

A Product-Based Assurance Model for Mixed-Integrity Markets

Brenton Atchison and Alena Griffiths

Invensys SCADA Technology Group

brenton.atchison@invensys.com, alena.griffiths@invensys.com

Abstract

Many markets use a Commercial-Off-The-Shelf (COTS) or product-based approach to engineering in order to reduce project cost, schedule and risk, take advantage of product maturity and secure long-term support. The product-based approach presents challenges for both product developers and project engineers when applied to safety-related applications. Project engineers are obliged to present evidence of product integrity to support the overall safety argument. In such cases, the safety integrity requirements for a product may not be known until a safety analysis of a specific system architecture in its target environment is performed. Once determined, evidence of integrity needs to be obtained and presented to suit the customer requirements and industry standards. Concurrently, product developers need to engineer products and assurance evidence to support the requirements of high-integrity markets in the face of constant product change and the competing demands of different markets.

This paper discusses the issues involved in engineering products for use in Supervisory Control and Data Acquisition (SCADA) systems in a diverse range of applications, both safety-related and non-safety-related. In particular, we address the issue of how to provide a base level of product assurance that can be used, if it ultimately proves necessary, to support system safety cases.

Keywords: Safety-related systems, COTS, SCADA

1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems are a class of control systems used in a variety of applications to provide centralised control over a geographically diverse area. SCADA systems are generally produced in a COTS-like development model, with base products customised and configured for each customer.

Although they rarely directly control safety-critical functions, such systems may be relied on to assist in

hazard management and are often vital to the management of critical civil infrastructure.

This paper builds on a previous paper (Atchison and Griffiths 2002) and discusses the development and assurance of products for SCADA systems. Section 2 provides an overview of SCADA systems and the products used to engineer them, and discusses SCADA system safety requirements in different application domains. Most companies who supply SCADA systems develop a number of SCADA products, which are synthesized as required to build systems. Section 3 explains why this product-based approach is unavoidable but also points out assurance issues that arise when one seeks to develop a product for use in a diverse range of systems. Section 4 discusses an approach to the “product problem”, which does not involve product certification. The approach is outlined in more detail than previously discussed and is based on the combination of strategies in product architecture, testing and change management.

2 SCADA Systems and Safety

This section provides an introduction to SCADA systems, outlines the products used to build such systems, and discusses the safety integrity requirements that can attach to SCADA systems in different application domains.

2.1 SCADA Systems – An Overview

SCADA systems are generally characterised on the basis of their architecture and control paradigm (Landman 2000). A typical SCADA system architecture is illustrated in Figure 1. The rounded rectangles represent standard SCADA system components, the ovals represent optional, additional SCADA components often considered to form a part of modern SCADA systems. The rectangles represent external components with which SCADA systems sometimes interface.

In essence, a SCADA system consists of one or more *master stations* located in a control centre, with connections to many *remote terminal units* (RTUs) that are physically connected to plant. Alternatively, the system interfaces with third-party external devices (IEDs). The SCADA system enables an operator at the control centre to monitor and control plant that is close to the RTUs. The master stations are sophisticated information systems, usually implemented on platforms such as Unix or Windows, while the RTUs are simpler computing devices built on custom hardware, including custom I/O hardware.

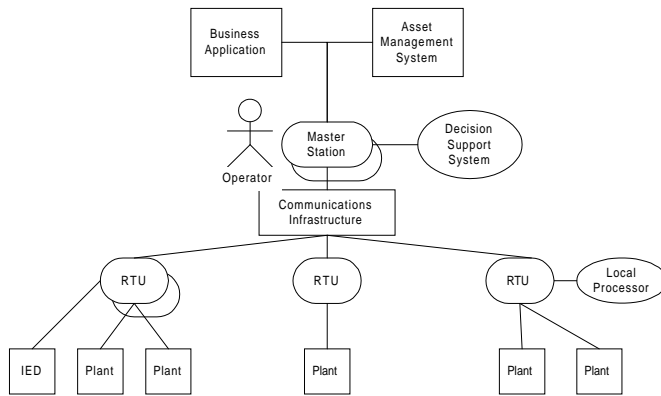


Figure 1 - Typical SCADA Architecture

RTUs can be spread over very large distances and are connected to the master stations by a communications infrastructure. Communications media can include dedicated wide-area-networks, radio links or public telecommunications networks. The physical communications media is typically not considered to be part of a SCADA system and a characteristic of SCADA is its ability to operate and recover in the context of unreliable communications.

