Assurance Case Driven Design based on the Harmonized Framework of Safety and Security Requirements

Vladimir Sklyar, Vyacheslav Kharchenko

National Aerospace University "KhAI", Kharkiv, Ukraine
v.sklyar@csn.khai.edu, v.kharchenko@csn.khai.edu

Abstract. Assurance (Security and Safety) Case is an approach to prove critical systems and software compliance with security and safety requirements. We propose an advanced framework named as Assurance Case Driven Design (AC DD) to improve cost-effectiveness of certification and licensing processes. AC DD is based on Claim-Argument-Evidence-Criteria (CAEC) notation and Development-Verification&Validation-Assurance Case (DVA) notation. An example of AC DD application for Functional Safety Management part of requirements of the standard IEC 61508 is considered.

Keywords: certification, assurance case, safety and security life cycle, IEC 61508

Key terms. MathematicalModeling, MathematicalModel, SoftwareSystems

1 Introduction

Assurance (Security and Safety) Case implementation goals are, firstly, proving a conformance of security and safety critical systems and software with requirements, and, secondly, discovering a gap in these requirements conformance [1]. Assurance Case Driven Design (AC DD) is proposed to apply Assurance Case building as soon as possible for the earliest stages of life cycle activities. The main goal of this approach is to improve cost-effectiveness of certification and licensing processes [2].
AC DC also supports the following important topics:

- Research of integral security and safety features of modern critical control and communication systems and networks as an integral property; security importance increasing requests implementation of security requirements as a part of licensing issues; such approach is named as Security Informed Safety Case [3]; such approach is targeted to analyze safety and security in a structured way and creating Security Informed Safety Case that provide justification of safety taking into particular consideration the impact of security [4];
- Research of different type of embedded components, such as Field Programmable Gates Arrays (FPGAs) and microprocessor units (MCUs);

- Research applications for specific market, for example, cloud computing, big data analytics and IoT with high level requirements to safety, security and quality of service (QoS).

At the present Assurance Case methodology progress lays in multidisciplinary dissemination of theory and experience [5]. Experts form different area may develop a general and cross-platform security and safety assurance approaches. At the same time there are some potential areas for Assurance Case improvement, such as:

- Assurance Case should faster find gaps in compliance with requirements than demonstrate such compliance;
- It is reasonable to implement Assurance Case from the earliest stage of life cycle; one more reason to do it is a prospective idea to combine of Assurance Case with argument based design approach, what is a basis for elimination a board between design and modeling;
- Assurance Case should provide as many details as it is needed for comprehensive analysis;
- Assurance Case should support re-using of system safety and security files during system operation and maintenance;
- Assurance Case should support cost effectiveness of system life cycle;
- It is reasonable to improve formalism of Assurance Case against empirics in descriptions.

Assurance Case has two sides of description and implementation:

1. A static part which describes an approach to combine arguments for assurance support;
2. A dynamic part to support a static part movement between stages of analyzed system life cycle.

The modern researches in the Assurance Case area are mostly oriented to industry, such as new coming applications covering [6,7], patterns development [8] and computer tools improvement [9,10]. At the same time, theoretical researches can improve Assurance Case usability for different industries [11].

CAE and GSN formalisms are based on classical set theory, graph theory and relation algebra. Such relations tracing allows us to propose extensions for existing notations. In this article we discuss an approach to develop Claim-Argument-Evidence-Criteria (CAEC) notation as an extension of CAE notation [4].

The second side of Assurance Case implementation is dynamic application via life cycle stages. We propose Development – Verification & Validation –Assurance Case (DVA) notation for description of dynamic Assurance Case application.

We choose the standard IEC 61508 “Functional safety of electrical/electronic/programmable electronic safety-related systems” and applied it for Industrial Control System (ICS). It allows implementing a pilot application of AC DD for Functional Safety Management part of the IEC 61508 requirements [12,13].

This paper continue series of researches in Assurance Case domain [4,14]. A novelty of this paper consists in application of previously developed AC DD approach to ICS safety and security assurance and assessment.

2 Concept of Assurance Case Driven Design

2.1 General Framework for Assurance Case Driven Design

A general AC DD framework is described in details in [4,14]. Usually the first step in any system development is signing a contract. This contract is an input for system functional requirement as well as certification or licensing framework for safety and security critical applications. The Requirement Specification has to be developed on the base of contractual functional requirements.

Safety and security critical systems shall have an important addition to the Requirement Specification describing not functional requirements targeted to implement system integrity. AC DD approach proposes to present such requirements in a view of a preliminary Assurance Case. Such preliminary Assurance Case is not a result of assessment but a target which has to be achieved after the system implementation. Not functional requirements of Assurance Case are an input for Safety or Security Management Plan which has cover life cycle description with all development support processes. Some parts of not functional requirements (for example, self-diagnostic requirements) may affect the Requirement Specification. After that staged life cycle with V&V and other supporting processes activities (Project Management, Configuration Management and other) has to be implemented in accordance with Safety (Security) Management Plan. After the contract and the Requirement Specification stages life cycle usually includes design, implementation, integration, validation, installation, and commissioning stages. Assurance Case activities have to be implemented after each of the stage. Safety or security certification has to finalize system life cycle before transfer it in operation at the customer site. Also during operation a periodical assessment or certification has to be done with associated update of Assurance Case.

