

Emerging Software Best Practice and how to be Compliant

Roger S. Rivett

Rover Group Ltd.

Several documents have been published recently concerning software, systems and safety, for example IEC 1508 (draft) and MISRA Guidelines. While the former is a generic industrial standard, the latter are guidelines specifically for the automotive industry. A recent ESPRIT project, CASCADE, has produced a Generalised Assessment Method for assessing the safety of railway and automotive systems. A common theme running through all these publications is the use of safety analysis to determine a safety integrity level and then the use of the integrity level to determine appropriate target reliability figures and design procedures.

This paper will introduce the above documents with particular emphasis on MISRA and CASCADE and explain how their use can increase the confidence in the correctness of software based systems and maintain a credible best practice position for the developer and vehicle manufacturer.

Future lightweight and small cars will increasingly use software based systems to increase functionality for minimum hardware increase and so the development of demonstrably safe systems is essential.

1. Introduction

This paper looks at the growth of software within automotive vehicles and suggests that its use will increase in the future. It also suggests that the safety implications of these future systems will become greater. The environment in which software is currently developed (e.g. legislative requirements, component & vehicle validation) is discussed and it is suggested that this will be inadequate for future systems. Two recent publications concerned with the development of software for safety related systems, IEC 1508 and the MISRA Guidelines, are introduced together with their key themes of safety integrity levels and corresponding process rigour. It is suggested that the automotive industry agree a common standard of good practice for software development along the lines of these documents. For the most advanced or safety related systems it is suggested that some form of independent assessment should be used to demonstrate that this good practice has been followed. In this context the ESPRIT CASCADE project is briefly described.

2. The Growth Of Software In Vehicles

2.1. The Past

Over the last 15 to 20 years software has become a key component within most vehicles sold for private transport use.

Software based engine management controllers have been used since the 1970s and are now virtually standard on all vehicles. This growth in software has been driven by the emissions legislation which has specified lower and lower emissions levels. Some systems now use “throttle by wire” where the mechanical link of the throttle cable is replaced by electronics and software.

Anti-lock brake systems also have a long pedigree and, although they are not fitted as standard on all vehicles in the way that engine management controllers are, they are a very mature software based product which are fitted to most vehicles in the upper medium, executive and luxury segments. The automotive industry could not do without them.

More recently software has been used in air-bags, control body functions such as lights, wash-wipers and security systems, as well as chassis systems such as active suspension.

2.2. The Present

In addition to the essential functionality associated with the systems already mentioned, software is increasingly being used to provide “customer delight” features. These are features which, while not essential for the primary purpose of the vehicle, have great customer appeal and help to differentiate one product from another. In some respects they are analogous to trim and styling features. Lazy locking is perhaps a good example.

Table 1 lists current typical applications together with an indication of the likely software content given as the size in kilo bytes of the executable. These figures are quite conservative.

Description	Code size (k)
Heating, ventilation and air-conditioning controller	20
Body electrics controller and security	50
Window lift controller	4
Instrument pack and trip computer	44
Engine management system	50
Transmission controller	16
Anti-lock brakes and traction control	32
Active Suspension	8
Multiple airbag controller	20
Navigational equipment/mobile phone/in car entertainment	10
Total	254

Table 1 Typical Current Software Applications

2.3. The Future

Many of the products currently in the concept stage replace more of the traditional mechanical links with electronics and software, for example steer by wire, or else take more and more of the primary control away from the driver, for example altering the characteristics of power assisted steering so as to encourage the driver to keep in lane. Perhaps the most extreme example is the concept of the “road train” where software is used to link many vehicles together on highways so that they can travel at high speed as a single entity. In this case all driver control is removed for the time that the vehicle is part of the “train”.

While these type of systems will be first introduced in the executive and luxury segments, if they get market acceptance they will be fitted throughout the model range. So these issues will affect Lightweight and Small Cars in the near future. In fact small vehicles are being designed now which are full of this new technology. It might well be the case that Lightweight vehicles actually have *extra* needs for electronics. E.g. powertrain & body might need more sophisticated control; traction and ride control might need to be more sophisticated than in a heavy vehicle.