Controls are issued from the master station and reach the plant via the connecting RTU. Information from the plant is telemetered back to the master station, again via the RTU. SCADA systems are characterised by an open-loop control paradigm, that is, control decisions are usually made by an operator based on the state of the plant, as indicated by displayed telemetered data. Operators are generally only present at the control centre, and one of the main functions the master station performs is to provide a suitable human machine interface that allows the operator to rapidly assess the overall state of the plant and so to make decisions about suitable controls to issue.

Modern SCADA systems often contain features that extend the control paradigm by providing local processing at the RTU and additional automation at the master station. Support for sophisticated local processing at the RTU can be useful to distribute processing load or to allow continued operation of plant during communications outages with the Master Station. Advanced master station applications such as decision support systems provide advice to the operator about controls s/he should issue, based on a computer-based analysis of the state of the plant. Such support systems are sometimes permitted to run in semi-automatic mode in which case controls are issued directly by the system, with the operator able to intervene if necessary.

There is also a growing trend toward the integration of business enterprise systems with SCADA systems. Examples of systems to which SCADA systems may interface include asset management systems used to optimise the owner's investment in their asset, and enterprise management applications, where SCADA information is used to run a business, for example record electricity or gas flows to different customers.

2.2 SCADA Products

Although every SCADA installation is unique, and systems differ substantially between application domains, there are nonetheless a number of common building blocks used in every SCADA system. For this reason, most companies that supply SCADA systems develop and maintain a number of SCADA products. The main products are the RTU and the master station. RTU and master station products are typically developed separately and may, in turn, be comprised of modules that can be acquired separately.

The base products can be configured in a large number of ways by the systems integration activity. Each application will use a different number of RTUs and master stations to match the distribution of their field equipment and requirements for load balancing and redundancy. Other configuration parameters include:

- 1) selection of hardware architecture, including modules to connect to field equipment and communications infrastructure;
- 2) mapping of field data and controls to a data and communications models in the RTUs and master station;
- 3) customisation of displays and the definition of alarm conditions;
- 4) production of applications or RTU logic and integration with other information systems.

The remainder of this section discusses the use of SCADA systems in some different application domains, and the safety integrity requirements that attach to such systems in these different domains.

2.2.1 Rail

In the Rail Industry, uses of SCADA systems include:

- 1) Traction power distribution and control;
- 2) Monitoring and control of environmental plant on underground railways, (e.g. chillers, smoke extraction fans, tunnel ventilation fans, etc.); and
- 3) An integration mechanism for many diverse systems (e.g. Passenger Address, Passenger Information Display, Closed Circuit Television, miscellaneous station plant control such as escalators, lifts, lighting, etc.)

The main hazard scenarios associated with traction power distribution and control are electrocution of trackside maintenance workers, and electrocution of emergency services personnel during some sort of emergency in the vicinity of high-voltage power equipment (e.g. derailment or incursions onto the track). In such scenarios, the SCADA system is often deemed not to be safety-related, because work safety procedures involving physical isolation and earthing of high voltage equipment are relied on to protect trackside workers or emergency services personnel. Where such procedures are not used, the SCADA system must be used to isolate the relevant

section and safety integrity requirements will probably attach.

In the case of environmental plant control, the SCADA system may be used to ventilate tunnels when a train is stranded or to extract smoke in the event of a fire. These functions generally give rise to safety integrity requirements. Sometimes such functions can be achieved via a station-based, local control panel, which means that the safety integrity requirements may be avoided or limited to the RTUs only.

Where SCADA systems are used to integrate other railway systems, the safety integrity requirements depend on the operational paradigm for the railway involved. While it is unusual for any particular function to give rise to a safety integrity requirement, it may be that the availability of an integrated picture of the current status of a range of railway subsystems is considered critical to the correct handling of emergency situations in general. As such, system availability becomes a safety integrity requirement.

In the rail industry, there is a general awareness of the potential safety implications of such systems. This is reflected in the fact that many contracts for the supply of SCADA systems will explicitly mandate a target safety integrity level (SIL) for the system (SIL 1 or 2 is typical).