Assurance Case structure depends from a type of the application. For example a typical structure of Assurance Case for industrial functional safety related application includes: security activities coordinated with safety, process implementation and assessment, and product implementation and assessment. Assessment can be done in a view of deterministic analysis, probabilistic analysis or demonstration.

2.2 Claim-Argument-Evidence-Criteria (CAEC) Notation

There are two the main notation used for Assurance Case [1]: Claim-Argument-Evidence (CAE) and Goal Structured Notation (GSN).

Usually both CAE and GSN are presented in a graphical view. Also an approach to represent the CAE in a table view is widely used in industry [5,6]. If we discuss about formal background of Assurance Case notations, all of them can be described with set theory and graph theory apparatus.

Another side of theoretical approach is structured development of Assurance Case. A typical multi levels required by regulations include the following: Level 0 (L0) – conceptual level; Level 1 (L1) – design level; Level 2 (L2) – implementation level.

In the AC DD framework we propose some addition for Assurance Case CAE notations to be able assess specific features of critical systems. Acceptance criteria and coverage criteria are two additional entities which have to be taken into account for

support arguments and evidences. Acceptance criteria are the conditions when stated requirements are met. From the point view of Assurance Case, acceptance criteria provide us ability to state the right arguments which are consistent with the claim and to provide the evidences which are consistent with the arguments. Coverage criteria describe how completely the claim is met.

From the point view of Assurance Case, coverage criteria provide us ability to state multiple arguments to completely cover all claim features and to provide multiple evidences which completely cover the arguments. Acceptance criteria for a claim can be extracted from both argument and/or evidence. In general case acceptance criteria provide a quantitative and qualitative description of a situation when the claim is met. A coverage criterion is a measure used to describe the degree to which evidence for specific arguments is provided. A modified CAE notation which we name Claim-Argument-Evidence-Criteria (CAEC) notation is given on Fig. 1.

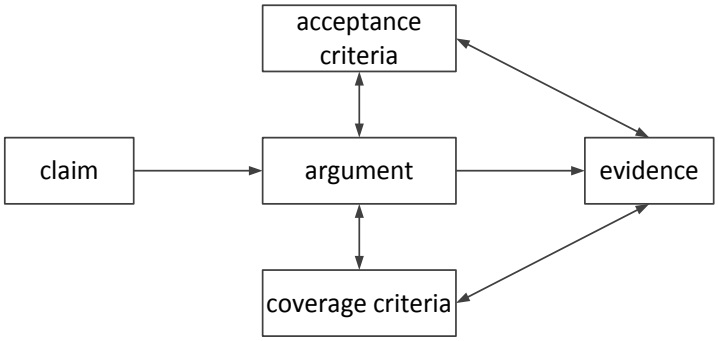

Fig. 1. Claim-Argument-Evidence-Criteria (CAEC) Notation

The next step of CAE / CAEC notation development is to support activities of Safety & Security Life Cycle (SSLC) stages with implementation of Assurance Case. Specification and design requirements are the inputs for each of the SSLC stage. After any stage fulfillment, requirements implementation assessment has to be performed.

2.3 Development-V&V-Assurance Case (DVA) Notation

The following activities are mandatory for each of the SSLC stage (see Fig. 2):

- Development targeted to move an implemented product representation stage by stage through SSLC;
- V&V targeted to check conformance of the SSLC stage development outputs to the SLC stage development inputs;
- Assurance Case update based on assessment of performed development and V&V activities.

The proposed DVA Development-V&V-Assurance Case (DVA) notation is based on following fundamentals:

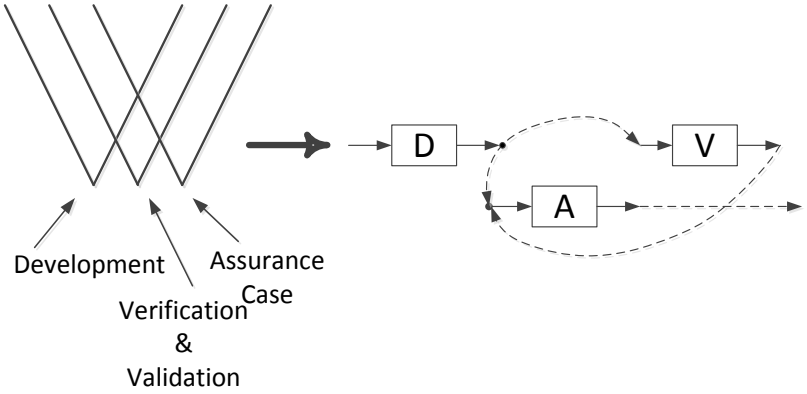

Fig. 2. Transition from V-shape Life Cycle to Development-V&V-Assurance Case (DVA) Notation

- Safety & Security Life Cycle can be represented in a view of three components: Development (D), Verification and Validation (V&V) and Assurance Case (A);
- Development activities are staged implementation of requirements in design description of system, hardware and software, and after that implementation of requirements in a physical system, hardware and software;
- Development also covers processes implementation to support development of the product; processes also are described in a view of requirements which are collected in project plans;
- Typically requirements are represented and handled as database records; from this point of view the main operation with requirements are CREATE (to add), DELETE, MODIFY (if requirement needs some sense correction), EDIT (if requirement needs only editorial correction without changing of a sense);
- Forward and backward requirement tracing shall be implemented at each of Life Cycle stage to assure: 1) all previous stage requirements are implemented into the next stage documents; 2) no new requirement appears in the next stage documents; 3) all the requirements are verified or validated;
- Compliance of the product of next Life Cycle stage with the product of the previous Life Cycle stage is checked by implementation of V&V process;
- Compliance of processes implementation (including development and V&V processes) is checked by audits when processes implementation evidences are investigated against the project plans requirements; these audits can be a part of Assurance Case activities;
- All three D, V and A components of Safety & Security Life Cycle have specific inputs and outputs for each of the Life Cycle stage; so a diagram on Fig. 3 represents DVA relations for some single stage.

