Data: An often-ignored component of safety-related systems

Alastair Faulkner, MSc.; CEng.; MBCS.; CSE International Ltd. Flixborough. UK

Neil Storey, Ph.D; CEng; FBCS; University of Warwick, Coventry. UK

Keywords: Data, data-driven, safety-related systems

Abstract

Safety-related systems are being constructed from hardware, software and data. The data often describes the real world environment in which the system will operate and plays a vital role in ensuring its correct operation. Logic as well as good engineering practice dictates that data is produced to the same integrity requirements as the other system elements. Unfortunately, experience and anecdotal evidence suggest that this is all too commonly not the case.

Data-driven systems use data from a number of sources including data extracted (and possibly processed) from existing external information systems and data produced specifically for the required system. This data is used to describe the system environment using configuration data (which is largely static or slowly moving) and status data (which is dynamic and will sometimes change rapidly). In addition, a minority of systems may use data to describe a changing use of the system with time. This additional data may be thought of as a schedule or timetable identifying control requirements as sequences or combinations of control actions.

Typically, information is supplied to these systems through a '*data supply chain*' that may involve transformations and adaptations by external information systems and human processes. The management of the data supply chain can introduce significant errors to the development and operation of safety-related systems. The work described in this paper sets out to provide much needed guidance on appropriate methods of dealing with data, which is a largely ignored system component.

Introduction

A key task in the production of any safety-related system is the identification of its safety requirements. These in turn dictate the safety integrity requirements of the system, which must cover all aspects of the system's implementation, including that of its data. Functional safety standards such as IEC 61508 (ref. 1) offer guidance on the treatment of the software and hardware within safety-related systems, but say very little about the data. Indeed IEC 61508-3 (ref. 2) suggests that data should be treated as software. However, consideration of the nature and characteristics of data shows that it is very different from executable software. It is therefore essential that data is appropriately and adequately treated during system development and maintenance if it is to contribute effectively to the safe operation of the system.

A (written) safety-justification is often required before a safety-related system is put into service. The purpose of the justification is to demonstrate, objectively, that the risks associated with the system are acceptable. In general, the approval of a safety justification may be a regulatory or contractual requirement (or both). The safety justification can be known by a variety of names, including safety case, safety argument and safety assessment (ref. 3). The safety justification may be either qualitative or quantitative but it is more usually a mixture of both. The justification lives with the system throughout the lifecycle and will be updated if the existing system is modified, enhanced or decommissioned.

Data-driven systems offer the opportunity to modify the data component and hence influence the behaviour of the system without changing the hardware and software components. Engineering sense and logic dictates that in order to maintain the integrity of the overall system, this data modification process should also have a safety-justification. A data-driven system should therefore have two safety-justifications: one for the generic system and one for the process of data modification.

The responsibility to ensure that the data integrity requirements have been met is identified by DO 200A (ref. 4) as resting with the user of the data. However, Tillotson (ref. 5) observes an alternative approach where the responsibility for data integrity may rest at the data source with data entry. These two positions are discussed in this paper. The questions of trust, compatibility and commonality of intent (use) are fundamental to the positions stated. Each position places different requirements on data production and these substantially influence the data lifecycle.

Data-driven systems

A widely used form of data-driven system is a management information system, which interfaces with invoice, purchase and inventory systems within a company. However, we are increasingly seeing data-driven systems at the heart of safety-related, real-time applications, where the integrity requirements may be very different.

In the past, design techniques for real-time systems have favoured functional structuring (ref. 6) and, more latterly, Object Orientated techniques (ref. 7). Software from existing designs may be reused either as Commercial-Off-The-Shelf (COTS) products or as libraries of code, which implement specific functionality. The term Software-Of-Uncertain-Pedigree (SOUP) has been used by Sinclair (ref. 8) to describe COTS software elements, however this term is misleading as the pedigree of these software elements may be impeccable. A more accurate term would be *"pre-existing"* software elements. The desire amongst suppliers and operators to use data-configured pre-existing (COTS) products in safety-related systems for reasons of cost and timescale (ref. 9) gives rise to a requirement for the safety management of data-driven safety-related systems. Within such systems, data enables the characterisation of standardised components for use in a particular situation.