2.2.2 Oil and Gas

Typical functions of SCADA systems in the oil and gas industry include:

- 1) Product flow measurement and control;
- 2) Product quality monitoring;
- 3) Monitoring of pipeline control equipment, such as gas compressor and valves; and
- 4) Integration with business systems to manage purchase and sales transactions.

SCADA systems are generally not considered safety-related in the oil and gas industry since independent safety and emergency shutdown systems will manage pipeline hazards. Nevertheless, SCADA systems can provide an early indication of emerging hazardous situations, such as overpressure or pipeline leakage, and is often used as the first means of hazard control.

Triggering of an emergency pipeline shutdown is a costly exercise and can lead to other indirect hazards stemming from the loss of energy supply. The SCADA system also provides an essential service in the early diagnosis of pipeline equipment failures, for example failure of a key compressor, so is a critical factor in the overall pipeline availability.

2.2.3 Power Distribution

SCADA systems are used extensively to manage our electricity supply. Key functions include:

- 1) Monitoring of substation current and voltage levels;
- 2) Monitoring of substation equipment and automatic recovery from some substation failure conditions;

- 3) Manual control of power supply;
- 4) Automatic regulation of transformers to achieve constant voltage under variable load conditions;
- 5) Isolation of substation equipment for maintenance;
- 6) Indication of personnel presence on site and alarming of emergency switch triggering; and
- 7) Load measurement, trending and forecasting.

Although a number of these functions are safety-related, SCADA is rarely used as the sole means for achieving them. For example, substation maintenance personnel are trained to physically isolate equipment before commencing maintenance. Despite not having direct responsibility for safety functions, it is possible that a SCADA fault or misuse, followed by a period of SCADA unavailability, could lead to insufficient power supply or loss of power.

2.2.4 Water Supply and Waste Water Treatment

Large towns and cities use SCADA systems to manage the supply of water and treatment of waste water. While the systems are different, many of the functions are similar and include:

- 1) Measurement of water reservoir and waste water storage tank levels;
- 2) Remote water flow control, including overflow of waste water in storm conditions and supply of drinking water in high demand periods;
- 3) Isolation of valves for pipe maintenance;
- 4) Monitoring of pumping station equipment and power supplies;
- 5) Isolation of pumping station equipment for maintenance; and
- 6) Monitoring of personnel presence in pumping stations.

Water supply and waste water utilities can generally be managed through manual equipment control, indeed not all parts of a utility need be automated. In some instances, large pumping stations are augmented by local closed-loop control systems that monitor and manage water flow. The use of SCADA for large utilities allows for a centralised view of the utility and a more effective and cheaper management response.

Inappropriate operation of SCADA can lead to overflow of waste water in civil areas or to shortage of water supply. It can increase the likelihood of burst pipes by causing fluctuations in pipe pressure, or the stagnation of drinking water in reservoirs or pipes. SCADA is also essential in the detection of utility equipment failures and the subsequent coordination of a management response.

2.2.5 Summary

Across different industries we see a broad range of safety integrity requirements that may attach to SCADA

systems. In most application domains, SCADA is not considered safety-related due to the presence of mechanical, procedural or independent electronic mitigations. Nevertheless, SCADA is often used as the initial detection and response mechanism for hazardous situations. Its advantage in providing centralised supervision and control of a complex system is important and unreliability or unavailability of the SCADA system may increase the overall risk of failed hazard management.

As our civil infrastructure grows more complex, SCADA systems are an integral factor in their effective management. As a result, system availability is often business critical, and system unavailability can have indirect safety implications. As such, many contracts for the supply of SCADA systems include clauses mandating quantitative reliability and availability targets.

Few application industries are as mature as the Rail Industry in terms of adopting a risk-based approach to safety integrity requirement determination (CENELEC 2001). However, increasingly, contracts involve penalty clauses for failure to meet target availability levels. Given that SCADA systems are heavily software-based, the issues associated with attempting to demonstrate compliance with quantitative reliability and availability targets can be similar to those faced in engineering software-based systems to an appropriate safety integrity level.