To develop a graph and theoretical-set based model for DVA notation a diagram on Fig. 2 should be elaborated to reflect feedback relations after V&V and Assurance Case performance (see Fig. 3). Direct data transmission and feedback data are highlighted with different templates of lines.

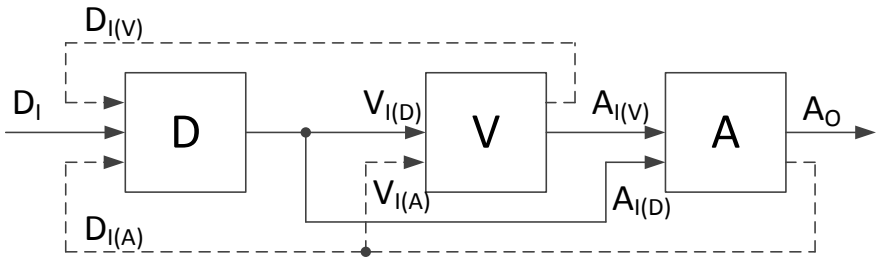

Fig. 3. Graph and theoretical-set based description of DVA Notation

It is clear for Fig. 3, that input and output sets have some overlapping, so sets of DVA data flows ate described in terms of inputs. There are the following data sets transmitted between components of DVA:

- $D_I = \{d_{i1}, d_{i2}, \ldots, d_{iK}\}$ – a set of development process inputs transmitted from the out of the previous life cycle stage;
- $V_{I(D)} = \{v_{id1}, v_{id2}, \ldots, v_{idL}\}$ – a set of V&V process inputs transmitted from the out of development process;
- $A_{I(D)} = \{a_{id1}, a_{id2}, \ldots, a_{idM}\}$ – a set of Assurance Case process inputs transmitted from the out of development process;
- $A_{I(V)} = \{a_{iv1}, a_{iv2}, \ldots, a_{ivN}\}$ – a set of Assurance Case process inputs transmitted from the out of V&V process;
- $D_{I(V)} = \{d_{iv1}, d_{iv2}, \ldots, d_{ivP}\}$ – a set of development process inputs transmitted from the out of V&V process (a corrective feedback);
- $D_{I(A)} = \{d_{ia1}, d_{ia2}, \ldots, d_{iaQ}\}$ – a set of development process inputs transmitted from the out of Assurance Case process (a corrective feedback);
- $V_{I(A)} = \{v_{ia1}, v_{ia2}, \ldots, v_{iaR}\}$ – a set of V&V process inputs transmitted from the out of Assurance Case process (a corrective feedback);
- $A_O = \{a_{o1}, a_{o2}, \ldots, a_{oS}\}$ – a set of Assurance Case process inputs transmitted to the next life cycle stage after all the internal corrections.

The following software tools are available to develop Assurance Case [10]:

- ASCE (Assurance and Safety Case Environment) developed by British company Adelard supports both CAE and GSN;
- Astah GSN developed by Japanese company Change Vision supports only GSN;
- NOR-STA developed by Polish company Argevide supports GSN and specific list-oriented TRUST-IT notation.

3 Framework of Safety and Security Requirements

Result of industrial standards considering [12,15] allows representing existing security requirements to Industrial Control Systems (ICS) related with a restricted set of categories (see Fig. 4).

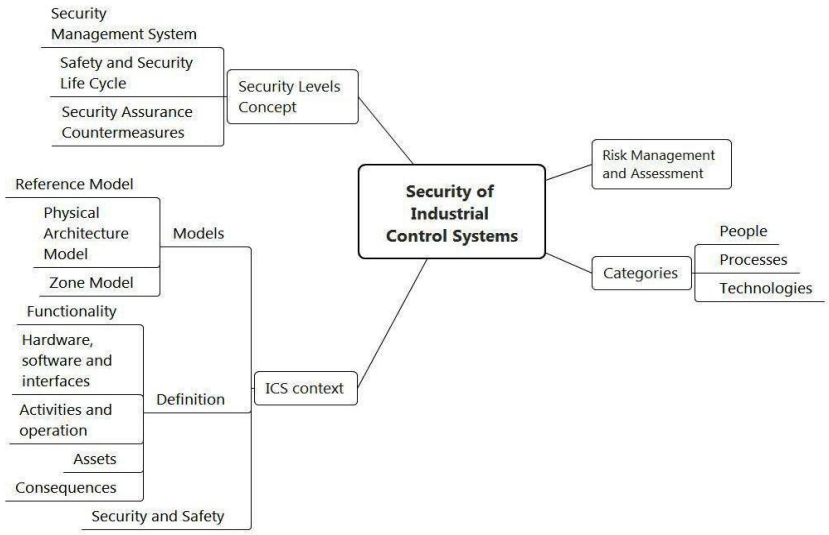

Fig. 4. Security concepts and requirements taxonomy

This conceptual security requirements taxonomy include four the main parts:

- Risk management and assessment as a corner stone for definition of acceptable risks levels and countermeasures for risks reduction;
- Categories of security features implementation which include triad "People – Process – Technologies";
- ICS context which drive to define requirement taking into account specifics of ICS; this concept includes three types of models (reference, physical architecture and zone models) as well as functionality, components, assets and other definitions, and security and safety coordination issues;
- ICS security levels concept which grades risk levels for ICS separated parts and establishes different life cycle processes and countermeasures for different security levels.

