# Reliability of Smart Instrumentation

A Dobbing, N Clark, D Godfrey, P M Harris, G Parkin, M J Stevens and B A Wichmann,
National Physical Laboratory and Druck Ltd

September 4, 1998

## 1 Introduction

This paper gives an overview of a recently completed study undertaken for the UK Nuclear Industry on the Reliability of Smart Instrumentation, that is instrumentation which depends upon digital electronics and computer software for its correct operation.

In many critical contexts, high reliability is a requirement which must be justified from an analysis or actual data obtained from use of the device. The methods used to undertake this analysis for hardware, such as FMEA, are not directly applicable to software. Also, software is known to have failure modes and characteristics quite unlike hardware. Hence obtaining suitable reliability data for Smart instrumentation is not straightforward. For the background to this study, see [6].

The information is this paper is published with kind agreement of the companies responsible for the research contract, namely Nuclear Electric Limited, Scottish Nuclear Limited, British Nuclear Fuels Limited and Magnox plc. NPL and Druck Ltd acknowledge these companies support and thank them for their agreement to publish this paper (under contract IMC reference PC/GNSR/5001).

## 2 Study Method

Druck Ltd manufactures pressure transducers with digital electronics and software, i.e produces Smart instrumentation. This research study involved NPL following the actual design process of one of Druck's devices in order to investigate the reliability issues in depth.

It is logically possible to have no design defects in software so that any observed faults would be with the hardware. In many cases, the underlying hardware can be analysed to give theoretical reliability figures for the device. Hence the problem in using this conventional approach is to determine the most suitable value to ascribe to the reliability of the software components.

Software reliability has been extensively studied in the context of safety critical systems, since faults could give rise to loss of life. The conclusions from this work can be summarised as follows:

1. If the reliability requirements are modest, then the claims can be justified by systems testing of the device as a black box [5]. However, for higher reliability requirements, systems testing is inadequate.

2. The problems with quantifying reliability increase rapidly with the complexity of the system. For instance, data produced by IBM for large main-frame systems [1] showed that significant faults could occur every 5000 years of use!

3. When reliability claims cannot be justified from test results alone, safety standards accept evidence from the design process, see [3, 9, 4, 12].

4. Many validation techniques are very effective in locating faults, but there is no silver bullet [13]. One can design with the intent of having no faults, but never be totally sure that this has be attained (which is different from the approach in [7]).

The approach taken in this study is to quantify the risks of design faults and see which of the available validation techniques provides the best engineering solution.

One aspect of this work was immediately apparent. A user of the Smart instrument would not usually have any access to the design documentation. However, our study did have access to this, which was essential for most of the validation methods investigated.

The study itself was divided into a number of items which were either undertaken largely by NPL or Druck and reviewed by the other party. The actual items are listed in Appendix A, with a summary of the individual conclusions.

Several aspects need to be considered for all these study items which have influenced the work undertaken:

- Druck would not usually need to allow other parties to review its design documentation, and therefore has not been prepared for that purpose. The NPL staff involved were software engineers rather than instrument designers and therefore communication (in both directions) was not always straightforward.

- Druck's actual instrument used for this study was very simple. This was advantageous in that a very detailed analysis was possible within the resources of the project, but a disadvantage in not giving insights into more complex Smart instrumentation.

## 3 Study Conclusions

The study conclusions are numbered in (approximate) order of decreasing importance:

1. High reliability of Smart instrumentation can only be justified by means of an independent assessment with full access to the design documentation.

2. Instrument suppliers produce their products to a quality that they regard as appropriate, but this cannot take into account the nature of a specific application (such as a nuclear power plant).

3. There is currently no standard specifying the design documentation for Smart instrumentation nor even a list of issues that should be addressed. In consequence, instrument suppliers have no generic mechanism to satisfy those customers which might require independent evidence to support a claim for high reliability.

4. For the simpler Smart instrument, such as the one studied here, the validation methods researched in this contract, which are we believe appropriate to SIL3 of 61508, would cost about £8 per machine instruction. This is much cheaper than the more demanding safety standards or the techniques used in the aerospace sector which are usually quoted as costing around £500 per machine instruction.

5. The cost of providing meaningful evidence of no defects in Smart instrumentation increases rapidly with the complexity of the instrument. Hence the fact that users have no means of judging the complexity of the software within a Smart instrument is a major concern.

6. Based upon information obtained from this project, together with knowledge of similar applications in NPL, it would be possible to define an evaluation procedure for Smart instruments. Such a procedure could be used by purchasers with high reliability requirements and also by instrument suppliers intending to supply such markets.

More specific technical conclusions are contained in the summary of the study items in Appendix A.

## References


[1] E N Adams, Optimizing preventive service of software products, IBM Journal Res. and Dev., vol. 28, no. 1, pp.2-14, 1984.

[2] BS 7925:1998. Software testing. Part 1: Vocabulary, Part 2: Software component testing.

[3] IEC 880:86. Software for computers in the safety systems of nuclear power stations. 1986.