3 The Product Approach

As already mentioned most companies that supply SCADA systems maintain a number of SCADA products. Individual systems are engineered using these products. This section explains why there is no practical alternative to the product-based development model, and then goes on to discuss assurance problems associated with this model. A number of possible solutions are considered, but each is shown to have one or more obstacles to overcome.

3.1 Why a Product Approach?

Despite discussion in some forums about assurance pitfalls that may be associated with using COTS and open systems to implement safety-related functions, there is nevertheless a seemingly irreversible trend towards the use of such systems. In the Rail Industry this trend is evident despite an awareness of the safety assurance issues. In industries that do not acknowledge any safety integrity requirements for SCADA systems, the use of COTS products and open systems/standards is standard practice.

The reasons why purchasers prefer COTS products are well-documented (Lindsay and Smith 2000). In brief, using COTS can reduce development risk and can increase reliability due to product maturity. It is very difficult for a solution provider seeking to develop a system “from scratch” to compete with a solution provider who offers a system composed of pre-existing components. The lack of competitiveness exists on several levels, including price, and market credibility.

There is another reason why purchasers of SCADA products tend to prefer COTS. Most purchasers have a significant investment in some sort of asset (e.g. power distribution plant). Such assets are maintained over a long period of time and, since the SCADA system assists in management of that asset, customers require a SCADA solution that will continue to be supported long-term. Over time, just as the purchaser will want to upgrade their plant, so they will look to upgrade their SCADA system. As such, purchasing a system whose component products have large user-bases means that, because a product with a large user base is likely to be upgraded, so it is likely the purchaser will be able to upgrade their SCADA system by purchasing component product upgrades. For this reason, most purchasers would prefer a product with a broad user-base, even if many of the installations are non-safety-related, than a product with a small user-base which, although it specifically targets the high-integrity market, is nevertheless at greater risk of being discontinued.

In practice, this means that most contracts for the supply of large SCADA systems are won by companies who possess (a suite of) SCADA products, that implement open standards communications protocols. The business and market forces are such that there is virtually no alternative to a product-based approach to the provision of SCADA systems.

3.2 Product Model Assurance Issues

If a SCADA system is determined to be a safety-related system, a safety case will need to be developed for that system. If the SCADA system is composed from a number of standard SCADA products, it will be incumbent on the supplier to show that the SCADA products are fit for the purpose for which they are intended. Typically, this will mean showing that a product correctly implements, to a suitable level of integrity, a number of functions that are safety-related in that system context. In this section, we consider a number of strategies that can be used to show that a component implements a safety requirement to a suitable integrity level. For each strategy, we explain why problems can arise in the context of the product model.

There are a number of COTS assurance strategies that can be employed (O'Halloran 1999). We discuss how some of these strategies relate to SCADA.

3.2.1 High Quality Development Process

Many modern safety standards enable one to use evidence of a rigorous development process to support product integrity claims. Most companies developing SCADA products have generally developed them over a number of years and in response to an increasingly diverse set of requirements, including safety integrity requirements. During this period, the notion of best practice has evolved and development practices that were considered state-of-the-art 10 years ago would probably not produce evidence sufficient to support a modern safety case. As such, safety cases based exclusively on evidence associated with the product development process are difficult to support.

3.2.2 Operational Evidence

Claims of product integrity can sometimes be made given sufficient evidence of use in operation. However, most SCADA products evolve over time, with incremental releases, so it is difficult to build up sufficient operational time with each release. Furthermore, such products are configured differently in different systems and in any case exhibit a different operational profile so “proven-in-use” arguments are difficult to sustain. Some industries also display a parochialism that sees them reluctant to accept evidence of use in another industry as any guarantee for correct operation in the application industry.

3.2.3 Extensive Testing

Despite the bias in safety standards towards an assessment of the overall development process, many purchasers still consider evidence of extensive validation testing to be the best guarantee of correct performance. Extensive release testing is time-consuming, particularly because modern SCADA products can be extremely feature-rich. Moreover, SCADA products are also highly configurable, which adds another dimension to the test state space and compounds the problem further. This means that test-based assurance of correct operation of all product features in all system configurations will not be persuasive in most modern safety cases.