A risk management process should be employed throughout an organization, using a three-tiered approach to address risk at the organization level; mission/business process level; and information system level (IT system and ICS). After that the principle of the "People – Process – Technologies" categories triad shall be implemented as the core of Information Security Management System (ISMS) [15].

A context of the ICS can be represented by combination of three models, which are Reference Model, Physical Architecture Model, and Zone Model (see Fig. 5).

Zone Model provides the context for assessing common threats, vulnerabilities, and the corresponding countermeasures needed to attain the level of security required to protect the grouped assets. After grouping assets in this manner, a security policy is defined for all assets that are members of the zone. The results of this analysis are used to determine the appropriate protection required based on the activities performed in the zone [15].

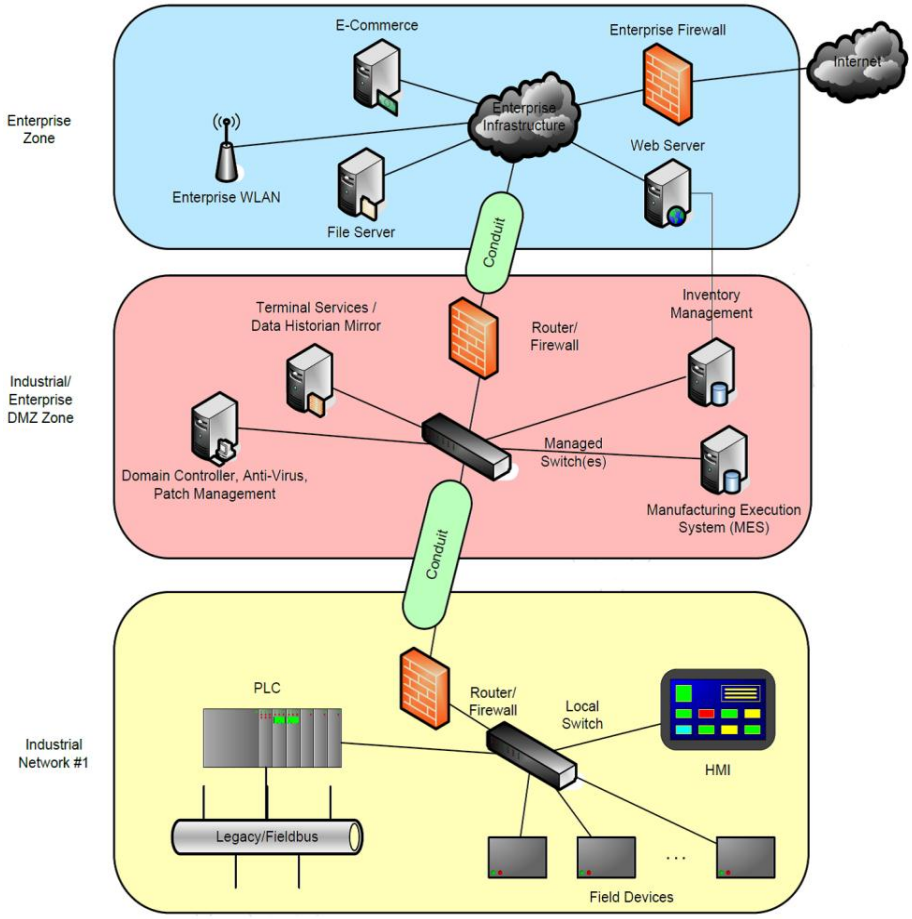

Fig. 5. Zone model of Industrial Control Systems (source: ISA/IEC 62443)

Every situation has a different acceptable level of security. For large or complex systems, it may not be practical or necessary to apply the same level of security to all components. Differences can be addressed by using the concept of a zone, defined as a logical or physical grouping of physical, informational, and application as sets sharing common security requirements. This concept can be applied in an exclusive manner where some systems are included in the security zone and all others are outside the zone. A conduit is a particular type of zone that groups communications that can be logically organized into a grouping of information flows within and also external to a zone. Channels are the specific communication links established within a communication conduit.

A set of ICS functional safety requirement can be found in series of industrial standards, for example, IEC 61508 "Functional safety of electrical/electronic/programmable electronic safety-related systems".

These functional safety requirements can be divided in some following categories:

- Requirements to functional safety management;
- Requirements to functional safety life cycle;
- Requirements to systematic (system and software design) failures avoidance;
- Requirements to random (hardware) failures avoidance.

A scope of the above requirements is highly dependent from as named Safety Integrity Level (SIL) which establishes relation between system risk level and a scope of the related safety assurance countermeasures.

This requirement taxonomy can be applied for security concept (see Fig. 4). Firstly, Security Levels shall be implemented for ICS taken into account risks levels. Secondly, ISMS shall be implemented and coordinated with functional safety management issues. Thirdly, a common security and safety life cycle shall be established to cover all the process of ICS development, verification and validation. Fourthly, common safety and security risks shall be avoided to implement coordinated countermeasures against random (hardware) and systematic (system and software design) failures. Examples of common safety and security random failures avoidance countermeasure are redundancy, self-diagnostic, hazards protection and others. Examples of common safety and security systematic failures avoidance (attacks avoidance for security) are access control and configuration control. Fifty, assessment shall be periodically performed for both, security and safety. The discussed approach is the base for security and safety coordination, as it is represented on Fig. 6.