So the use of software in vehicles is set to grow. In fact it is probably not an over-statement to say that the future of the automotive industry, in common with much of the modern world, is dependent upon software.

The safety implications of these concept systems are much greater than the current ones and consequently the need for the systems and the software to function as intended is also much greater.

3. Ensuring Correct Functioning

3.1. Drivers For Correct Functioning

There are a number of factors which encourage the automotive industry to ensure that their products function correctly. These include:

- Type Approval legislation
- Published reports for buyers
- Product liability legislation
- Automotive magazines
- Insurance companies
- New Car Assessment Programme (NCAP)

Type Approval Legislation applies at three different levels of detail, components (e.g. headlights), systems (e.g. braking) and whole vehicle (e.g. exhaust emissions). While some legislation gives quite detailed mechanical specifications, for example tyre minimum tread depth, none of the current legislation specifically mentions electronic components let alone the software they contain. The requirements specifications are given and checked treating the item under consideration as a black box. Details of how the requirements have been met is outside the scope of the legislation.

The other organisations in the above list all make independent assessments of different vehicles and publish the comparative results. While the tests performed do not have any legal status, the results put great pressure on manufacturers whose products perform less well than their competitor's.

All of these focus on the performance of the overall vehicle. Details of how a feature is implemented are not considered and so the question of whether or not the underlying design is sound are not asked.

3.2. Techniques for Achieving Correct Functioning

There are a number of techniques used by the automotive industry to establish that the products function as intended, of which the principal ones are component validation and vehicle validation.

Again these focus on the overall performance, be it at component level or vehicle level. This is not unreasonable as the assessment of the vehicle, as described above, will also be at these levels. The inherent soundness of the design is the responsibility of the component engineer and is not generally checked by any higher authority.

3.3. Implications for Software Based Components

With the introduction of the ever more complex and safety critical software the automotive industry needs to be pro-active in ensuring sound design and correct functioning of software. Waiting for Type Approval or for other publications and agencies to turn the spotlight on these aspects of vehicles is not in the best interests of the automotive industry.

The extensive use of component and vehicle validation, while having a good track record, will not be sufficient in the future for the more complex and safety critical software employed. The automotive industry needs to make rigorous use of best practice with regard to software development at all levels of the industry, that is vehicle manufacturers, component suppliers and software subcontractors.

Standardisation is a common means within the automotive industry to facilitate use of common components (e.g. OSEK operating system, Keyword 2000 diagnostics protocol). In the light of the increasing importance of software, its use in ever more critical systems and the absence of any other imperative, the automotive industry ought to adopt a commonised approach to the development of software. Work on defining what this common approach might be has already started.

4. An Evaluation of Recent Best Practice Publications

In recent years at least 2 documents have been published dealing with the issues of software, systems and safety.

4.1. IEC 1508

The first was what is now known as IEC 1508 [IEC 1508]. The first draft of this was published in November 1989 and was entitled “IEC 65A(Secretariat)94 : Draft. Software for computers in the application of industrial safety-related systems”. In June 1995 another draft was issued in seven parts entitled “Draft IEC 1508 - Functional safety : safety related systems”.

Although this document has its background in the process control industry, its intention is to be a generic industrial standard which can be instantiated for different industrial sectors. Its main themes are:

- Have a quality management system equivalent to ISO9000
- Perform hazard analysis to determine safety integrity level
- Define a development process which is appropriate for the safety integrity level. Many recommendations with regard to appropriate development techniques are made
- Have appropriately trained personnel
- Make product measurements
- Keep a maintenance and operational log
- Produce an overall safety case

Although not yet issued as a final release this document has been very influential.

4.2. MISRA Guidelines

In November 1994, in-between the first and second drafts of IEC 1508, the UK's Motor Industry Software Reliability Association (MISRA) published their “Development Guidelines For Vehicle Based Software”, known as the MISRA Guidelines [MISRA].