3.2.4 Safe Design Techniques

A common technique used when designing systems that perform many functions, only some of which are critical, is to partition the system so that critical functions are implemented in a safety kernel, and the remaining functions are implemented elsewhere. Safety assurance for the non-critical functions is limited to showing that malfunction can not adversely impact correct and continuing operation of the critical functions. This design technique is not straightforward to apply in the case of a SCADA product, where the product must be suitable for use in a variety of systems, and where the sets of features considered safety-related in different systems can differ quite markedly.

Another common design-for-safety technique is the fail-safe approach. This approach requires a known safe state exists, and then involves engineering the system so that in the event of any anomaly, the system will fail to this safe state. It involves a sacrifice in the availability of some non-critical functions, in order to ensure that system reliability from a safety perspective is increased. Unfortunately, this strategy does not translate well to SCADA product engineering, since a state that is “safe” in one application may not be safe in another. To amplify this, consider the issuing of controls to plant. In rail traction power distribution, where the key hazard scenario is electrocution of staff working trackside, the safety requirement is that no spurious controls (say to reclose a circuit breaker) occur during the period when maintenance is in progress. As such, in this application, “no controls”, whilst occasionally inconvenient, is nevertheless considered safe. Compare this with the

environmental plant control application, where it is critical to be able to issue controls to smoke extraction fans when required. In such applications, the possibility of an occasional spurious control would be far preferable to the risk of non-availability of controls when required.

3.3 Possible Solutions to the Product Problem

Against the backdrop of a market preference for a product-based approach and the assurance problems associated with adopting that approach, the following generic options exist.

3.3.1 Multiple Products

One solution is to maintain multiple products, each targeted to the functional and assurance requirements of a particular market sector. Apart from compounding the cost of ownership, such a strategy is also unlikely to be welcomed by purchasers who want to buy a product with a large user base, so as to be assured of continued support for the product line.

3.3.2 Project-based Safety Certification

This solution involves developing individual safety cases for each deployed system. This approach has the advantage of relieving projects that have no safety requirements of any safety assurance overhead. On the other hand, duplication of effort is likely to result. Also, since the safety assurance evidence is developed by and stays with the project team, future product evolution may render this evidence irrelevant. As such, where future system upgrades involve product upgrades, and because the product upgrades may not have occurred with that particular system’s assurance requirements in mind, safety case maintenance may become extremely expensive.

3.3.3 Product Certification

This solution involves achieving certification of a product to a certain safety integrity level. Typically, the certification is against a particular requirements baseline, and would involve the analysis, by an independent party, of the product development process and in-service performance. This approach has the benefit of significantly reducing the cost of producing system-specific safety cases (certification by a renowned third party is usually very persuasive). However, it also means that all product upgrades would need to be re-certified by the independent party, even those upgrades that were motivated by requirements for systems in non-safety-related domains. This places a considerable burden on the product release process, which may lead to product inertia. It also represents an overhead on the cost of the product as a whole, which may price the product out of certain markets.

4 Proposed Solution

This section proposes a solution to the problem of engineering a product that can be deployed across a

diverse range of industries, with diverse functional and assurance requirements.

4.1 Organisational Structure

The basis of the solution is to distribute responsibility for the assurance of SCADA systems across an engineering organisation. Responsibility for the certification of systems lies with the market-specific, project engineering teams but is supported by evidence provided by the product development group. Requirements for evidence are forecast by market-specific, sales and marketing teams. The distribution of responsibility is illustrated in Figure 2.