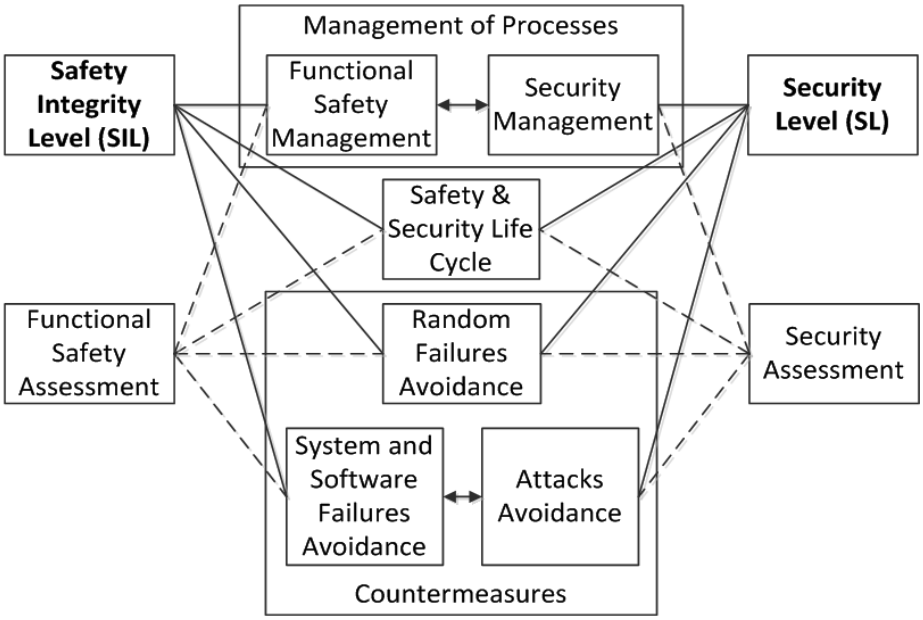

Fig. 6. A concept of harmonized safety and security requirements

4 Functional Safety Management Part of Assurance Case Driven Design Framework

The standard IEC 61508 "Functional safety of electrical/electronic/programmable electronic safety-related systems" has been chosen to specify safety requirements to ICS. This umbrella safety standard contains seven parts (see Fig. 7) covering different types of computer control systems for different industrial domains.

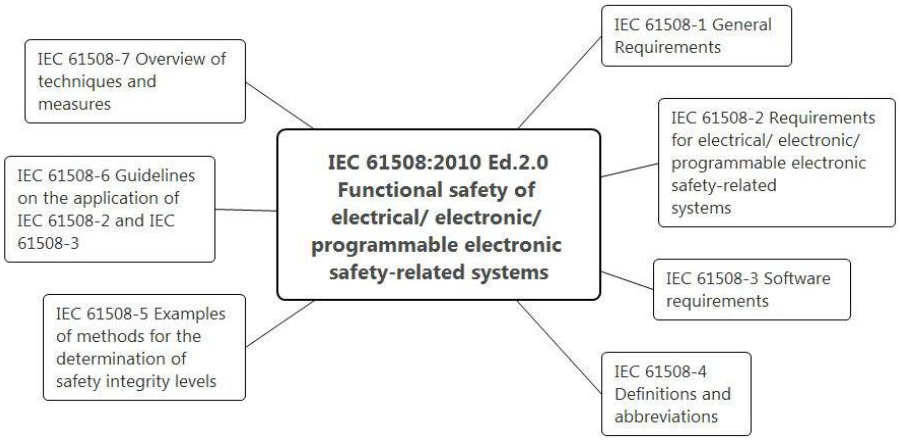

Fig. 7. Seven parts of IEC 61508

Let's consider requirements to Functional Safety Management in accordance with the proposed safety and security requirements structure (see Fig. 8).

These requirements are contained in Section 6, which is included in parts 1, 2, and 3 of the IEC 61508. The bulk of the requirements is presented in IEC 61508-1. There is only a link to IEC 61508-1 in IEC 61508-2. IEC 61508-3 contains an amendment related with software configuration management. It is reasonable also to consider Section 5 "Documentation" and Section 8 "Functional Safety Assessment" of IEC 61508-1, Section 7.1 "General (System safety lifecycle requirements)" of IEC 61508-2, as well as Section 7.4.4 "Requirements for support tools, including programming languages" of IEC 61508-3.

The Functional Safety Management Plan (FSMP) shall be elaborated for a specific ICS implementation project to comply with the above requirements. The FSMP shall include the following parts:

- Configuration Management (see Fig. 8);
- Verification and Validation (see Fig. 9);
- Documentation Management (see Fig. 10);
- Tools Selection and Evaluation structured in accordance with safety affect (see Fig. 13);
- Functional Safety Assessment (see Fig. 11);
- Human Resource Management (see Fig. 12).

The above activities include specific management directions, and this is a reason why any of such activity request a separated plans and reports. So, the FSMP should contain only general information without many details.