However, these applications not only use static data to describe the environment in which the data driven-system will operate, but also, in many cases, make use of additional dynamic data to inform the system of the status of external entities. Work by the authors in this area (ref. 10 -14) has been primarily concerned with systems that make extensive use of large amounts of data. Typical examples of large-scale data-driven systems are transportation control systems that use data to describe the infrastructure and vehicles, as well as the intended use of the system in a schedule or timetable. These data types are described in table 1 below (ref. 11). A large amount of data forms an essential element within the system and plays a vital role in ensuring its correct operation (ref. 10).

Table 1 – Types of data used by data-driven systems

Data type	**Description**
Static	Configuration data as a description of the infrastructure and its capabilities and constraints. The infrastructure includes both fixed assets and mobile equipment such as vehicles. This description is most conveniently regarded as "*static*" data in that it represents the entities in the real world, which changes only in response to the action of maintenance or modification of these entities. Static configuration data is also used to configure the actual software components and may also provide the characterisation of these standardised components including parameterisation that describes how the software is to operate.
Status	Status Data is provided through interfaces to external reporting systems and direct status information from connections to local sensors and other inputs.
Operational	Individual operational conditions are commonly communicated to the control system via manual input. The operator receives these operational conditions through human communication interfaces such as telephone and fax. The set of operational conditions represents persistent restrictions on the use of the infrastructure through reports of flood, landslip or failed vehicle.
Schedule	A Command Schedule is used to describe the mesh of the required use of the infrastructure. For example a railway control system would use a train schedule to describe the planned movements of multiple trains across the rail infrastructure.

Data used by these systems will either be produced specifically for the individual application or may be re-used from an existing data source. As the scale of the application increases, the more likely is the re-use of data, thus offering economies from sharing data between several systems. Existing data may require modification or adaptation before use by the data-driven system. All manipulation, whether it is transmission or transformation (adaptation), has the potential to introduce errors. Data that has been transferred and manipulated by several systems will accumulate data errors.

Errors, faults and failures

This paper will use the convention that an error leads to a fault that subsequently causes the system to fail. Where a system comprises a number of subsystems, errors present in a subsystem may cause an error in another system that subsequently fails. The application of the definition (ref. 3) of both error and fault to the analysis of a real world system may lead the analyst to consider that both errors and faults lead to failure. This view is supported by the railway industry publication "Engineering Safety Management" which states that a failure is "*a deviation from the specified performance of a system, product or other change. A failure is the consequence of a fault or error*" (ref. 15). In complex systems, combinations and sequences of errors and faults may also be observed to lead to failure.

A risk-based approach to system development subjects the system elements to a classification of risk, based upon the probability of the identified hazards occurring and the severity or consequence of any accident that may result. A hazard is a condition of the system that has effects at the system boundary and that can lead to an accident. The system condition may be caused by a failure of the system or may exist as a result of incomplete design and development of the system.

Key to the identification of hazards is a definition of a system boundary, as it is important to define what elements are external to the system and what elements are internal. The hazard and risk analysis process is used to establish system integrity requirements. These requirements are then apportioned between components of the design, based upon the part played by each function, the cause and consequence of the failure of each function, and the implementation of each function by hardware, software and data components of the system. The integrity requirements apportioned to the data component of the system are termed in this paper '*data integrity requirements*'.