The MISRA consortium was formed in response to the UK Safety Critical Systems Research Programme, supported by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council. The consortium was formed because there was a perceived need for a unified approach to software development, using agreed techniques, across the automotive industry. The MISRA consortium consisted of:

Vehicle manufactures	Ford Motor Company Ltd Lotus Engineering	Jaguar Cars Ltd Rover Group Ltd
Automotive suppliers	AB Automotive Electronics Ltd Delco Electronics	AP Borg & Beck Lucas Electronics
Consultants	The Centre for Software Engineering The Motor Industry Research Association	Rolls-Royce and Associates Ltd The University of Leeds

The Guidelines are an 82 page document giving recommendations covering the whole of software development. The purpose of the Guidelines is to assist the automotive industry in the creation and application within a vehicle of safe, reliable software. The content includes:

Software lifecycle	Safety analysis leading to integrity level Project planning Design Testing	Development practices Requirements specification Programming Product support
Software quality planning	Management responsibilities Human factors in software development Documentation requirements	Education and experience Quality assurance Subcontracting
Emerging Technologies	Neural networks Fuzzy logic	Object orientation Formal mathematical methods

In addition to the Guidelines there are also 8 separate reports containing supporting information:

Diagnosics & Integrated Vehicle Systems	Integrity
Noise, EMC and Real-Time	Software in Control Systems
Software Metrics	Verification and Validation
Subcontracting of Automotive Software	Human Factors in Software Development

The Guidelines are endorsed by:

- UK Department of Trade and Industry
- UK Department of Transport
- The Society of Motor Manufacturers and Traders Ltd

The Guidelines were written in the light of, and as far as possible sought to be compatible with, the first draft of IEC 1508. In common with IEC 1508 they require a hazard analysis to determine the safety integrity level and then a development process appropriate to the safety integrity level. Unlike IEC 1508 they do not make comprehensive recommendations of particular techniques for particular safety integrity levels and they do not require a safety case to be written.

4.3. Safety Integrity Levels

Some of the systems currently fitted to vehicles have the potential to harm vehicle occupants and other road users should they fail. Some systems have a greater potential for harm than others, for example a full authority cruise control system has greater potential for harm than an engine management system which only has idle speed control. This potential for harm will increase for the future type of systems described previously. Both IEC 1508 and the MISRA Guidelines use the concept of a scale of Safety Integrity Levels (SILs) to handle safety implications of the systems under consideration. The MISRA scale is 0 to 4 with 0 being systems with no safety related implications and 4 being those with the worst case implications. While IEC 1508 also has a scale starting at 0 for systems with no safety implications, the two scales are not necessarily equivalent.

It is generally thought that most current automotive systems are in the range SIL 0 to SIL 2. There is no published material giving SIL values for standard systems such as engine management systems or anti-lock break systems because the assessment technique is still quite subjective and the value will dependent upon the nature of the design. Also, if standard assessments were published, it may discourage developers from performing their own analysis. This would not be a good thing because performing the analysis gives the developer a much greater understanding of their system and its safety implications.

The greater the potential for harm the greater the confidence required that this harm will not come about. A system's integrity is a measure of the confidence we have that it will not fail in such a way as to bring this harm about. Therefore the Safety Integrity Level for a system is a measure of its required integrity given the potential effects of its failure.

There are two types of failure that the system can experience. These are systematic failures, arising from design errors, and random failures, typically due to some sort of physical failure. Software is not subject to random failures in the same way as a mechanical part but is susceptible to systematic failures due to design errors.

Therefore to gain confidence that design errors are unlikely, the software must be developed using a robust process, and the greater the confidence required (i.e. the higher the safety integrity level) the more robust the process must be.

4.4. Process Rigour

In order to achieve development processes which are more and more robust it is necessary to use increasingly precise design techniques. As an example Table 2 gives the MISRA Guidelines recommendations for development techniques for the different Safety Integrity Levels, 0 to 4, where 0 represents those with no safety implications and 4 those with the worst case safety implications.

