software-system together with the high variety of its operational profile, which determines the hazard of the software to fail, i.e. its dependability. Therefore, prediction of software dependability, based on complexity measurement is one of the promising approaches for quantifying the reliability of software [7].

In order to facilitate this approach for digital I&C systems applied in the nuclear field, research effort was spent to identify and quantify complexity of digital I&C systems and its correlation to reliability [1-3]. The approach introduced in previous projects was modified in order to enhance significance and also to apply the complexity measurement to a wide range of programmable I&C systems including Programmable Logic Controller (PLC) and FPGA technology. The focus of this paper is directed to the application of the methodology to different technologies [8], [10], [11].

2 CONCEPT FOR COMPLEXITY MEASUREMENT

The concept was introduced in [1], [2], and [4] and can now be applied to any generic, graphic-based, digital I&C platforms (e.g. TELEPERM XS, COMMON Q, TRICON CX, RadICS). The application software of I&C systems based on such platforms is represented as logic diagram (LD) of elementary functions based on either PLC or FPGA technology.

The elementary functions can be classified as:

- logic or arithmetic functions such as OR, AND, ADD etc.,
- basic I&C functions e.g. for implementing a comparison or an interpolation curve,
- specific functions such as ramp generator or sorter.

The elementary functions represented by elementary blocks in the graphical representation will further on be called function blocks in case of PLC logic and basic blocks in case of FPGA logic. They are implemented as modules of a library. They are defined as the lowest level of programming items in current digital I&C systems. Large and complex I&C functions can be designed and generated by combination of these elementary blocks.

The functionality of the I&C system is implemented by logic diagrams of either function blocks or basic blocks. The logic diagram is represented as function diagram in case of PLC logic or as netlists in case of FPGA logic.

The complexity measurement of the application software is subdivided into two levels, according to the structure of the logic diagram modules implementing the I&C functionality. These levels are

- Measurement of the elementary blocks,
- Measurement of the logic diagram.

The complexity of the different blocks is the first level for determining the overall complexity of an I&C function.

There is a limited and fixed set of elementary blocks available to implement the logic diagrams. Therefore a meaningful procedure for complexity measurement of an I&C system should generate a matrix describing the complexity features of all the blocks which are available. This complexity matrix is independent of any specific application and of any logic diagram.

On the second level a complexity metric for the interconnection of elementary blocks of logic diagrams has to be defined and evaluated. The overall complexity of a digital I&C system will be quantified by the complexity of the application specific logic diagrams taking into account the complexity matrix generated on the first level for elementary blocks.

3 COMPLEXITY MATRIX

The function blocks represent a set of software functions, which are implemented in some standardized high level language such as ANSI-C, ADA, etc.

For these software functions implementing the function blocks a complexity matrix was generated. The complexity matrix comprises all typical function blocks and refers to the relevant aspects describing the complexity of the function blocks. These aspects are compiled using a White-Box-View based on the code and a Black-Box-View based on documentation and user-manuals [2], [4].

The complexity characteristics obtained are put together in a complexity matrix. Examples for details of a complexity matrix are given in two separate tables (table I. and table II.) for reasons of clarity.

Table I. Excerpt of a Complexity Matrix (signals, parameter)

		signal					parameter				
	Fan Out	inputs		outputs			not change-able	change-able	sum	condi-tions	de-rived
		bi-nary	ana-log	bi-nary	ana-log	re-port					
FB01 PAR	1	0	0	0	1	0	2	1	3	0	0
FB02 ADD	1	0	3	0	1	0	0	0	0	0	0
FB03 CURVE	2	0	1	0	1	0	1	2N+1	2N+2	N+2	0
FB04 COMP	1	0	3	1	0	0	1	4	5	2	6

Table II. Excerpt of a Complexity Matrix (variables, resources)

	int. var.	memory	memory usage		runtime [µs]			error-codes	function	
			code (bytes)	stack (bytes)	init.	param.	calc.		text	picture table
FB01 PAR	0	0	125	24	13	5	5	1	1	0
FB02 ADD	2	0	276	34	19	10	10	2	4	0
FB03 CURVE	8	0	592	48	101	92	92	2	7	0
FB04 COMP	2	1	503	32	28	19	12	1	11	0

The examples are taken from a PLC based platform.