Data integrity requirements

The data used by a data-driven system may have extensive influence over both the normal and abnormal behaviour of the system. In this context abnormal behaviour is used to describe the behaviour of the system in the presence of failures. Data integrity requirements provide one means by which the suitability of data models may be assessed. Data integrity requirements should also influence the selection of techniques and measures used to attain and maintain these requirements. Furthermore, data integrity requirements should also be used to select suitable data representation techniques such as redundancy or diversity within the data model. Additional design analysis may identify sensitivity of the system to errors in either single data elements or collections of data. These data integrity requirements may also be used to identify verification and validation requirements for the system.

Integrity requirements are used to identify the lower limits of target failure rates that may be claimed for a safety-related system. Integrity requirements are apportioned between random and systematic failures. The hardware component will exhibit both random and systematic failures, whilst the software and data components will only exhibit systematic failures.

Systematic errors are also present in the data supply stream. This will subject the data to a number of transformations by a number of information systems. Each transformation or adaptation may introduce errors that go un-noticed by other consumers of the data. The final user of the data must ensure that the data is fit-for-purpose. In order to enforce data validation in both the on-line and off-line system an adequate system boundary model should be defined.

Data integrity validation at the data source

Tillotson (ref. 5) observes that where data entry activities are undertaken 3, 4 or 5 systems away from where the data is used, the responsibility for data integrity may rest at the data source with data entry. However as data entry is far removed from the system consuming the data, several assumptions may be identified. The first is that the use of data entry validation is sufficient to demonstrate that adequate data integrity has been attained (by all systems using this data). This assumption requires that the quality of the internal transmission, distribution and adaptation processes is such that these introduce a sufficiently low number of errors.

Where data integrity validation takes place at the data entry point then the data entry systems should maintain a '*register of interest*' of all the uses of the data entered. The purpose of this register of interest is to facilitate the management of changes to the data entry process through the use of techniques such as impact assessment. Many systems may grow from a single initial implementation, which has subsequently been extended through a series of enhancements. The observation made here, is that the original data model, its adaptations, extensions and use, *must* remain consistent with the original design intent (and the design assumptions must remain valid). This requirement is often compromised where there are many of pre-existing legacy systems, such as those information systems which are employed within the UK rail industry.

Additionally Tillotson infers a '*data supply chain*' by acknowledging that data is entered several systems away from its use. Where validation at a 'remote' data entry point is employed, errors in this source data introduce risk, which will only be significant in the system using the data.

Data integrity as the responsibility of the user of the data

In using a risk-based approach, it is common practice to identify a system boundary. Interfaces are defined for information which crosses the system boundary, and through these interfaces a number of design options for data validation may be identified. The most obvious position for data validation is at that point where the data crosses the system boundary. Although an obvious approach, data validation at the boundary may not always be possible. Gross errors of omission or range check failures are readily detected at the interface, but more subtle errors may be plausible, and may pass the gross error detection criteria. Additional development activities will be required to specify the information model to be used in conjunction with the data integrity requirements.

One of the few standards attempting to provide guidance in this area is DO 200A (ref. 4). Although limited to aeronautical navigation data, this standard provides a means of identifying the data sources and components in the supply of data to its users. The standard also addresses data integrity requirements.

DO 200A identifies a number of '*data quality*' criteria and the concept of the '*aeronautical data chains*'. Data quality consists of:

1. the *accuracy* of the data;
2. the *resolution* of the data;
3. the confidence that the data is not corrupted while stored or in transit (*assurance level*);
4. the ability to determine the origin of the data (*traceability*);
5. the level of confidence that the data is applicable to the period of (its) intended use (*timeliness*);
6. all of the data needed to support the function is provided (*completeness*); and
7. the *format* of the data meets the users requirements.

Although '*format*' identifies the '*data element*' and the '*relationship between elements*', DO 200A also cautions as to the adequacy of data relationships, suggesting that data sets are to be tested using simulation before use.

Aeronautical data chains are the conceptual representation of the path that a set, or element, of aeronautical data takes from its creation to its end use.