Development Process	Integrity Level				
	0	1	2	3	4
Specification and design	I S O 9 0 0 1	Structured method.	Structured method supported by CASE tool.	Formal specification for those functions at this level.	Formal specification of complete system. Automated code generation (when available).
Languages and compilers		Standardized structured language.	A restricted subset of a standardized structured language. Validated or tested compilers (if available).	As for 2.	Independently certified compilers with proven formal syntax and semantics (when available).
Configuration management: products		All software products. Source code	Relationships between all software products. All tools.	As for 2.	As for 2.
Configuration management: processes		Unique identification. Product matches documentation. Access control. Authorized changes.	Control and audit changes. Confirmation process.	Automated change and build control. Automated confirmation process.	As for 3.
Testing		Show fitness for purpose. Test all safety requirements. Repeatable test plan.	Black box testing.	White box module testing - defined coverage. Stress testing against deadlock. Syntactic static analysis.	100% white box module testing. 100% requirements testing. 100% integration testing. Semantic static analysis.
Verification and validation		Show tests: are suitable; have been performed; are acceptable; exercise safety features. Traceable correction.	Structured program review. Show no new faults after corrections.	Automated static analysis. Proof (argument) of safety properties. Analysis for lack of deadlock. Justify test coverage. Show tests have been suitable.	All tools to be formally validated (when available). Proof (argument) of code against specification. Proof (argument) for lack of deadlock. Show object code reflects source code.
Access for assessment		Requirements and acceptance criteria. QA and product plans. Training policy. System test results.	Design documents. Software test results. Training structure.	Techniques, processes, tools. Witness testing. Adequate training. Code.	Full access to all stages and processes.

Table 2 MISRA Guidelines: Development Techniques versus Safety Integrity Levels

4.5. Determining Safety Integrity Levels

Both IEC 1508 and the MISRA Guidelines require the developer to determine the safety integrity level of the system they intend to develop in order to be able to define an appropriate development process. Both documents recommend performing a hazard analysis to determine system failures with the potential for harm. From a knowledge of the potential failures the safety integrity level has to be determined.

IEC 1508, which has its origins in the process industry, advocates defining the sequence of events from the failure to the point where someone is injured and then assigning probabilities to these events occurring. The value of the overall probability together with the severity of the injury are used to determine the safety integrity

level. The relatively static environment of a process plant means that it is possible to assign values to the probabilities. However this approach is inappropriate for systems which do not have a static environment because it becomes impossible to assign meaningful values to the probabilities.

A vehicle is a good example of a system which does not have a static environment, in fact the potential environment is practically infinite. For example the following are just some of the factors which would need to be taken into account:

- nature of road, e.g. housing estate access, main road, motorway
- weather conditions
- experience of driver
- skill of driver
- traffic density
- nature of vehicle in close proximity, e.g. passenger vehicle or lorries
- proximity of pedestrians
- maintenance state of the vehicle

Therefore the MISRA Guidelines have adopted the concept of **controllability** as a means of determining safety integrity levels for systems which do not have a static environment. The concept of controllability was first developed in the DRIVE Safely Project [DRIVE]. Although space does not permit a full explanation of the concept of controllability, an indication of how it works will be given.

Rather than asking the question “What may happen as a result of a failure?”, the controllability approach asks the question “How would the failure diminish the ability of those present (driver and passengers) to avoid a hazardous situation?”. Most of the hazards arising in a vehicle context are related to the movement of the vehicle. In particular a moving vehicle has a large amount of stored energy which if not correctly controlled may lead to an accident. The greater the loss of control the more serious the failure. Therefore the more the failure removes the ability to control the vehicle the higher the controllability category. This principle can also be applied to situations where the vehicle is not moving, for example failures of the security system, although the application is perhaps less intuitive.

