

Safety-Critical Systems in Rail Transportation

Transportation systems, including train and tram, use electronics for subsystems previously controlled mechanically or manually. As a result, rail systems deliver far greater flexibility and are capable of real-time adjustments for speed, route, and passenger comfort. While these new electronic control and monitoring systems proffer many benefits to assure their safe operation, regulations mandate that such systems comply with industry standards for hardware and software development and are thoroughly tested and documented.



*Figure-1
London Underground*

In other industries, similar evolutions have taken place. For example, in avionics, safety-critical electronic systems have been subject to stringent requirements for government approval before they can be used in commercial aviation for decades. Similarly, medical devices are regulated in Europe by IEC-60601 and IEC-62304 standards to assure the correct operation of medical device software.

In rail transportation, as more electronic systems come into play, it becomes necessary to do whatever is possible to assure correct operation of these advanced systems. This very point is exemplified by the Altona Railway system malfunction in 1995.

“The German Railway attempted to replace its long established railway switch tower at Hamburg-Altona station by a fully computerized system made by Siemens. Immediately after starting the new systems, the central computer failed. Siemens' experts could not find the cause for hours, so German Railway decided to close the whole station, forcing thousands of passengers to start from locations up to 25 kilometers

away. The search for the fault was difficult as it was very rare. Two full days later, experts detected that under certain conditions a stack overflow happened. When looking into the routine which should handle stack overflows, they found that this went into a deadlock due to a programming error. In a press conference, the responsible Siemens manager argued that the "hidden" faults were difficult to find, and that Siemens experts had assumed that the routine handling stack overflow would NEVER be used!"

---- As described by Klaus Brunnstein (Univ. Hamburg, 17 March 1995)

Software used in safety-critical systems is, of course, a key element in the correctness of the system's operation. Most commonly, this software consists of an application running on top of an operating system.

In the case of Hamburg-Altona station, they would have been well served to use a commercial RTOS with the ability to analytically determine the worst-case maximum stack size needed by the application. Had they used such a tool, they might have averted the system failure entirely. Tools, such as Express Logic's StackX, can determine maximum stack size requirements through analysis of an .elf executable file.

Bahn-Sicherheitsstandards ergänzen IEC 61508

Because it's critical that electronic systems operate safely and because incidents such as the one in the Hamburg-Altona station, various government-sponsored agencies and independent technical standards organizations have become involved in defining regulations for safety-critical systems. The International Electrotechnical Commission (IEC), a worldwide organization for standardization, promotes international co-operation on all questions concerning standardization in the electrical and electronic fields. IEC-61508, the international standard for electrical, electronic and programmable electronic safety-related systems, sets out the requirements for ensuring that systems are designed, implemented, operated and maintained to provide the required safety integrity level (SIL).

In Europe, CENELEC—the European Committee for Electrotechnical Standardization—governs European railway standards, and these standards are beginning to make their way into the North American railway and public transport market as well. The CENELEC standards EN 50126, EN 50128 and EN 50129 are typically applied to define appropriate safety analysis for such systems:

- EN 50126 deals with Reliability, Availability, Maintainability and Safety for the entire railway system.
- EN 50129 applies to safety-related electronic control and protection systems.
- EN 50128 applies to safety-related software for railway control and protection systems.

IEC standards are applied at various "Safety Integrity Levels" (SIL), representing varying degrees of criticality based on the system's use. IEC EN 61508 outlines the toleration of a probability for failure at each level with the most critical aspects of the system (i.e., SIL 4) having the least tolerance for failure. As detailed in Figure-2, the standard defines both a system's PFD (Probability of Failure on Demand) and RRF (Risk Reduction Factor) for each SIL level. The term Probability of Failure on Demand (PFD) means the likelihood that the system will fail when asked to perform a particular operation for which it is designed. The Risk Reduction Factor (RRF) is the amount of risk that can be reduced by implementing the corresponding SIL system.

SIL	PFD	RRF
1	0.1-0.01	10-100
2	0.01-0.001	100-1000
3	0.001-0.0001	1000-10,000
4	0.0001-0.00001	10,000-100,000

Figure-2
PFD and RRF for SIL Levels

Common Regulatory Elements

All international safety-critical software standards incorporate common elements that apply to all software systems, regardless of their end application. While the different standards have their own particular phraseology and individual features, they all generally require that software is developed according to a well-documented plan, and that its operation is consistent with the plan.

In particular, safety-critical software must demonstrate, through rigorous testing and documentation that it is well designed and operates safely.

- **Process.** The process through which the software was designed, developed, and tested must be fully described and shown to be consistent. This is broken down into several sub-categories:
 - Planning – the objectives of the system are stated, along with the plan for achieving and verifying that these objectives have been achieved.
 - Design – the system design is specified, including hardware and software, with theory of operation and other aspects of design that enable examiners to understand how the system intends to achieve its objectives.
 - Development – the development process, including the tools used, code reviews, test plan, documentation, and staff training.
 - Requirements – the functional requirements of the system are explicitly identified and correlated with the system capabilities.
 - Verification – the process of assuring that the system performs in accordance with the specifications, and that it achieves its objectives.
 - Configuration management – control of incremental revisions over time, enabling reproducibility of results and protecting against the introduction of faults that cannot be backed out.
 - Quality assurance – processes and procedures that assure that the system has been developed and produced in accordance with its goals, and that it delivers the capabilities it is intended to provide.
- **Code** refers to the source code produced by the developers or development tool and it includes all system and application source code, test code, scripts, and object code. This code is to be reviewed as part of the regulatory compliance process, and must agree with the actual code used in the system.
- **Test** includes specific tests performed to verify the correct operation of the code, as well as its ability to achieve all design goals and system requirements. Testing includes code coverage and analysis to insure that all program instructions are tested. Finally, unit/white-box, integration/black-box, and final acceptance testing generally are included.
- **Results** consist of complete results of all tests compiled into a unit and integration test report.