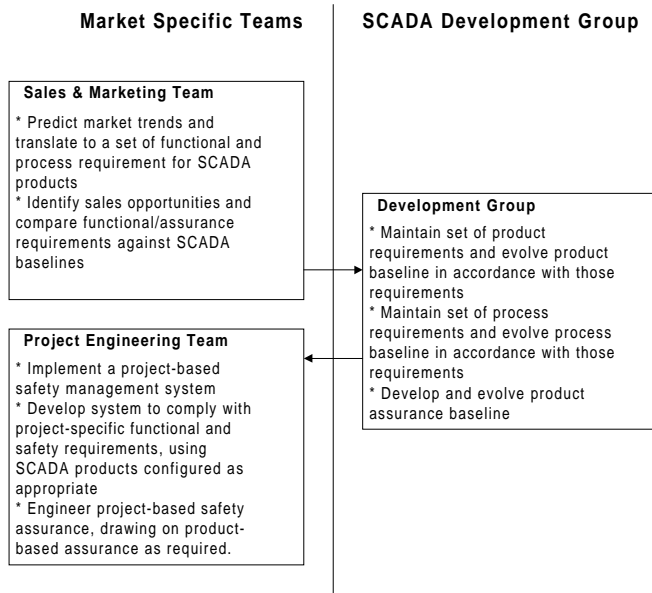


Figure 2 - SCADA Organisational Model

The proposed assurance model allows each project engineering team to build a safety case in a form suitable to meet specific market and project demands, using product assurance as necessary. This engineering group must fulfill two key roles in this model.

Firstly, the sales and marketing team are contributors to the set of requirements contained in the product baseline. This will involve identification of trends in the market related to safety. This can range from physical requirements on the product deliverable (e.g. a compulsory requirement for redundant CPUs on the RTU), through to requirements on the product development process (e.g. a trend towards the use of formal source code analysis techniques as a compulsory pre-requisite for SIL 2 certification).

Secondly, the project engineering teams are required to make application-specific safety cases to get their systems accepted into service. The safety case is prepared in cooperation with the SCADA system procurer and end-users and will address aspects of the SCADA system, as well as its operational use. Activities will typically include production of a hazard analysis to identify product safety requirements, with knowledge of the system architecture and operational context. A subsequent activity will then be performed to map the

product and assurance evidence into a safety case that satisfies sector specific standards.

The model requires the product team to develop and evolve the required product assurance in line with forecasted market requirements. While engineering groups may fund production of the assurance, the product-based approach exploits the fact that the costs of engineering the required product assurance can be amortized not only across many system sales, but also across the entire life of the product.

4.2 Extending the Notion of "Product"

In line with the idea of distributed responsibility for assurance, our proposed solution entails a broader definition of "product", and hence a broader job description for product development groups. For computer-based systems, there is a tendency to think of a product as consisting solely of a physical hardware platform and the executable software that runs on it. We proposed broadening this concept so that a product is considered to consist of three baselines, as illustrated in Figure 3.

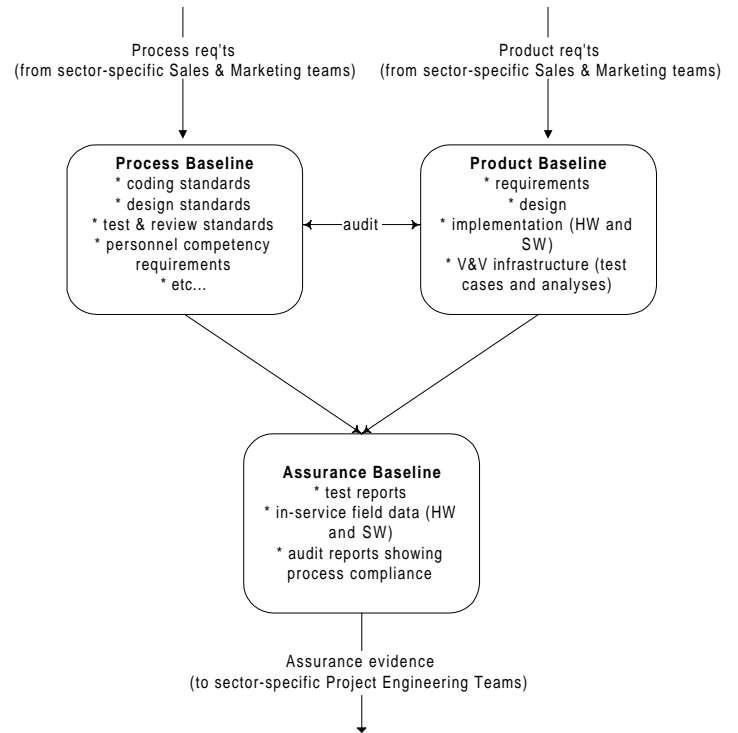


Figure 3 - Product Baselines

The **product baseline** comprises the traditionally maintained artefacts and comprises:

- 1) the product requirements (both functional and non-functional). This could include specific process requirements demanded by some sectors (e.g. requirement to demonstrate 100% statement coverage at unit testing),
- 2) the product design,
- 3) product deliverables (i.e. hardware and executable software),

- 4) associated verification and validation infrastructure (e.g. test cases, test harnesses, etc.).