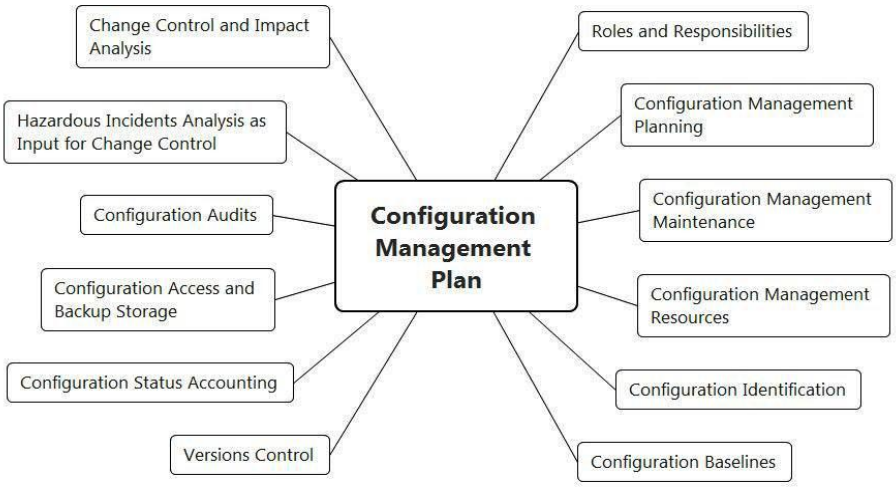

Fig. 8. A structure of Configuration Management Plan complied with IEC 61508-1 (Section 6)

Fig. 9. A structure of V&V Plan complied with IEC 61508-2 (Section 7.1)

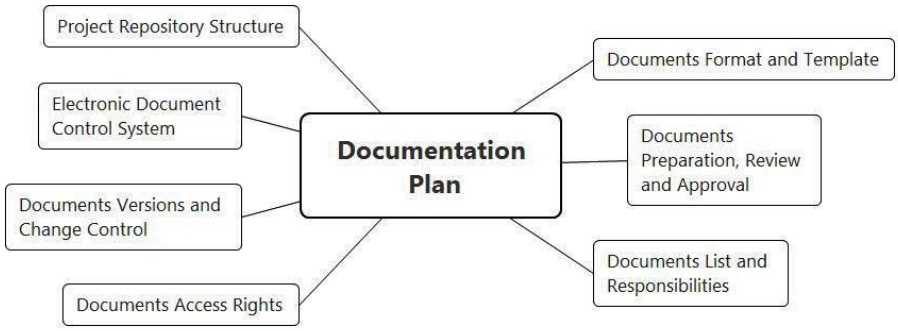

Fig. 10. A structure of Documentation Plan complied with IEC 61508-1 (Section 5)

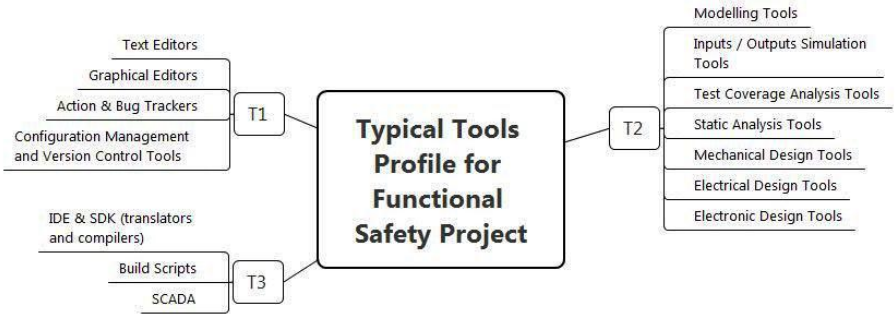

Fig. 11. A structure software tools profile complied with IEC 61508-3 (Section 7.4.4); (T1 – generates no outputs which can directly or indirectly contribute to the executable code, T2 – supports the test or verification of the design or executable code, T3 – generates outputs which can directly or indirectly contribute to the executable code)

Fig. 12. A structure of Functional Safety Audit Plan complied with IEC 61508-1 (Section 8)

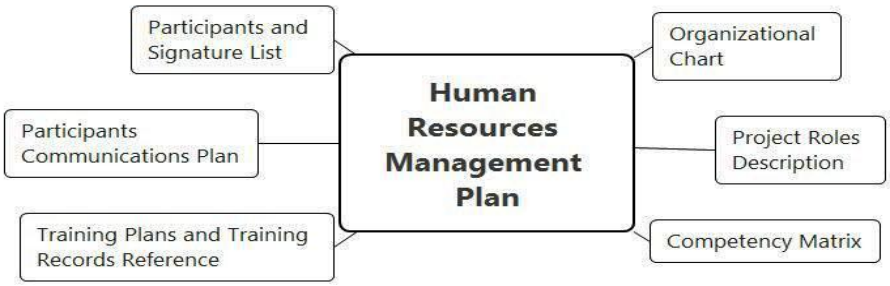

Fig. 13. A structure HR Management Plan complied with IEC 61508-1 (Section 6)

Also the FSMP should contain the following:

- Project Policy and Strategy;
- Functional Safety Life Cycle and Requirement Tracing;
- Suppliers Management in relation with Quality Management System;
- Security; it should be noted the IEC 61508 contains just very top level requirements to security; taking into account safety and security requirements harmonization approach, it would be appropriate to put reference to ISMS documents in this part of the FSMP.

Fig. 14 represents a full framework for Functional Safety Management requirements in accordance with the IEC 61508 (applicable sections from part 1, 2, and 3).