4 METHOD FOR MEASURING THE COMPLEXITY OF LOGIC DIAGRAMS

Logic diagrams are implementing the I&C functions. They are established by interconnection of elementary functions.

The features describing the complexity of a logic diagram can be subdivided into three groups:

- Basic counting metrics, that can be taken directly from the graphic representation of the logic diagram namely number of input and output signals, the number of elementary blocks and number of interconnections,
- Metrics to describe the complexity of the interconnection of the elementary blocks
- Variability features of digital I&C systems that have an impact on complexity, such as number of internal memories and changeable parameters

It is in the nature of the logic diagrams that individual elementary blocks or even sets of individual elementary blocks are often used for computing more than only one output signal. This overlapping

interconnection of function blocks for the computation of various output signals is the normal case for logic diagrams. Another essential impact on complexity is introduced by the usage of common input signals.

A metric V(LD) for these two types of interconnection within a logic diagram is given by

$$V(LD) = \sum_{Sai} \frac{|VB(S_{ai})|+|IN(S_{ai})|}{|BLD|+|SIN|} \quad (1)$$

Where S_{ai} indicates all the individual output signals of the LD

$VB(S_{ai})$ indicates the set of elementary blocks that are involved in the computation of the output signal S_{ai}

$IN(S_{ai})$ indicates the set of input signals that are involved in the computation of the output signal S_{ai}

BLD indicates the set of elementary blocks making up the LD

SIN indicates the set of input signals of the LD

| | indicates cardinality (number of elements of a set)

This metric basically is an average value reflecting the complexity regarding the output signals only. For LD with only one output signal the value of V(LD) is always 1, no matter how many input signals are used for computing the output signal and how the signal is processed via elementary blocks.

Therefore a metric $V(LD)_{mod}$, for the modified interconnection complexity is introduced in [5]. It is given by

$$V(LD)_{mod} = \sum_{Si} \frac{|VB(S_i)|+|IN(S_i)|}{|BLD|+|SIN|} \quad (2)$$

Where S_i indicates all the individual signals (internal and output) of the LD

$VB(S_i)$ indicates the set of elementary blocks that are involved in the computation of the signal S_i

$IN(S_i)$ indicates the set of input signals that are involved in the computation of the signal S_i

Determining the difference

$$V(LD)_{mod} - V(LD) = V(LD)_{int} \quad (3)$$

yields a metric for the internal interconnection complexity $V(LD)_{int}$, since $V(LD)_{mod}$ considers the interconnection of all signals (of elementary blocks and output signals) and V(LD) only takes into account the interconnection of signals contributing to output signals. That modification is an essential extension of the methodology described in [4] and presented at previous meetings [1-2]. The modified approach exhibits a more detailed view on the complexity characteristics of logic diagrams. It should be mentioned that the calculation of $V(LD)_{mod}$ requires long calculating times in case of large logic diagrams.

5 COMPLEXITY VECTOR

The characteristics identifying the complexity of logic diagrams are made up by simple volume measures, by the complexity of the interconnection and by variability characteristics of digital I&C systems.

These characteristics comprise a wide range of complexity properties. To combine these to a single number would hide all the detailed information gained during complexity analysis. Therefore, the complexity features of a logic diagram should be kept separated and represented in a complexity vector.

According to the previous considerations the components of the complexity vector are defined by

K1: Number of elementary (function or basic) blocks

K2: Number of inputs signals of a LD

K3: Number of output signals of a LD

K4: Number of interconnections between inputs, elementary blocks and outputs

K5: Number of upstream FDs for the relevant FD (as part of an I&C function)

K6: Number of downstream FDs for the relevant FD (as part of an I&C function)

K7: Interconnection complexity V(LD) for the interconnection of elementary blocks

K8: Modified interconnection complexity $V(LD)_{mod}$

K9: Internal interconnection complexity $V(LD)_{int}$

The components K1, K2, K3 and K4 can directly be taken from the graphical representation of the logic diagram or determined tool-based for large function diagrams e.g. for an FPGA netlist directly out of an Electronic Design Interchange Format (EDIF) file.

