MDEP Generic Common Position

Multinational Design Evaluation Programme
Generic Common Position
DICWG No2 – PUBLIC USE

Date: 12 March 2013
Validity: **until next update or archiving**
Version C

MDEP Generic Common Position No DICWG-02

Related to: Digital Instrumentation and Controls Working Group activities

COMMON POSITION ON SOFTWARE TOOLS FOR THE DEVELOPMENT OF SOFTWARE FOR SAFETY SYSTEMS

Participation

Countries involved in the MDEP working group discussions:	Canada, Finland, France, India, Japan, People's Republic of China, Republic of Korea, Russian Federation, South Africa, the U.A.E., the U.K. and the U.S.
Countries which support the present common position	Canada, Finland, France, India, Japan, People's Republic of China, Republic of Korea, Russian Federation, South Africa, the U.A.E., the U.K. and the U.S.
Countries with no objection:	
Countries which disagree	
Compatible with existing IAEA related documents	Yes

Multinational Design Evaluation Programme

Digital Instrumentation and Controls Working Group

GENERIC COMMON POSITION DICWG NO2: COMMON POSITION ON SOFTWARE TOOLS FOR THE DEVELOPMENT OF SOFTWARE FOR SAFETY SYSTEMS

Summary

The Digital Instrumentation and Controls Working Group (DICWG) has agreed that a common position on this topic is warranted given the increase of use of Digital I&C in new reactor designs, its safety implications, and the need to develop a common understanding from the perspectives of regulatory authorities. This action follows the DICWG examination of the regulatory requirements of the participating members and of relevant industry standards and IAEA documents. The DICWG proposes a common position based on its recent experience with the new reactor application reviews and operating plant issues[1].

Context

The use of appropriate software tools can increase the integrity of the I&C development process, and hence product reliability, by reducing the risk of introducing faults during the process. The use of tools can also have economic benefits as they can reduce the time and human effort required to produce systems, components, and software. Tools can be used to automatically check for adherence to rules of construction and standards, to generate proper records and consistent documentation in standard formats, and to support change control. Tools can also reduce the effort required for testing and to maintain automated logs. In some cases tools are necessary because a specific development methodology requires their use.

Tools are most powerful when they are defined to work co-operatively with each other.

Scope and Definition of terms

This common position applies to software tools used in the development of software for safety systems and software tools are defined to:

- support the capture of requirements,
- support the transformation of requirements into the final system code and data (there may be many intermediate steps),

[1] The goal of MDEP is not to independently develop new regulatory standards. Common Positions are not legally binding and do not constitute additional obligations for the regulators or the licensees but are guidelines, recommendations, or assessments that the MDEP participants agree are good to highlight during their safety reviews of new reactors. Any MDEP member may decide to implement the common positions through its national regulatory process.

- directly support the performance of verification, validation and testing,
- prepare and control application data, and
- manage and control of the processes and products involved in the software development.

In this document *safety system* means a Class 1 system as defined in IEC 61226. The term safety system as used in this document is equivalent to the term *safety system* as defined in IEEE 603 which is incorporated into 10 CFR 50.55a (h) and the term *safety related system* as defined in 10 CFR 50.2.

This common position does not apply to:

- tool support for complex programmable logic devices such as FPGAs,
- off-line tools, used to calculate important variables used during the design and analysis of safety systems, or
- office administration tools used to support tasks not directly concerned with software development (e.g., word processors and project management tools).

Generic Common Position on Software Tools:

1. Tools should be used to support all aspects of the I&C life cycle where benefits result through their use and where tools are available.

A key element of integrated project support environments is to ensure proper control and consistency. If tools are not available, the development of new tools may need to be considered.

2. The benefits and risk of using a tool should be balanced against the benefits and risk of not using a tool.

The important principle is to choose tools that limit the opportunity for making errors and introducing faults, but maximise the opportunity for detecting faults.

System development may be adversely affected by the use of tools in several ways. For example, design tools may introduce *faults* by producing corrupted outputs; and *verification* tools may fail to reveal certain *faults* or types of *faults*.

3. The functionality and limits of applicability of all tools should be identified and documented.

4. The tools and their output should not be used outside their declared functionality or limits of application without prior justification.

For example, tools cannot replace humans when judgement is involved. In some cases, tool support is more appropriate than complete automation of the process

5. Tools should be verified and assessed consistent with the type of tool and the potential of the tool to introduce faults.

For example:

- Verification is not necessary for tools that cannot introduce or fail to detect faults.
- Less rigor in tool verification may be accepted if there is mitigation of any potential tool faults (e.g. by process diversity or system design,),
- Verification is not necessary for the tool outputs that are always systematically verified.

6. The qualification process should take into account experience from prior use.

7. All tools should be under appropriate configuration management.

8. Tool parameters used during the development, verification, or validation of baseline equipment or software should be recorded in the development records.

This is useful not only for the final software consistency; it also helps in assessing the origin of a fault, which may lie in the source code, in the tool, or in the tool parameters. It may also be necessary in the assessment of the potential for common cause failures due to software tools.

9. Section 14 of IEC standard 60880 provides acceptable guidance for the selection, qualification, and use of software tools for the development of software for *safety systems*.

References

Four Party Regulatory Consensus Report On The Safety Case For Computer-Based Systems In Nuclear Power Plants, November 1997

IAEA NS-G-1.1, “Software for Computer Based Systems Important to Safety in Nuclear Power Plants”

IEC 60880, “Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions”

IEC 61226 Ed. 3, “Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions,” International Electrotechnical Commission, Geneva, Switzerland, 2009.

IEEE 7-4.3.2, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations”

Licensing of safety critical software for nuclear reactors. Common position of seven European nuclear regulators and authorised technical support organisations, 2007.