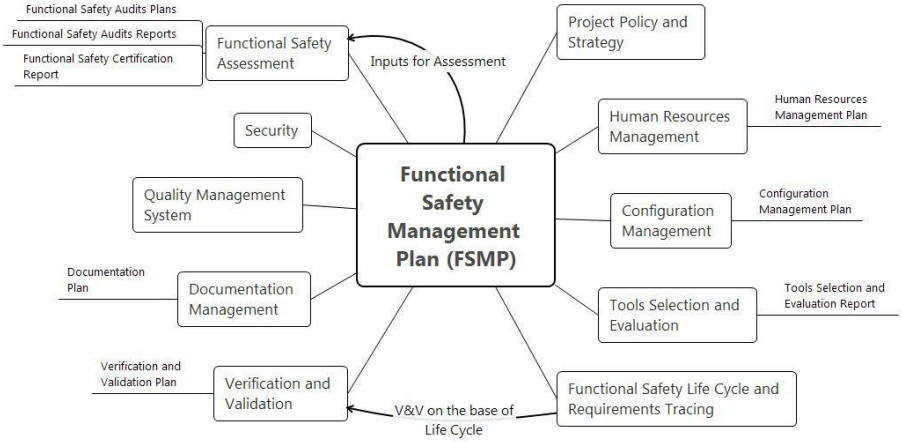

Fig. 14. A proposed structure of Functional Safety Management Plan complied with IEC 61508

To give details on the atom requirements level let's consider Human Resource Management process. The Human Resource Management Plan (HRMP) shall be developed to comply with Functional Safety Management requirements. This plan shall include the following parts which are response to IEC 61508 requirements, marked below as HRj (see Fig. 13):

HR1: {The HRMP shall contain Organizational Chart of the ICS implementation Project};

HR2: {The HRMP shall contain descriptions of the Project roles in traceable to Organizational Chart};

HR3: {The HRMP shall contain Competency Matrix with specified competencies required for each of the Project role and with results of competencies compliance evaluation};

HR4: {The HRMP shall contain references to Training Plan and Training Records, which are needed to support personnel competencies at the required level; Training Plan and Training Records shall be available};

HR5: {The HRMP shall contain participants Communication Plan};

HR6: {The HRMP shall contain participants list with signature that confirm awareness concerning the Project roles and responsibilities}.

5 Application of Assurance Case Driven Design Methodology: Human Resource Management Part

To apply the proposed AC DD methodology to specified requirements {HR1,…,HR6} we need to establish a structure of SSLC. An example of SSLC for ICS is presented on Fig. 15.

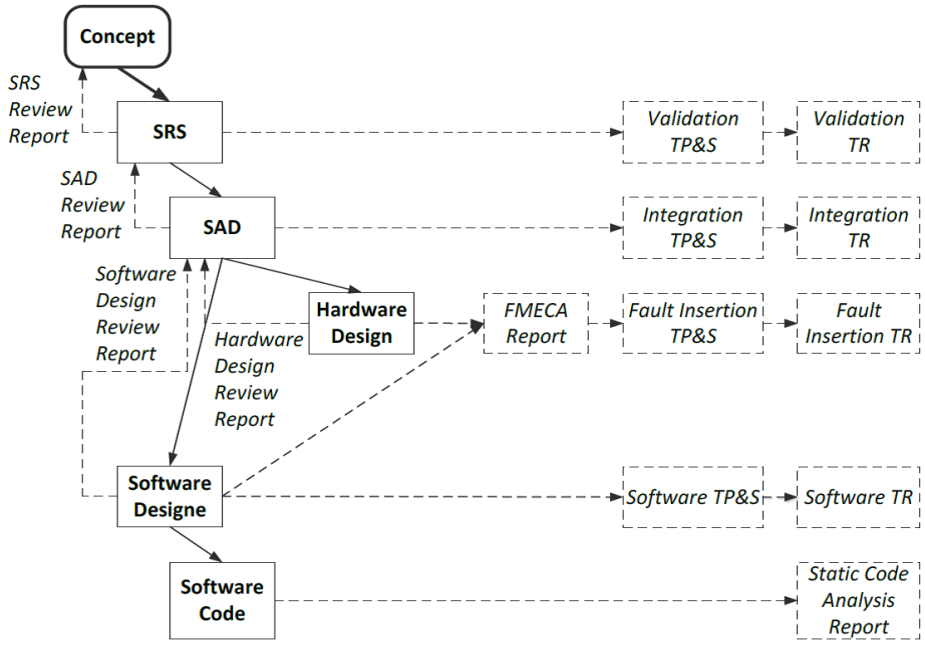

Fig. 15. V-shape Safety and Security Life Cycle

Inputs to implement AC DD can be presented in a view of a table (see Table 1), when rows are equivalent to Life Cycle stages and columns are equivalent to requirements. In this case the table cells will specify the Assurance Case operations for a specific requirement at a specific Life Cycle stage.

A general view of Assurance Case operation can be represented as Aij(Di,Vi,HRj) = {Cij, a1ij,…,aKij, e1ij,…, eLij, cc1ij,…CCMij, AC1ij,…, ACNij}, where Cij is a claim for i-th SSLC stage (development or V&V) and j-th requirement, akij is an argument with index from 1 to K, elij is an evidence with index from 1 to L, CCmij is a coverage criterion with coverage index from 1 to M, ACnij is an acceptance criterion with index from 1 to N. Some Aij can be repeated without changing through the SSLC stages, and some Aij can be done once for some specific SSLC stages and after that missed later SSLC stages.