The components K5 and K6 are taken into account in case the relevant FD acts as a part of an I&C function, where the functionality is achieved via interconnection of several FDs. When the complexity vector of a whole I&C function or a whole system is determined, the value of both K5 and K6 will be zero.

The components K7, K8 and K9 can in principle be computed by hand, but for large function diagrams such as FPGA logic or the examples given in chapter 6, the metrics for the interconnection complexity were computed by a script [6].

6 COMPARISON OF TWO SIMILAR APPLICATIONS

The Complexity Vectors of two similar setups of PLC logic in nuclear application are compared. The two functions are developed for the same type of I&C system and constitute the same functionality but are used in different NPPs and have been developed by different teams under different provisions concerning the development of function diagrams. The two functions are simply called function 1 and function 2. The function diagrams compounded to the respective functions (see Figure 1) are called controls and engines in the following.

The functionality is achieved by a “control” and three “engines”. For function 1 redundant engines of the same type are used, while for function 2 three different engines are in place. The interconnections between the items are slightly different.

As opposed to function 1, function 2 was developed using specific macros built from the original function blocks. The approach applied to the function diagram design of function 2 can save time during function diagram development, but in turn may result in a utilization of more function blocks and

interconnections as required for the intended function.

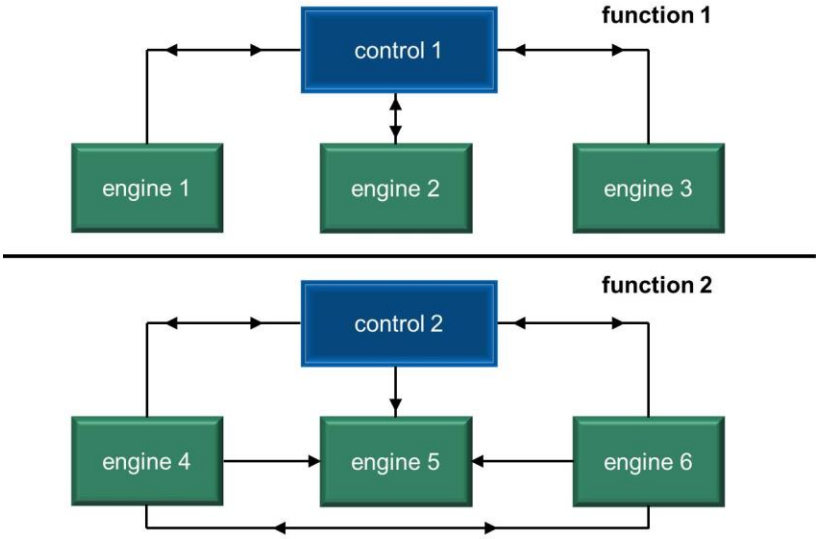

Figure 1. Comparison of two similar setups in nuclear application with the same functionality.

Comparison of the Complexity Vectors (see table III.) of the two controls shows, that almost all values of the complexity vector of control 2 are higher than those of control 1. The number of used function blocks is significantly higher for control 2. Additionally the value of the interconnection complexity V(LD) is more than four times higher. Considering the determination of the interconnection complexity, where the number of function blocks appears in the denominator, the complexity of the function diagram of control 2 is identified as being more complex.

Table III. Comparison of Complexity Vectors for PLC logic

PLC	Definition	control 1	control 2	engine 1	engine 4	function 1	function 2
K1:	# function blocks	17	68	11	112	50	383
K2:	# inputs	14	21	18	17	53	45
K3:	# outputs	14	50	16	27	47	99
K4:	# connections	55	177	46	224	178	778
K5:	# upstream LD	3	2	2	3	0	0
K6:	# downstream LD	6	3	2	3	0	0
K7:	V(LD)	9	43	14	13	42	67
K8:	$V(LD)_{mod}$	16	72	20	68	68	282
K9:	$V(LD)_{int}$	7	29	6	55	26	215

The comparison of the Complexity Vectors of the engines draws a similar picture. The internal interconnection complexity is almost 10 times higher for engine 4 as compared to engine 1, while the interconnection complexity V(LD) has a similar value. This is remarkable because the significantly higher number of used function blocks compensates the number of signals contributing to outputs for the calculation of V(LD), but not for the calculation of the internal interconnection complexity $V(LD)_{int}$. This demonstrates the higher complexity of the engines of function 2 compared to function 1.