Data supply chain

Changes made to aeronautical data are made at regular intervals, in the case of adaptation data, which describes the sectors for an Air Traffic Control (ATC) system, these updates occur every 28 days.

DO 200A identifies a number of sources of aeronautical navigation data and an aeronautical data processing model. The data processing model identifies the following components:

1. origination; whereby values, names or other information are determined and assigned to required data elements for subsequent use;
2. transmission; a process where data is moved from one physical location to another;
3. data preparation; where a variety of aeronautical data elements are analysed, translated, complied and/or formatted to produce data configured for use;
4. data application integration; a process whereby data, in an application specific configuration and format, is made available to the target application; and
5. end-use of aeronautical data; a process for assessing and acting upon the output of an application.

These components are then arranged to create a description of the aeronautical data chain. This description may then be modified based upon feedback from the final end-use of the aeronautical data. Each of these components may be further decomposed within the DO 200A framework.

It is clear that DO 200A provides much useful information on the use of data within aeronautical navigation systems. However, while data is clearly a major element within systems in a range of other industrial sectors, the use of data is not addressed in the literature or standards associated with these areas, or in generic standards such as IEC 61508 (ref. 1). Also, while DO 200A provides guidance on the use of data within particular systems, this is restricted to the largely static data used in such equipment. Many modern systems use dynamic as well as static data and little guidance is available on the management of such elements.

An incident in the wilderness

Systems making use of dynamic re-assignment data are already in common use. Consider an aircraft transporting prisoners which crashes in an undisclosed wilderness. The hostile occupants survive and escape into the wilderness. The location of the crash is remote and although the geography of the location is known (including maps and terrain data), it is difficult terrain with no modern infrastructure such as communications or roads. The terrain affords the prisoners substantial cover. Small (4 man) search teams are dispatched equipped with limited range communications devices. The terrain contains mountains and valleys creating communications black spots. Each team communicates with one or more base stations, preferably two if range and terrain allows.

As the search progresses the progress of the search teams is plotted by a central incident control facility, a broad search front develops, and logistics issues begin to arise; communications devices require replacement power supplies; food and perhaps medical evacuation due to injury. One or more of the hostiles overcomes one or more of the search teams, and (false) progress is reported via the base station. If contact with a search team is lost and then re-established can this status data still be trusted?

Although this incident is used as an illustration, many systems contain substantial elements of dynamic behaviour that would have been traditionally implemented as static components. It is in this context that dynamic data descriptions are being more commonly used to implement this dynamic behaviour.

Data entry in this illustration is based upon progress (status) reporting from the search teams. In describing this incident an assumption has been made that all the described equipment is equally capable and compatible. In extending this scenario consideration should be given to search teams drawn from different agencies or even different nations. The incident may now be considered to comprise dissimilar (not necessarily incompatible) equipment and a potential for differences in the use of language (between agency and nation).

A train control system

The incident in the wilderness (above) illustrates a number of issues associated with dynamic data, particularly where dynamic information may be used to characterise system behaviour. Modern mobile phone technology allows railway signal engineers to consider replacing line side signals with in-cab signalling to regulate the passage of trains across a rail infrastructure. The removal of line-side equipment offers substantial potential for maintenance cost savings. However, a number of significant engineering problems need to be overcome.

Each train will be commanded to move a safe distance, known as the '*movement authority*'. The movement authority will be calculated based upon data about the train (braking capacity, traction force and train type) and the rail infrastructure ahead of the train (gradient, curvature, speed restrictions). The rail network is implemented across countryside and uses (both over and under) bridges, viaducts, tunnels and cuttings. Coverage from the base station cells is incomplete, manifest as black spots in the coverage of the transmission signal. The radio signal has to compete with the electromagnetic effects of the overhead electrification of electric train traction. Transient system conditions such as wet weather (causing higher than normal attenuation of the signal), or lightening strike increase the effective size of the black spots. The seasonal status of vegetation may also affect signal attenuation and hence the black spot may experience seasonal variance.