Table 1. AC DD implementation: operation with requirements through SSLC stages

SSLC stage name	SSLC index	HR1	HR2	…	HR6
Concept	D1	A(D1,HR1)	A(D1,HR2)	…	A(D1,HR6)
SRS	D2	A(D2,HR1)	A(D2,HR2)	…	A(D2,HR6)
SRS Review	V2	A(V2,HR1)	A(V2,HR2)	…	A(V2,HR6)
SAD	D3	A(D3,HR1)	A(D3,HR2)	…	A(D3,HR6)
SAD Review	V3	A(V3,HR1)	A(V3,HR2)	…	A(V3,HR6)
HW Design	D4	A(D4,HR1)	A(D4,HR2)	…	A(D4,HR6)
HW Design Review	V4	A(V4,HR1)	A(V4,HR2)	…	A(V4,HR6)
FMECA	V5	A(V5,HR1)	A(V5,HR2)	…	A(V5,HR6)
SW Design	D5	A(D5,HR1)	A(D5,HR2)	…	A(D5,HR6)
SW Design Review	V6	A(V6,HR1)	A(V6,HR2)	…	A(V6,HR6)
SW Coding	D6	A(D6,HR1)	A(D6,HR2)	…	A(D6,HR6)
Code Analysis and Review	V7	A(V7,HR1)	A(V7,HR2)	…	A(V7,HR6)
SW Testing	V8	A(V8,HR1)	A(V8,HR2)	…	A(V8,HR6)
Fault Insertion Testing	V9	A(V9,HR1)	A(V9,HR2)	…	A(V9,HR6)
Integration Testing	V10	A(V10,HR1)	A(V10,HR2)	…	A(V10,HR6)
Validation Testing	V11	A(V11,HR1)	A(V11,HR2)	…	A(V11,HR6)

6 Conclusions

The proposed AC DD approach may provide some benefits on the base of from cost-effective as named “embedded certification” briefly described by DVA notation. This cost-effective solution can work under conditions when the total cost of life cycle with application of “embedded certification” would be less than the cost of usual life cycle with usual after life cycle certification, i.e.:

Cost (DVA Life Cycle) < Cost (DV Life Cycle) + Cost (Certification).

This paper provides a framework of harmonized safety and security requirements which are divided into the following groups:

- Requirements to management of safety and security;
- Requirements to Safety and Security Life Cycle;
- Requirements to avoidance of random and systematic failures;
- Requirements to safety and security assessment.

The next practical steps of AC DD development have to be directed to analyze existing Safety and Assurance Cases for Internet of Things, cloud computing and big data analytics as well as to enforce Assurance Cases for IoT products with Security Informed approach.

References

1. Kelly T (1998) Arguing Safety: A Systematic Approach to Managing Safety Cases. PhD thesis. University of York
2. Leveson N (2004) A New Accident Model for Engineering Safer Systems. Safety Science 42:237-270
3. Netkachova K, Müller K, Paulitsch M, Bloomfield R (2015) Security-Informed Safety Case Approach to Analysing MILS Systems. In: Proceedings of International Workshop on MILS: Architecture and Assurance for Secure Systems
4. Kharchenko VS, Sklyar VV (2016) Assurance Case Driven Design for software and hardware description language based systems. Radioelectronic and Computer Systems 5(79):98-103
5. Guerra S (2014) Understanding, assessing and justifying I&C systems using Claims, Arguments and Evidence. Nuclear Safety and Simulation 5:15-26
6. Ye F, Cleland G (2012) Weapons Operating Centre Approved Code of Practice for Electronic Safety Cases. Adelard LPP, London
7. Denney EW, Pai GP (2015) A Methodology for the Development of Assurance Arguments for Unmanned Aircraft Systems. In: Proceedings of 33rd International System Safety Conference (ISSC 2015)
8. Denney EW, Pai GJ (2015) Safety Case Patterns: Theory and Applications. NASA/TM–2015–218492. Technical Report, NASA
9. Hawkins R, Habli I, Kolovos D, Paige R, Kelly T (2015) Weaving an Assurance Case from Design: A Model-Based Approach. In Proceedings of 16th IEEE International Symposium on High Assurance Systems Engineering (HASE'15)
10. Valkonen J, Tommila T, Linnosmaa J, Varkoi T (2016) Safety demonstration of nuclear &C – an introduction. SAUNA Task 3.1 report. VTT-R-00167-16. Research report, VTT
11. Sklyar V (2016) Safety-critical Certification of FPGA-based Platform against Requirements of U.S. Nuclear Regulatory Commission (NRC): Industrial Case Study. In: Proceedings of the 12th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge (ICTERI 2016)
12. IEC 61508 (2010) Functional safety of electrical/electronic/programmable electronic safety-related systems (in 7 parts). Geneva, IEC
13. Medoff M, Faller R (2010) Functional Safety – An IEC 61508 SIL 3 Compatible Development Process. exida.com L.L.C., Sellersville, PA, USA
14. Sklyar V, Kharchenko V (2016) Assurance Case Driven Design for Computer Systems: Graphical Notations versus Mathematical Methods In: Proceedings of the Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI 2016).
15. NIST SP 800-82 (2015) Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations such as PLC. – National Institute of Standards and Technologies, Gaithersburg, MD, USA