Finally the Complexity Vector for each function, which consists of the control and the three engines, respectively, is compared. Almost all values of the complexity vector are higher for function 2. As a logical consequence of considering the previous results the values for number of function blocks, interconnections, modified interconnection complexity and internal interconnection complexity are significantly higher for function 2 as compared to function 1. Again despite a similar value for the interconnection complexity V(LD), the value of the internal interconnection complexity is much higher for function 2. This clearly shows that the crucial difference between the two function diagram designs has an impact on the complexity of the interconnection of the function blocks of which the respective function diagram is composed of.

7 APPLICATION TO FPGA LOGIC

The method for measuring the complexity can also be used for FPGA logic. FPGA logic cells have a finer granularity than PLCs. As a result, for the same functionality, it takes more cells to implement a function in an FPGA than in a PLC [9]. The example used to show the implementation of the method was provided by [8]. The I&C function implemented by the FPGA logic is a hysteresis function for the reactor trip of a NPP and is illustrated in figure 2.

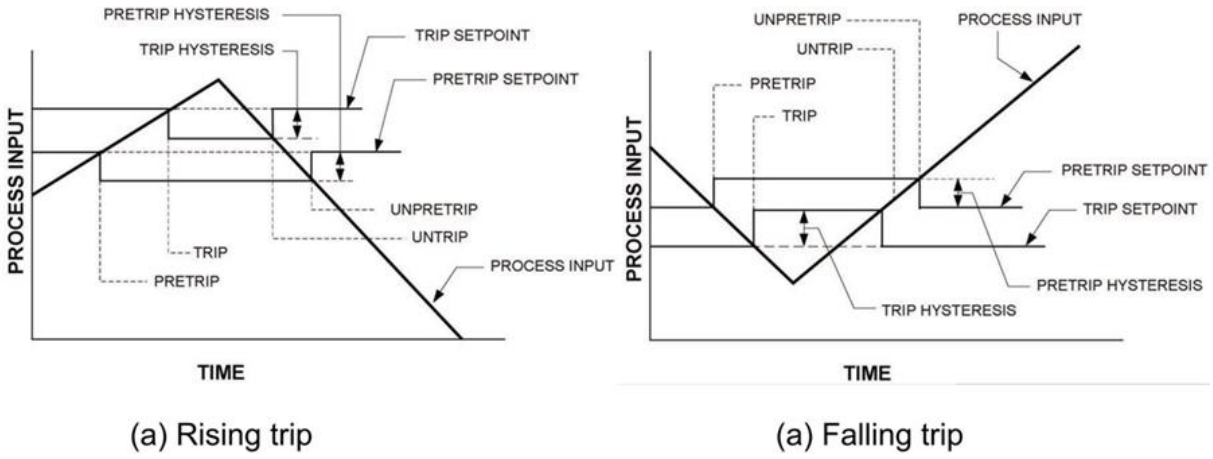

Figure 2. Example of a function implemented by FPGA logic in nuclear application [8].

FPGAs have segmented interconnect structures [9]. The FPGA logic is divided into several levels in the visual representation. In the selected example the netlist can be divided into four levels (0 to 3). The elementary functions are represented as cells on each level. With each higher level, a cell is split up into several sub-cells where the functionality of a mother cell is realized by interconnection of daughter-cells, which feature a more and more elemental functionality. On the level having the finest granularity (level 3 in the given example), the remaining cells are represented by basic blocks featuring the fundamental basic functions. In order to depict the changing functionality on each level the complexity vector is presented for each level of the visual representation of the FPGA logic in table IV and fig. 3. At the end the most relevant level concerning the complexity of the FPGA logic is the base-level complexity vector (level 3 in

the given example).