Manufacturers of rail transportation systems have development teams that are fully capable of generating the documentation required to comply with these safety-critical standards, as required by applicable regulations and have done so for years.

However, there are some aspects of this work that developers would like to avoid or that pose challenges. As safety-critical systems evolve in complexity and make use of more powerful microprocessors, they increasingly employ commercial RTOS technology. The real-time operating system controls and manages the application software to maximize system resources for a given processor.

Generally, such applications involve multiple application tasks, or threads, and a priority-based real-time scheduling OS, with interrupt capability for real-time responsiveness. A commercial RTOS provides these functions with an easy-to-use application programming interface (API) that can save developers substantial time in product development. A commercial RTOS also enables developers to use other commercial middleware components such as network stacks, graphics, USB communication, and more.

If a commercial RTOS is used in a safety-critical system, the developer is challenged to comply with the regulatory requirements that call for documentation and testing of software developed by an independent entity. Regulators refer to this type of software as "software of unknown pedigree" (SOUP). As difficult as it is to provide all the required documentation for regulatory compliance for the application code that the development team created themselves, it is that much more difficult to do the same for code developed by another organization. In addition, the time required to generate and organize all the RTOS-related documentation can represent a significant portion of the project schedule. The challenge of documenting and the time involved become even more difficult if the commercial RTOS used was not delivered with full source code.

These concerns cause some development teams to use an in-house developed OS instead of a commercial product, which certainly eliminates the problem of SOUP and makes the documentation task that much less demanding. But this choice also sacrifices the advantages of the commercial RTOS, so the developer is left in a quandary.

If the RTOS is delivered with full source code and is manageably small, then it might be feasible for the development team to complete the regulatory tasks internally. However, to accomplish this, the team requires a fair amount of time to understand the outside code sufficiently well that they can document and test it sufficiently to satisfy regulatory review. This time and effort – not to mention the risk of error and failure – still leave this approach as less than satisfactory since it requires additional time and increases overall development costs.

For many companies, the solution for certifying RTOS and tool technology comes in using a 100% turnkey solution geared to meet the demands of regulatory compliance. Developers then can keep their attention focused on their own application, which they know well, and all documentation to meet regulatory compliance for external software is provided by the RTOS company. Using such a turnkey solution also eliminates the risk of error and failure through the guarantee of successful regulatory approval.

Express Logic's new Certification Pack™, for example, provides exactly this comprehensive solution to the dilemma of in-house versus commercial RTOS. The Certification Pack consists of all the RTOS design, code, test, and results documentation that each standard requires, fully prepared and guaranteed to be accepted by the governing agency. Figure-3 shows the contents of a typical Certification Pack for IEC-61508.

• Software Safety Requirements
• Software Safety Validation Plan
• Development Plan
• Configuration Management Plan
• Quality Assurance Plan
• Verification/Test Plan
• Coding Standards
• Requirements Standards
• Design Standards
• Requirements Specification
• Design Description
• Unit Test Procedure
• Unit Test Plans
• Unit Test Reports
• Integration Test Procedure
• Integration Test Plan
• Integration Test Report
• Trace Matrices
• Configuration Index
• Software Accomplishments Summary
• Safety Manual

*Figure-3
Contents of a typical IEC-61508 Certification Pack*

Delivered 100% complete, the Rail and Transportation Certification Packs are ready to submit for IEC-62679, EN-50128, IEC-61508, 49CFR236 Subpart H certification, and are TUV and CENELEC approved and have been proven for certification of devices up to and including SIL 3/4. Rail and Transportation Validation Suites contain everything needed to comply with IEC-62679, EN-50128, IEC-61508, 49CFR236 Subpart H: designs, source code, test code, test results, trace matrices and all related documentation for certification up through SIL 3/4.

Recommendations

Developers should make the trade-off between efficient use of in-house personnel for a certification effort versus the cost savings of using a commercial package prepared for the RTOS. Commercial solutions for the RTOS should be 100% turnkey, and are available for virtually all safety-critical software standards in the areas of medical, avionics, industrial, and transportation. Developers who wish to tackle part of the effort in-house are able to do so as well. For those developers, a partial set of material is provided and the in-house team typically might opt to perform target testing internally.

As we move forward as a technology-rich society, we can expect to see more safety-critical requirements intended to reduce the number of system failures that lead to injury and loss of life. Turnkey regulatory compliance solutions will likely become the norm, and will help greatly in the development of quality, well-proven embedded software.

About the author:

John A. Carbone, vice president of marketing for Express Logic, has 35 years experience in real-time computer systems and software, ranging from embedded system developer and FAE to vice president of sales and marketing. Prior to joining Express Logic, Mr. Carbone was vice president of marketing for Green Hills Software. Mr. Carbone has a BS degree in mathematics from Boston College.