The **process baseline** describes how the product baseline was developed and includes:

- 1) the suite of meta-processes under which development of the product baseline occurs, such as the Software Quality Plan, the Verification & Validation Plan, the Configuration Management Plan, etc.; and,
- 2) the suite of specific processes used for various development activities, such as Coding Standards, Defect Reporting Guidelines, Change Management Guidelines, Technical Review Guidelines, etc.

The **assurance baseline** provides evidence of the integrity of the product and process baselines, including:

- 1) Evidence to demonstrate that product deliverables comply with product requirements (e.g. test results, design analyses, traceability matrices, user trials, etc.);
- 2) Evidence to demonstrate that the development occurred in accordance with the process baseline (e.g. documented evidence of reviews, checklists, and quality audit reports);
- 3) Data, both quantitative and qualitative, which reflects the in-service performance of the product deliverables. This would include a range of information, spanning issues such as observed in-service MTBF of a hardware component, through to the safety case used to justify inclusion of the product in (say) a railway traction control system.

4.3 Generating the Evidence

For a product-based approach to safety-related systems engineering to be feasible, it must be possible to generate and maintain product assurance evidence at reasonable cost with minimal impact on delivery schedule and product advancement. A number of areas critical to achieve this are discussed below.

4.3.1 Product Architecture

A sound product architecture is the foundation for cost-effective product assurance. Key requirements of the architecture are to minimise critical functionality in both the product itself and in the deployed project applications and to provide an effective means of partitioning critical and non-critical functions. Such an architecture allows the product and project assurance efforts to be focussed on those areas requiring most rigour. Modification to the critical functions can be more tightly controlled and, as the product matures, operational evidence of stable critical functions can be accumulated more effectively. Similarly, non-critical functions can be developed under less stringent control in response to market demands or by market sectors themselves. This provides flexible, responsive product functionality without compromising critical functions.

A Master Station product with these attributes is currently under study in a joint venture by Invensys Rail and

Invensys SCADA Technology. The product comprises a core infrastructure that provide interprocess communication and redundancy management, and a set of services deployed to suit the target application. Each service controls its own data and data is exchanged via the shared infrastructure. Services may be classified as critical or non-critical. Data exchange from non-critical to critical components are prohibited and data exchange from critical to non-critical are controlled to ensure that processing load on critical services is restricted. To avoid interference through common hardware resources, critical and non-critical services may be physically partitioned by execution on separate machines.

An additional feature of the architecture is the division of the system data into groups, with each group managed by instances of the system services. Again, critical data groups and services can be partitioned onto different machines, with restricted communication between critical and non-critical service groups. Partitioning data in this way allows the project engineering group to focus its assurance efforts for activities such as system data and display configuration, and communications engineering.

4.3.2 Testing

Extensive testing is fundamental to product assurance arguments but, as mentioned in Section 3.2.3, the cost and time required for testing can be high for a feature-rich product under continuous evolution. While standards provide details of how to test new developments, there is limited guidance on the application to legacy software maintenance and evolution.

We propose that the testing be managed by a number of combined strategies:

1. Use a product architecture that allows impact of change to be minimised and well understood. Impact analysis is applied to each change to determine the scope of testing required.
2. For clearly localised changes, manual testing of specific changes may be sufficient.
3. Develop automated module and integration testing to allow for rapid detailed regression testing of key system components. Maintain the test suites to incorporate functionality changes or tests for previously undetected faults.
4. Where automated test infrastructure does not exist or is not effective, conduct manual testing.
5. Conduct validation testing focused against a number of “standard system architectures”, so as to reduce the possible configuration space needed for thorough testing.

4.3.3 Dealing with change

For any solution to be workable, it must be possible to evolve the product at reasonable cost. Along with the expansion of the notion of “product” to accommodate three distinct baselines, so change management practices must be sophisticated enough to cope with changes in each baseline. Each baseline must maintain an accurate

record of all product changes as well as evidence that product quality is preserved. Poorly conceived or rushed changes can easily render pre-existing test data useless.