The formal definitions of the controllability categories given in the MISRA Guidelines are shown in Table 4.

Uncontrollable:	This relates to failures whose effects are not controllable by the vehicle occupants, and which are most likely to lead to extremely severe outcomes. The outcome cannot be influenced by a human response.
Difficult to control:	This relates to failures whose effects are not controllable by the vehicle occupants, and which are most likely to lead to extremely severe outcomes. The outcome cannot be influenced by a human response.
Debilitating:	This relates to failures whose effects are usually controllable by a sensible human response and, whilst there is a reduction in safety margin, can usually be expected to lead to outcomes which are at worst severe.
Distracting:	This relates to failures which produce operational limitations, but a normal human response will limit the outcome to no worse than minor.
Nuisance only:	This relates to failures where safety is not normally considered to be affected, and where customer satisfaction is the main consideration.

Table 3 definitions of Controllability Categories

Having determined the controllability category there is a simple mapping from this to the safety integrity level as shown in Table 4.

Controllability Category	Safety Integrity Level
Uncontrollable	4
Difficult to control	3
Debilitating	2
Distracting	1
Nuisance only	0

Table 4 Mapping of Controllability Category to Safety Integrity Level

This approach of considering to what extent the failure removes the ability to control the situation is also advocated by those developing air traffic control systems where again the environment is not static [Smith 97].

5. Product and Process Assessment

Although a system may have been developed according to some best practice standard, for systems with higher SIL values it becomes necessary to independently confirm that this is so. IEC 1508 highly recommends independent assessment for systems at SIL 3 and SIL 4. MISRA also recognises the need for assessment. However assessing software is notoriously difficult. A recent CEC project, CASCADE, has specifically addressed this problem.

5.1. CASCADE Introduction

CASCADE (Certification and Assessment of Safety-Critical Application Development) was a multi-national project funded by the CEC under the ESPRIT III programme. CASCADE addressed the certification and assessment of software intensive safety critical systems in the railway and automotive industries. The project came to completion in January 1997.

The CASCADE partners are:

Lloyds Register	Assessor partner - Project Manager	TUV Rheinland	Assessor partner
INRETS	Assessor partner	Danish State Railways	Developer partner
Matra Transport	Developer partner	Railtrack	Developer partner
Rover Group	Developer partner	TA Consultancy	Developer partner

The principal deliverable of CASCADE is the Generalised Assessment Method (GAM) for assessing software intensive safety critical systems. The GAM was developed using real projects supplied by the developer partners as case studies which were assessed by the assessor partners. The GAM is concerned with both process and product evaluation and addresses the following issues:

- Assurance of the correct integrity level of safety critical software.
- The role of a safety case and its contents.
- The types of demonstration (testing, proof, audit, inspection, etc) to be used for the assessment of safety-related software.
- The procedures, methods and techniques for software assessment.
- Common criteria and rules to adopt.
- Application proof of conformity to the relevant standards for certification purposes.

The GAM is intended to be applicable to any system and be repeatable and objective.

5.2. CASCADE Results

The main CASCADE deliverable, the Generalised Assessment Method, is contained in three formal documents:

- GAM Part 1: Rules
- GAM Part 2: Guidelines
- GAM Part 3: Examples

Part 1: Rules contains a procedure for performing an assessment:

- Planning the Assessment

- Specifying the Requirements for the Assessment
- Assessing the Process used to develop the Product
- Assessing the Product
- Documenting the Findings in a Report

It also gives general principles which must be fulfilled for all assessments. This is the heart of the assessment method. For each of the general principles more detailed criteria must be defined by the assessor. These criteria are based on the appropriate sector standard being used by the developer (MISRA Guidelines in the case of Rover) taking into account the nature of the product being developed and the development environment. During the assessment, evidence of these criteria is sought to determine if the general principle has been fulfilled. In this way the generic approach can be applied to many different types of products and development environments.

The developer is also required to produce a safety case giving their overall justification of the safety of the product.