Part of the attraction of the Train Control System (TCS) is the ability to increase the capacity of the line by moving from fixed blocks of control to moving blocks associated with the capabilities of both the train and the rail infrastructure. Passenger safety is paramount to the continued economic future of the railway. A train crash where a train comes to a sudden uncontrolled halt blocking its running line and adjacent running lines poses a significant hazard (and potential loss of life). Dynamic data is then urgently required to inform the control system of the incident, in order to command trains to a halt. In a number of locations the cell base stations are installed adjacent to the railway. It is feasible for an incident to disable a base station. A missing base station coupled with an adjacent black spot and poor weather conditions may render the system inoperable even without an incident.

The association of a number of data items may be required to provide an assessment of the operational capability of one or more route sections. Design mitigations may require that in locations where significant technical difficulties cannot be permanently overcome, line side signalling must be retained. The retention of line side signalling equipment will reduce the potential for maintenance cost savings.

Data used by the TCS may be drawn from a number of data sources. Static data will describe the environment in which the system operates and include the capabilities and constraints of rolling stock and infrastructure. Dynamic data is also required to record and report operational conditions, which form a persistent reduction of the capability of one or more system elements. These operational conditions may represent a failure in one of the two train braking systems or high wind in association with a bridge or viaduct. Indeed static configuration data is also used to configure the actual software components and may also provide the characterisation, including parameterisation, that describes how the software is to operate. In addition dissimilar equipment may be supplied by different vendors giving rise to potential for differences in capabilities and constraints.

The overall required operation of the system might also be described as data. In the context of a rail network this data would be the schedule or timetable. Re-planning the service would require that the schedule is modified. This changed description of operation will inherently adjust the probability of an incident, which may lead to an accident. Consider a rail junction; trains leave and join the main line at the junction; the efficient use of the junction is termed junction optimisation and balances the traffic through the junction with infrastructure control actions such as setting of one or more routes. Clearly the risk associated with the operation of the junction is influenced by the proximity of trains and their approach speeds. The process of construction (and subsequent modification) of the train schedule demands high integrity to reduce risk to an acceptable level.

Discussion

This paper does not represent an exhaustive treatment of data in data-driven systems. However, it does suggest that this is an area that is worthy of further study.

Although safety-related systems may be subjected to Hazard Analysis, the integrity requirements of the system are rarely apportioned to the data component. Indeed data in these systems is often poorly structured making error detection difficult. A conservative analysis policy would consider that all data failures are a potential cause of hazards. All data, therefore, should be developed to the highest integrity of the system. This policy takes no account of the data model, the structure of the data model or the relative strengths and weaknesses of relationships between data elements. When looking through the tables of techniques and measures in a standard such as IEC 61508 (ref. 1) the common goal is a need to partition software (and hardware) components as a means to control the influence of failure on the overall system. The identification of modules and diverse elements all assert resilient behaviours, which contribute to the system integrity. Changes due to maintenance or system upgrade can be controlled by the demonstration of independence of the replacement or upgraded component.

As data-driven systems become larger, making more extensive use of data, the identification and management of data integrity becomes a significant factor in the demonstration of system integrity. The use (and reuse) of data shared within the commercial enterprise and modifications to the data structure(s), data retrieval mechanisms and data content, may each have a significant impact on the correctness of the data. In some cases, a large amount of data forms an essential element within the system and plays a vital role in ensuring its correct operation. Furthermore this data forms an essential part of the design and to ensure that the system as a whole has the correct behaviour and the appropriate safety integrity, the data must be developed and verified with as much care as the software and hardware.

Considerable financial resources are required to satisfy a conservative hazard and risk analysis policy that treats all data as the highest integrity of the system. An alternative policy would be to develop data integrity requirements that would allow the targeting of development resources by a classification of risk, based upon failures due to data errors or data faults.

