

MISRA Activities for Safety-Related Software Development

David D. Ward

MIRA Limited

Copyright © 2008 SAE International

ABSTRACT

MISRA (The Motor Industry Software Reliability Association) develops practical guidance to assist the developers of safety-related systems in the automotive industry and other sectors in implementing safe and reliable systems. This paper presents a short overview of MISRA's ongoing activities, with particular emphasis on guidance being produced in two areas: process measures for safety-related systems engineering, and product measures for safety-related systems development. Subsequent papers in the session give more details on these activities.

INTRODUCTION

The modern road vehicle has an essential dependence on advanced electronic systems to control its functionality and to deliver the demands for safety, environmental efficiency, comfort and brand differentiation required by manufacturers, legislators and consumers. As a result, system functional safety is a central part of the design and implementation process for these systems, and continues to grow in importance.

MISRA was formed in the early 1990s with the objective of developing practical guidance for the developers of automotive safety-related systems. This paper presents a short overview of MISRA's ongoing activities, with particular emphasis on the guidance being produced in the two areas of safety-related systems engineering and language subsets.

THE MISRA PROJECT

The MISRA project started in the early 1990s as part of a UK government-funded program in Safety-Critical Systems (originally called the SafeIT program) [1]. This program sought to address the challenges emerging in the various industry sectors affected by safety-related systems. In particular, work was underway on the documents that would eventually be published as the international standard and "basic safety publication" IEC 61508 [2].

At that time a number of trends were apparent in the automotive industry:

- Software-based electronic systems were beginning to move from luxury vehicles to the mass market, beginning with engine management systems, anti-lock braking systems and supplementary restraint systems (airbags).
- Software development for automotive systems was moving from programming in assembly language to the use of higher-level languages, notably the "C" language.
- The use of databus systems to interconnect individual electronic control modules was emerging.

It was acknowledged that the automotive industry had a good track record in regard to software, but despite this it did not have a unified approach to developing software. At the same time, while emerging standards such as IEC 61508 [2] provided a good basis for developing safety-related systems, they did not align well with the typical development processes used for road vehicles and their constituent systems.

Some 15 years later the growth in vehicle electronics continues unabated. Various estimates suggest that a typical European luxury vehicle may have up to 40% of its value in the electrical and electronic systems. It may contain on the order of 100 microprocessors, with 85% or more of the vehicle's functions being under the control of or influenced by software-based systems. Some specific trends may also be noted:

- Software-based electronic systems are pervasive in the mass market, with further cascading of systems such as adaptive cruise control and a lane departure warning from luxury vehicles to the mass market.
- Software development for automotive systems is moving from programming in languages such as "C" to the use of model-based development and automatic generation of source code for production systems from the models.
- The use of databus systems to interconnect individual electronic control modules is widespread, and many functions on the vehicle are now distributed, that is to say, they do not reside on a single physical electronic control unit.

THE MISRA PUBLICATIONS

The MISRA philosophy is to provide engineer-accessible guidance to address specific needs in the marketplace. MISRA does not seek to duplicate existing standards, guidelines or publications. Rather, the approach is to identify areas that are not covered by existing material, and to provide information that is suitable for day-to-day use by those responsible for developing electronic systems and also for managing the development of such systems.

THE MISRA GUIDELINES

The first MISRA publication was *Development Guidelines for Vehicle-Based Software* [3]. This document provides guidance on the development of safety-related software in the context of road vehicle embedded electronic control systems. One principal feature of these guidelines is that they took many of the principles from the draft documents that later emerged as IEC 61508 [2], but with a particular emphasis on interpreting them for the automotive industry. The consortium decided specifically not to produce a clause-for-clause interpretation of the standard at that time, since it was desired to provide a flexible approach that would take into account emerging technologies in software development rather than being closely linked to draft documents that underwent substantial change before emerging as the final version of IEC 61508.

The MISRA Guidelines introduced two important new concepts from the draft of IEC 61508 to automotive software development.

A Risk-based approach

The MISRA Guidelines pioneered the use of a risk-based approach to hazard analysis in the automotive industry. In this approach, hazards are identified through a preliminary safety analysis and classified according to the risk associated with them. This information is then used to set requirements on the development process to reduce the risk to the “broadly acceptable” level [4]. The level of risk reduction required is expressed by one of four “integrity levels” (compare safety integrity levels, or SILs, in IEC 61508 and many similar standards). This was a new concept, compared to the existing approach of simply classifying systems as “safety-related” or “non-safety-related”, and has now found broad acceptance in the automotive industry. The MISRA integrity levels not only encompassed risk reduction associated with safety hazards, but also other hazards, such as the impact of an unreliable system on warranty claims. However practitioners using the MISRA Guidelines now use the term SIL for compatibility with comparable standards.

Development Rigor

The MISRA Guidelines make recommendations for the rigor of both the (software) development process and the features of the product implemented through this

process. This is comparable to the extensive requirements for “Techniques and Measures” found in IEC 61508. The MISRA Guidelines contain a summary table that specify the process and product requirements, or “techniques and measures”, for each stage of a generic software development lifecycle against the (safety) integrity level. The MISRA approach is notable for two key differences compared to the IEC 61508 approach:

- Each (safety) integrity level implicitly contains the requirements of each preceding level; hence the SIL 2 requirement is the SIL 1 requirement plus additional measures; the SIL 3 requirement is the SIL 2 requirement plus additional measures, i.e. the requirements of SILs 1 and 2 plus additional measures. A similar approach is found in the approach to the requirements for security measures that were developed around the same time [5].
- The requirements are goal-based rather than prescriptive. This is particularly important in ensuring that the guidance can be adapted to emerging technologies. IEC 61508 is very prescriptive, and contains many requirements for “techniques and measures” that are either specific to the industrial process control sector that it originated in, are now of only historic interest, or both. Many other standards and guidelines in the safety-related systems community are goal-based, including those for civil aviation [6] and the UK defense industry [7]. The latter is of particular note, since the most recent version dispensed with the previous highly-prescriptive approach akin to IEC 61508 in favor of a set of goals expressed in only 17 pages of requirements.

MISRA C

A specific requirement of the “Techniques and Measures” table of the MISRA Guidelines is to use a “restricted subset of a standardized structured language” at SIL 2 and above. Following release of the MISRA Guidelines, it was noted that many automotive companies were starting to use the “C” programming language for embedded systems. The issues with the use of “C” for developing safety-related systems are well-documented, and it was quickly recognized that rather than individual automotive companies developing their own coding requirements for “C”, a MISRA publication addressing “C” would be beneficial to the automotive industry. The “C” language is defined by an international standard [8] that documents many of the issues with “C”, including behaviors of the language that are unspecified, undefined or implementation-defined. However the standard is some 200 pages long, and it is unlikely that the majority of “C” programmers have a copy, still less have read it. The standard also documents some fundamental issues in the arithmetic model of “C” such as integral promotion and the usual arithmetic conversions, but these are usually not covered in “C” programming courses.



Documents

Document Sharing Hub