Table IV. Complexity Vector of FPGA Application separated in levels

FPGA	Definition	Level 0	Level 1	Level 2	Level 3
K1:	# basic blocks	100	107	225	506
K2:	# inputs	75	75	75	75
K3:	# outputs	23	23	23	23
K4:	# connections	198	228	620	1227
K5:	# upstream LD	0	0	0	0
K6:	# downstream LD	0	0	0	0
K7:	V(LD)	20	17	13	5
K8:	$V(LD)_{mod}$	41	39	95	115
K9:	$V(LD)_{int}$	21	22	82	110

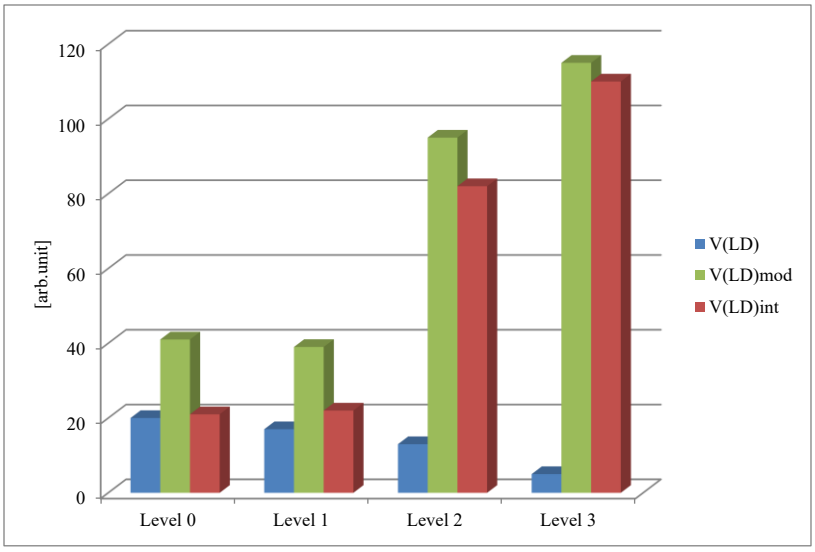

Figure 3. Comparison of the interconnection complexity of different levels of the FPGA logic.

The number of input and output signals is the same for each level of the visual representation, since only the cells are split up into sub-cells. Comparison of the complexity vectors of each level shows, that the interconnection complexity V(LD) is descending with each higher level but the internal complexity $V(LD)_{int}$ is increasing. This behavior meets exactly one's expectations.

8 CONCLUSIONS

Modern digital I&C systems, programmed by graphic-based specification of its functionality have a common structure. Therefore a unitary two-level concept can be used for description and quantification of their complexity.

The lower level is defined by elementary functions, the function blocks or basic blocks. Their complexity is calculated using a White-Box-View based on the code and a Black-Box-View based on documentation and user-manuals. The results of the White-Box-View and Black-Box-View are collected and represented in the corresponding complexity-matrices for the function blocks of a function diagram or the basic blocks of a FPGA design respectively.

The second level is defined in a graphical manner by the function diagrams or netlists. They represent the functionality of the I&C function and are represented by logic diagrams of the function blocks or the basic blocks. Their complexity is defined on basis of the signal-interconnection taken from their graphical representation of the LD in case of function diagrams or taken from a FPGA netlist. The quantification of the complexity features is compiled in a complexity-vector.

The quantification of complexity in two levels fits to the structure of digital I&C systems. It is rather descriptive and allows:

- identification of regions with raised complexity within a digital I&C design,
- comparison of complexity between I&C functions.

Additionally, the method developed for complexity measurement allows:

- identification of errors in the function diagrams or netlists respectively such as loops, isolated parts etc.,
- signal tracing e.g. of signals corresponding to a specific output and therefore supporting the generation of test cases.