A number of salient features of data-driven systems are listed below:

1. Data-driven systems are often complex in comparison with conventional computer-based systems.
2. Data-driven systems often form part of a hierarchy of computers that exchange real-time data.
3. This complexity, and the interchange of data, makes data-driven systems challenging to design. It may also make it difficult for those charged with configuring the systems to gain a full insight into the original design intent.
4. Current approaches to configuration and validation of specific applications depend on assumptions of modularity and independence of the data.
5. Modularity and independence require well-structured data that has well-defined interfaces to the application software. These are often missing from current data-driven systems.
6. These deficiencies can produce problems for system validation.
7. Data may be drawn from external information systems to be used either as static configuration data or to influence the dynamic behaviour of the system.
8. Data supplied to these systems is through a supply chain, which may introduce errors during transmission or adaptation.

The task of validating a particular instance of a generic system might be simplified by producing a generic safety case that covers the product, and a separate supporting document that relates to a particular application or instance. The latter could then be updated to represent the particular details of an individual installation. The generic part of the safety case would need to argue (and demonstrate) that changes to the data would not affect overall system safety. This might require that such changes could be shown to have a limited scope for affecting the operation of the system and that faults within the configuration data could be adequately detected by testing. It is likely that such an argument would require that the data was adequately structured and validated to ensure its integrity.

Data may change on a regular basis. For instance, we noted earlier that aeronautical data may be updated on a 28 day cycle. The process of data modification and upgrade should also be required to demonstrate that changes to the data do not affect overall system safety. Therefore this data process also requires a safety case.

Despite the varied approaches used, it is clear that data is treated very differently from application software. This would tend to suggest that it should receive separate treatment within standards and guidelines. It appears that data is often not subjected to the same degree of hazard and risk analysis as other system elements and this may result in data-related hazards being overlooked. It also appears that data is not normally assigned any specific integrity requirement. The use of a separate lifecycle for configuration data, with a specific data safety requirements specification, might overcome many of these problems.

Conclusions

Safety-related systems are being constructed from hardware, software and data, leading to an increasing number of systems that make use of large amounts of configuration data. Experience and anecdotal evidence suggests that this data is not always being developed in a manner that is consistent with the integrity requirements of the system in question. One reason for this could be that data is largely ignored within the various generic and industry-specific standards.

The ultimate responsibility for ensuring that data meets the quality required for its intended use rests with the end-user of the data. To a large extent, this responsibility can be satisfied by ensuring that data is supplied by an organisation that is appropriately accredited to provide data of the required integrity level (ref. 3).

Data is often supplied to these systems through a variety of means. DO 200A describes an aeronautical data supply chain to provide static navigation data. However, many systems also use dynamic data supplied through interfaces to external information systems. This dynamic data is also subject to errors based upon the transmission, storage and transformation of data. Indeed large systems may contain data models based upon similar data from several (possibly incompatible) sources. Data integrity requirements provide a means not only to assess the suitability of the data model, but also the suitability of the data supply chain.

During the creation, storage, manipulation and distribution of data some form of tool will be used. These tools should be defined including functionality, performance, operational environment and any constraints that are limitations of the tool rather than the operational system. Furthermore each tool should be assessed to ensure that it is capable of providing data of the requisite quality.

This paper suggests that within data-driven systems, data should be considered as a distinct system element with its own requirements documents and lifecycle. Generic standards such as IEC 61508 should also give specific guidance on the design, production and verification of data. This, it is hoped, will encourage engineers to give the production and management of data, the attention it deserves.

References

1. International Electrotechnical Commission; IEC 61508-1 Functional Safety of electrical / electronic / programmable electronic safety-related systems – Part 1:2000 General Requirements. Geneva, 2000

2. International Electrotechnical Commission; IEC 61508-3 Functional Safety of electrical / electronic / programmable electronic safety-related systems – Part 3:1998 Software Requirements. Geneva 1998.