[4] IEC 61508: Draft. Functional safety: safety-related systems. Parts 1-7. Draft for public comment, June 1998.

[5] B Littlewood and L Strigini. Validation of Ultra-High Dependability for Software-based Systems. Comm ACM. Vol 36, No 11, pp69-80.

[6] The Application of Smart Sensors in Nuclear Plant Control Systems. BAeSEMA. 21st February 1995.

[7] Defence Standard 00-42 (Part 2). Reliability and Maintainability Assurance Guides: Part 2: Software. 1st September 1997.

[8] Guidelines for assessment of software in microcomputer controlled equipment for safety-related systems. NT TECHN Report 287. May 1995.

[9] Software Considerations in Airborne Systems and Equipment Certification. Issued in the USA by the Requirements and Technical Concepts for Aviation (document RTCA SC167/DO-178B) and in Europe by the European Organization for Civil Aviation Electronics (EUROCAE document ED-12B). December 1992.

[10] Guidelines For The Use Of The C Language In Vehicle Based Software. The Motor Industry Software Reliability Association. MIRA. April 1998. ISBN 0 9524156 9 0.

[11] B A Wichmann. Microprocessor design faults. Microprocessors and Microsystems, Vol 17, No 7, pp399-401. 1993.

[12] B A Wichmann. A Review of a Safety-Critical Software Standard. NPL. June 1994.

[13] B A Wichmann. Why it is difficult producing safety critical software? Ingenuity. (ICL's Technical Journal). May 1995 pp96-104.

[14] B A Wichmann. Software in Scientific Instruments — Measurement Good Practice Guide No 5. May 1997.

# A Study items and result summary

The study items are listed below with a short summary of the conclusions.

## A.1 Evaluation against NORDTEST method

The questions raised in the NORDTEST method [8] were answered and the effectiveness of their approach assessed.

**Conclusions:**

1. Since the supplier does not know the context of use of an instrument, it is not possible for him to show that the instrument's reliability is sufficient.

2. Assessment of an instrument requires access to the instrument's design documentation which is not usually available to the supplier's customers.

## A.2 System testing

Evidence of testing at the system level cannot provide assurance to the highest levels, since it is practically impossible to run the systems testing long enough. The practical limits will be quantified in this case. Evidence of authenticated use of instruments by users will be investigated to see if the reliability bounds can be improved by taking such evidence into account.

The application of Def Std 00-42 (Part 2) was considered.

The advantages and disadvantages of using the NPL stress testing method will also be explored.

**Conclusions:**

1. Due the simplicity of the Druck instrument, their approach of alpha and beta testing should be sufficient to ensure zero defects.

2. The NPL stress testing is not appropriate, again due to the simplicity of the instrument.

3. The concept in Def Std 00-42 (Part 2) of designing for a specific reliability target does not seem appropriate, since Druck's process should provide zero defects.

## A.3 Component testing

The application of the British Computer Society component testing standard was analysed. The 13 methods were investigated to determine the most appropriate ones for trial use.

**Conclusions:**

1. Component testing, as now defined in BS 7925 [2], should be applied to the code in the signal path, either as a single component, or as a number of components (when the path is too complex to be tested as one unit).

2. Statement testing and boundary value testing, both with 100% coverage, should be applied.

## A.4 Predictable execution

Establishing that a program does not contain errors which would result in unpredicatable execution is an important step in quality assurance. This is sometimes done by use of the static code analysis method called 'semantic analysis'. This investigated an alternative approach of applying the Model C code validation service offered by NPL. This is a dynamic method, but provides a similar level of assurance as semantic analysis.

**Conclusions:**

1. The main signal path can be written in a strictly portable subset of C.
2. It is possible to check the strict adherence to a subset by means of tools working on a host environment which allows for execution of appropriate test cases.
3. Code using the C programming language should adhere to the MISRA Guidelines [10].

## A.5 Integrity of the signal path

A relatively small proportion of instrumentation software is code handling the main measurement data. Hence the approach to validation is to ensure that there is no interference by the less critical code to the signal processing code (and also to check the accuracy of the signal processing). The use of interrupts can provide a source of undetected interference unless strict design rules are followed. Hence this activity was to ensure the design is sound in this respect for Druck's actual instrument and to illustrate the issues arising in the evaluation of an arbitrary instrument.

**Conclusions:**

1. It is possible to define simple design rules to ensure the integrity of the signal path based upon constructs of the C programming language.
2. For the simpler instruments, it is possible to check such design rules by means of a code review.

## A.6 Accuracy of the signal processing

The numerical accuracy and stability of the signal processing algorithm was analysed by NPL.

**Conclusions:**

1. For Druck's instrument, the numerical calculations performed are sufficiently simple that their analysis is relatively straightforward. For more complicated calculations, a detailed floating-point error analysis may be necessary.
2. The analysis needs to encompass not only the calculations undertaken by the instrument for the user, but also calculations undertaken by the supplier to establish parameters such as calibration constants.