The approach for complexity measurement was applied to two PLC-based functions, which have the same functionality but were designed with different methods for different NPPs. Comparison of the respective complexity vectors showed that the internal interconnection complexity is significantly different. The two compared PLC-based functions possess a different complexity as a result of a different function diagram design.

The presented method of complexity measurement is also applicable to FPGA logic, although further research is required to compare the complexity of different FPGA applications.

The extended approach now allows the analysis of a broad range of generic, graphic-based, digital I&C platforms for different technologies like PLC and FPGA logic and exhibits a more detailed view on the complexity characteristics of logic diagrams as compared to the method that has been described formerly.

9 ACKNOWLEDGMENTS

The work on complexity measurement and reliability of software in digital I&C systems is funded by the German Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (BMUB) under the project number 3614R01310 within its scope of research in reactor safety. Special thanks go to Prof. Junbeom Yoo for providing data of FPGA-based logic for controllers in NPPs.

10 REFERENCES

1. J. Märtz, A. Lindner, H. Miedl, M. Baleanu “COMPLEXITY MEASUREMENT AND RELIABILITY OF SOFTWARE IN DIGITAL I&C-SYSTEMS”, *NPIC&HMIT 2004*, Columbus, Ohio, September, 2004.
2. J. Märtz, A. Lindner, H. Miedl, “COMPLEXITY MEASUREMENT OF SOFTWARE IN DIGITAL I&C-SYSTEMS”, *NPIC&HMIT 2009*, Knoxville, Tennessee, April 5-9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009).
3. J. Holmberg, S. Guerra, N. Thuy, J. Märtz, B. Liwång, “HARMONICS-EU FP7 PROJECT ON THE RELIABILITY ASSESSMENT OF MODERN NUCLEAR I&C SOFTWARE”, *NPIC&HMIT 2012*, San Diego, CA, July 22-26, 2012.
4. J. Märtz, H. Miedl, A. Lindner, C. Gerst, „KOMPLEXITÄTSMESSUNG DER SOFTWARE DIGITALER LEITTECHNIKSYSTEME“, ISTec-A-1569, February 2010.
5. F. Hellie, A. Lindner, A. Mölleken, “KOMPLEXITÄT UND FEHLERPOTENTIAL BEI SOFTWAREBASIERTER DIGITALER SICHERHEITSLEITTECHNIK”, Interim Report, ISTec-A-2891, February 2016.
6. R. Heigl, F. Hellie, A. Lindner, A. Mölleken, „KOMPLEXITÄT UND FEHLERPOTENTIAL BEI SOFTWAREBASIERTER DIGITALER SICHERHEITSLEITTECHNIK“, Final Report, (in preparation).
7. Smidts, C., “FROM MEASURES TO RELIABILITY”, *NPIC&HMIT 2000*, Washington D.C., November 13-16, 2000.
8. J. Yoo, E. Kim, D. Lee, J. Choi, ”AN INTEGRATED SOFTWARE DEVELOPMENT FRAMEWORK FOR PLC & FPGA-BASED DIGITAL I&C”, *ISOFIC 2014*, Jehu, Rep. of Korea, August 24-28, 2014.
9. IAEA Nuclear Energy Series No. NP-T-3.17, APPLICATION OF FIELD PROGRAMMABLE GATE ARRAYS IN INSTRUMENTATION AND CONTROL SYSTEMS OF NUCLEAR POWER PLANTS, International Atomic Energy Agency, Vienna (2016).
10. J.-E. Holmberg, P. Bishop, S. Guerra, N. Thuy, “SAFETY CASE FRAMEWORK TO PROVIDE JUSTIFIABLE RELIABILITY NUMBERS FOR SOFTWARE SYSTEMS,” *Proc. of 11th International Probabilistic Safety Assessment & Management Conference, PSAM 11,* Helsinki, June 25–29, 2012(2012).
11. S. Guerra, N. Thuy, “SAFETY JUSTIFICATION FRAMEWORKS: INTEGRATING RULE-BASED, GOAL-BASED, AND RISK-INFORMED APPROACHES, ”*Proc. of 8th International Conference on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC & HMIT)*, July 22-26, 2012, San Diego, CA(2012).