3. International Electrotechnical Commission; IEC 61508-4 Functional Safety of electrical / electronic / programmable electronic safety-related systems – Part 4:1998 Definitions and abbreviations. Geneva 2000

4. RTCA; DO-200A Standards for processing aeronautical data. 1998

5. J. Tillotson; "*System Safety and Management Information System"s*; Aspects of Safety Management: Proceedings of the 9th Safety Critical Systems Symposium Bristol UK 2001. Ed F. Redmill and T. Anderson; pp 13-34 ISBN 1-85233-411-8

6. J. E. Cooling; "*Software design for real time systems*"; Chapman and Hall London 1991 ISBN 0-412-34180-8

7. R. Stevens, P. Brook, K. Jackson, S. Arnold; "*Systems Engineering – coping with complexity*"; Prentice hall; London 1998; ISBN 0-13-095085-8

8. I. J. Sinclair; "*Use of Commercial-Off-The-Shelf (COTS) Software in Safety-related applications*"; HSE books. CRR80 1995 ISBN 0-7176-0984-7

9. J. A. McDermid "*The cost of COTS*", IEE Colloquium - COTS and Safety critical systems London, January 1998.

10. N. Storey and A. Faulkner (2002) Data Management in Data-Driven Safety-Related Systems. Proceedings 20th Systems Safety Conference, Denver, CO., August 2002, 466-475. ISBN 0-9721385-0-1.

11. A. Faulkner (2002). "*Safer Data: The use of data in the context of a railway control system*", Proceedings of the 10th Safety-critical Systems Symposium, Southampton, UK 2002. pp 217-230 ISBN 1-85233-561-0.

12. N. Storey and A. Faulkner (2001) *The Role of Data in Safety-Related Systems*. Proc. 19th Systems Safety Conference, Huntsville, AL., September 2001, pp 26-35.

13. A. Faulkner and N. Storey (2001) The Role of Data in Safety-Related Railway Control Systems. Proceedings of the 19th Systems Safety Conference, Huntsville, AL., September 2001, pp 793-800.

14. A. Faulkner, P. Bennett, R. Pierce, I. A. H. Johnson, N. Storey (2000) *The Safety Management of Data Driven Safety Related Systems* Proc. 19th International Conference. Safecomp 2000, Rotterdam, The Netherlands, 24-27th October, pp 86-95. ISBN 3-540-41186-0.

15. RAILTRACK Engineering Safety Management (Yellow Book 3); Issue 3, Volumes 1 and 2 Fundamentals and guidance. Published by RAILTRACK on behalf of the UK Rail Industry; London 2000 ISBN 0-9537595-0-4

Biography

Alastair Faulkner, MSc., MBCS, C.Eng; CSE International Ltd., Glanford House, Bellwin Drive, Flixborough DN15 8SN, UK. Tel. +44 1724 862169, fax +44 1724 846256 email - agf@cse-euro.com

Alastair Faulkner holds an MSc degree in Computer Science from Salford University and is a Chartered Engineer. His background is in software development mainly concerned with computer based command and control systems. He now works on a large UK Rail infrastructure project. Alastair's research interests are in the data management of data-driven safety-related systems. He is also a Research Engineer with the University of Warwick and is studying for an Engineering Doctorate.

N. Storey, B.Sc., Ph.D., FBCS, MIEE, C.Eng. School of Engineering, University of Warwick, Coventry, CV4 7AL, UK. Tel. - +44 24 7652 3247, fax - +44 24 7641 8922, e-mail - N.Storey@warwick.ac.uk.

Neil Storey is a Director within the School of Engineering of the University of Warwick. His primary research interests are in the area of safety-critical computer systems. He is a member of the BCS Expert Panel on Safety-Critical Systems and has a large number of publications including both journal and conference papers. Neil is also the author of several textbooks on electronics and safety, including "Safety Critical Computer Systems" published by Addison-Wesley.