For products with long life spans, change management must also address changes to the processes used to develop the product. This is an area that is not generally handled in texts on configuration management and change management. In general, applying new process requirements to the existing product baseline would involve major re-engineering and as such would be practically unfeasible. It is therefore necessary to find a way to apply new processes to new developments and major product modifications, while permitting minor modifications or “bug fixes” to occur via the processes that led to the original development. This approach to the issue of process evolution is endorsed in some standards (e.g. EN50128 (CENELEC 2001)). However, this approach also leads to a situation where product assurance derives from different sources both across the product, and over time. This needs to be carefully explained and justified in safety cases using the product assurance baseline.

Invensys SCADA Development are exploring the use of a tool for change management developed in a collaborative research project with the University of Queensland (Völzer, Atchison, Lindsay, MacDonald and Strooper: 2002). The tool models a product as a hierarchy of versioned subsystems, where each subsystem version is associated with a set of changes to configuration items, including source code files or documentation. The tool is built on existing commercial source code configuration management and change management utilities.

To address product assurance change management, the tool would be extended to include references to assurance evidence within a subsystem version. The subsystem-based approach is useful for safety-related products since it allows assurance records to be tailored at a subsystem level depending on the assurance strategy used. For example, a subsystem version may reference specific test cases executed for localised changes or evidence of complete automated subsystem test execution. The tool would be useful as a vehicle for change impact assessment by relating modified configuration items to affected subsystems and identifying invalidated assurance evidence. It would also collect measures of subsystem change over time as the basis for arguments of subsystem stability and reliability growth.

5 Conclusions

SCADA systems are used in many markets, but they are usually not thought of as safety-related systems. Section 2 presented an overview of SCADA systems and discussed the safety integrity requirements that may apply in different application domains. This analysis showed that, while SCADA systems are never solely responsible for performing safety-related functions, they occasionally perform safety-related functions (e.g. tunnel ventilation), and often contribute to early warning of and subsequent management of hazardous situations. Also, SCADA systems are relied on to manage the civil infrastructure

used to ensure energy and water supply, and so prolonged system unavailability or faulty system behaviour can compromise those supplies, which can in turn have safety consequences.

SCADA systems are usually composed from a number of standard SCADA products. Indeed, as was discussed in Section 3, companies seeking to supply SCADA systems have virtually no commercially viable alternative but to maintain a suite of SCADA products. This creates difficulties from a safety assurance perspective, since the products need to be fit-for-purpose in the face of diverse functional and safety integrity requirements. There are many parallels between the issues pertaining to the use of COTS in safety-related systems, and to the problems faced by SCADA system and product suppliers.

A number of solutions to this problem were briefly considered but determined to be unsatisfactory for various reasons. Instead, in Section 4, we proposed an alternative that sees responsibility for assurance distributed across an organisation. In this model, the product group’s responsibilities are expanded to involve maintenance of three baselines, and the application-specific project engineering groups use this information to build system and sector specific safety cases. Three essential features of this solution were examined, including a product architecture that partitions critical functionality, a testing approach that provides focussed rapid accumulation of assurance evidence and a change management system that deals not only with changes in a product’s functional requirements, but also with changes in a product’s assurance evidence.

6 References

- ATCHISON, B. and GRIFFITHS, A. (2002): Engineering SCADA Products for Use in Safety-Related Systems. *Proc. Safety Critical Systems Symposium*, Southampton, UK, Springer-Verlag.
- LANDMAN, R. J. (2000): Supervisory Control and Data Acquisition Systems
- CENELEC (2001): European Standard ENV 50128, Railway Applications - Communications, Signalling and Processing Systems - Software for Railway Control and Protection Systems,
- LINDSAY, P. and SMITH, G. (2000): Safety Assurance of Commercial-Off-The-Shelf Software. *Proceedings Fifth Australian Workshop on Safety Critical Systems and Software*, Melbourne, Australia, 43-51, Australian Computer Society.
- O'HALLORAN, C. (1999): Assessing Safety Critical COTS Systems. *Journal of the System Safety Society* 35(2):
- VÖLZER, H., ATCHISON, B., LINDSAY, P., MACDONALD, A. and STROOPER, P. (2002): A Tool for Subsystem Configuration Management. *Proc. ICSM 2002*, IEEE Press.