## A.7 Validation of software floating point

The small microprocessors used in many of Druck's instruments (and we suspect most others) lack floating point hardware. Since it is very easy to have 'rare' undetected errors in software floating point, NPL investigated the application of the NPL/NAG floating point validation package. This package is capable of detecting bugs like that in the original Pentium processor.

**Conclusion:**

1. For Druck's instrument, the most complex component is the C compiler's floating point package. The simplicity of the computations undertaken does not make the application of the NPL/NAG package worthwhile.

## A.8 Logical soundness of overall design

If the overall design of the system is complex, then logical flaws can be present. A Swedish tool from Prover Technology AB, Prover, can be used to model a complex design and check for vital properties. If any such property is not true, the tool will provide a counter-example. Hence this approach can verify logical soundness without testing.

**Conclusions:**

1. The complexity of the logic in Druck's device does not warrant the use of Prover. For their instrument, the most complex component in terms of the logic is required for factory calibration and is not available to the customer.

2. Any Smart instrument with more than about 10 switches and/or controlled by a simple protocol would gain from the use of Prover. This would provide increased assurance to the customer and reduce the need for systems testing.

## A.9 Qualification of the microprocessor

It is unlikely that any microprocessor is flawless. The Pentium bug has illustrated that one cannot ignore a microprocessor bug undermining a Smart instrument. This activity attempted to quantify the issue which is problematic due to the lack of traceability of the silicon to the mask, and is based upon [11].

**Conclusions:**

1. The assurance given by NEC includes consideration of high integrity applications, such as nuclear applications. This implies that there is no obvious barrier to the application of NEC's microprocessors, as used by Druck.

2. NEC would probably be prepared the disclose known microcode design faults, if its customers requested such information. Hence the ISO 9000 quality control loop could be closed on this source of errors. This loop should cover upgrades to the microprocessor.

## A.10 Paper evaluation against DO-178B

The civil avionics standard for safety-related software has highly specific requirements for the development process. Hence a paper evaluation of the software development of a new Druck instrument was straightforward. This assesses what technical measures Druck is not currently undertaking, and what measures they undertake not identified in DO-178B. This activity concentrated on those aspects not covered by NORDTEST.

**Conclusions:**

1. The civil avionics standard at the highest level is inappropriate for the development of an instrument as simple as the one used for this study.

2. Using a lower level of the civil avionics standard would not necessarily provide the assurance needed to apply the instrument in a safety context.

3. In practice, it is necessary to use professional judgement to determine the verification methods to be applied, which needs to take into account the complexity of the software.

## A.11 Smart versus conventional technology

This activity assesses likely reliability aspects of the move towards distributed control/measurement systems. Specific points identified prior to this study included:

- It is often thought that the drive to distributed intelligence will increase reliability. An overview of such systems does not necessarily support that view. The move replaces one central data acquisition system and simple field sensors with sensors containing complete acquisition systems in a less benign environment. Hence potentially the reliability of the system may be reduced. This is further exacerbated if duplex or triplex systems are required.

- The reduction in cabling can have a significant effect on costs, but even this is overstated as practical issues tend to force the bus to have spurs and star points that increase the cable runs. In safety critical systems a number of redundant and separately wired buses will be required to reduce common mode failures. The problems become worse if bridges, repeaters or routers are needed to extend the bus to cover the same sort of area accessible by 4-20mA loops.

- A positive feature of distributed systems for safety is the potential for autonomous control in the event of a common lane failure. By suitable wiring of the bus and the use of embedded control software (eg, PID loops) in the sensor/actuator sub-systems, the plant control can remain even if the main supervisor computer systems fails. The loss of the watch-dog signal from the central system can then be used to automatically shut the sub-systems down in a controlled manner.

- Re-configurable software in a distributed system brings its own safety problems of checking and control, as well as robust operation in the non-benign environment.

- Digital systems are assumed to be more immune to interference that analogue ones due to the higher signal levels and the ability to incorporate check values. Analogue sensors, however, have the advantage of gradual degradation in the face of excessive interference. Digital systems exhibit sudden catastrophic failure.

- Engineering competence is another big hurdle to distributed systems. Classical 4-20mA systems are well known, easily tested and checked within current technician skill levels. The introduction of digital signaling requires a new raft of skills and understanding, and dangerous mistakes can be made through ignorance. Even the small step for HART has bought mistaken re-configurations to the extent that remote programming is prohibited in some industries.

- Assess the likely problems in obtaining a high integrity software system using Fieldbus. The problem here is that since Fieldbus contains all 7 levels of the OSI model, the volume and complexity of the software involved is large.

- The results from this activity will be compared with Druck's conventional application of Mil Std 217 and FMEA.

**Conclusions:**

1. Depending upon the configuration, a Fieldbus system could be half as reliable as conventional technology.

2. Qualification of the potentially complex software in a Fieldbus system (or an ASIC, if the complexity is mainly within such a chip), could present major technical problem and be a large cost overhead.