Part 2: Guidelines contains various items of helpful information to the assessor, including how to deal with gaps and ambiguities in the standards.

Part 3: Examples contains some of the results of the case studies used to develop and validate the GAM.

5.3. CASCADE and MISRA

The projects provided by Rover Group for use as case studies were developed in accordance with the MISRA Guidelines. The work of the case studies not only helped develop the GAM but also validated the MISRA Guidelines as being a suitable standard for automotive applications.

5.4. Pre-requisites for an Assessment

From the experience gained from involvement in the CASCADE project, there seems to be a minimum set of working practices which must be used by the developer in order for a third party assessment to be viable. These requirements come from two sources. Some are necessary simply for the assessor to perform their work, e.g. documentary evidence must be kept. Others are common requirements of all standards which if not followed will inevitably cause the product to fail the assessment. Some fall into both categories.

The following are suggested as the minimum set of working practices necessary for a third party assessment to be viable:

- The developer must have a quality management system compliant with ISO9000-3 [ISO9000] or equivalent;
- There must be documentary evidence (e.g. design documentation, project plans, meeting minutes) which the assessor can inspect to verify that stated working practices have actually been followed;
- The developer must have been working to some recognised sector standard, e.g. MISRA;
- The developer must have been working to some form of documented lifecycle;
- There must have been some independence with regard to the staff who perform the following activities, i.e. the engineer must not check their own work:
 - Verification and Validation
 - Quality Control
 - Safety Analysis
- Safety issues must have been explicitly addressed;
- Some form of hazard/safety analysis must have been performed to determine risk and appropriate safety integrity level;
- Some argument must be written down to support the claim that the product is acceptably safe, e.g. safety case.

Whereas developers of systems at IEC 1508 safety integrity levels 3 and 4, e.g. aviation, nuclear power, railway signalling, are used to working this way, most of the automotive industry, working on systems at the lower safety integrity levels, are not working this way. If the automotive industry is going to develop more advanced systems, then these new working practices will have to be adopted in order to develop, and be seen to develop, acceptably safe systems.

6. Conclusion

Assuming that the automotive industry will continue to develop ever more advanced systems, both ones which are confined to a single vehicle and those which interact with other vehicles, then as an industry we must progressively adopt the measures described in this paper. There is now wide agreement on what is bad practice and converging agreement on what is good practice. Best practice may still elude us but as an industry we ought to be adopting the documented good practice.

In brief, what this means in practice is:

- Have a quality management system compliant with ISO9000-3 [ISO9000] or equivalent;
- Perform hazard analysis to determine the safety integrity level using the controllability technique;
- Be compliant with a sector standard, e.g. MISRA;
- Use third party assessment, e.g. CASCADE GAM, for systems at safety integrity levels 3 and 4.

We should do this to help maintain the good record of the automotive industry with regard to the safety of its products. We should do it to demonstrate publicly that we are doing all that is reasonably possible. We can not afford to make a mistake.

7. References

- [IEC 1508] Draft IEC 1508 - *Functional safety: safety-related systems* - June 1995. Ed.1
Geneva: International Electrotechnical Commission (IEC reference 65A Secretariat 123)
- [MISRA] The Motor Industry Software Reliability Association *Development Guidelines For Vehicle Based Software*, published by The Motor Industry Research Association, Watling Street, Nuneaton, Warwickshire, CV10 OTU, United Kingdom, ISBN 0 9524156 0 7
- [DRIVE] *Towards a European Standard: The Development of Safe Road Transport Informatic Systems, Draft 2*, DRIVE Safety (DRIVE I Project V1051) March 1992.
- [CASCADE] *CASCADE Summary*. Ricardo Hetherington (Lloyd's Register). February 7 1996, ref. CAS/LR/RHZ/R480/2 - Public.
- [Smith 97] Smith J.U.M. *Using a Layered Functional Model to Determine Safety Requirements*, in Proceedings of the Fifth Safety-critical Symposium, Redmill F. and Anderson T., editors, Springer 1997.