Strategies for the Management of Data-Intensive Safety-Related Systems

Alastair Faulkner, M.Sc.; CSE International Ltd.; Flixborough, UK

Neil Storey, Ph.D.; School of Engineering, University of Warwick; Coventry, UK

Keywords: data, data-intensive, data-driven, safety-related systems, software, safety

Abstract

It is common for large infrastructure projects to contain many computer-based systems, which require the flexibility offered by a data-driven implementation. Primary amongst the drivers for this flexibility is a requirement to reduce the acquisition cost of such systems whilst retaining the adaptability to implement the multiple stage works required by large-scale infrastructure changes. These projects become not only data-driven, but also data intensive. A large and significant component of each project is data. In these projects the safety of the system is likely to depend upon the correctness of this data.

Given finite economic resources, all data cannot be treated equally. Other more realistic strategies are required. One such strategy would be to develop data integrity requirements that allow the targeting of development resources by a classification of risk, based upon failures due to data errors or data faults. This pragmatic approach develops the position advocated by standards such as IEC 61508 for hardware and software.

However these large infrastructure projects often contain multiple overlapping stage-works, continually changing, upgrading and modifying the infrastructure as the project progresses to its conclusion. Data will also continually change, and hence influence not only the data integrity but also the data integrity requirements. Common sense and engineering judgement dictate that change is controlled, managed in such a way that the influence and impact of change is understood. In this respect, data integrity demands the same control.

This paper examines a number of options to exercise control over data integrity. One approach employs a common data dictionary in conjunction with the use of an integrated suite of computer-based tools to plan, produce, manage and implement the data required. At the other extreme is the use of paper-based requirements documents supported by a common approach to data. These options each have their strengths, and weaknesses.

Introduction

Data-intensive systems are increasingly used to implement safety-related systems. These systems range from single applications to a hierarchy of computer-based systems that may share data either through common interfaces or through some shared description of the real world. This data will form a substantial component of the system and influence, if not determine, the behaviour of the system. Therefore, data errors will significantly influence the system behaviour. With this increasing dependence upon the use of data, emerge requirements for the management of data. These requirements are primarily concerned with control over the propagation of data errors, particularly where the system requires upgrade or modification.

Data provision is strongly influenced by the integrity of the source data and the processes required to transport it to the systems which will consume it. This is a multi-facetted problem; on one hand are small-scale systems whose data can be adequately managed through data entry and

delivery of a validated dataset. At the other extreme are large-scale systems drawing data from a number of sources. This data is processed, transformed, consolidated, transported and finally delivered to one element of the overall system. An example of such a system is Air Traffic Management (ATM), where a number of control systems share common data such as aircraft type or 'adaptation data' describing the airways. A second facet is the maturity of the application. New systems may require completely new datasets, whilst new implementations of existing systems may re-use existing data.

This paper will set out the options for a generalised system. This system is neither novel nor an implementation of a replacement system. The paper will assume that a suitable hazard and risk assessment has identified the components of the required data that have safety responsibilities, and therefore that the data component has a set of data integrity requirements.

The first part of this paper discusses the driving forces for re-use, highlighting the trend towards the creation of data-driven systems. Data provision requires the definition of a strategy, drawing on the parallels between software development and the provision of data. In our generalised system an assessment model is described to provide the necessary assurance that the data has attained and maintained the required integrity. Assessment can only be effective if the data is adequately defined. The design criteria used to provide data to a number of systems with differing integrity requirements are discussed. And finally data provision is discussed. In our generalised system we now have the major components for options to exercise the necessary control over data integrity.

Requirements of a data strategy

The requirements of a data strategy parallel the requirements for the development of software. In each case a set of requirements are analysed, and a preliminary design is produced and assessed. This preliminary design is then refined until it is judged fit for purpose. This preliminary design is then developed into the detailed design, assessed, and hence moves on to implementation. It is at this point that the parallels between data and software falter. The processes for the creation of software are well known and practiced. Many tools support the development of software and are available to analyse its structure. The recommended lists of techniques and measures contained within standards such as IEC 61508 (ref. 1) direct the software developer towards best practice. A similar list of techniques and measures is not available for data.

The behaviour of data-driven software is directed to a greater or lesser extent by the data it consumes. The static analysis of the software structure may mislead the assessor, as the same levels of assurance cannot commonly be attained for data as for software. This is in part based upon the absence of an agreed set of techniques and measures, but also that there is a wide spectrum of practice ranging from the excellent to the very poor.

The major issue is that data is commonly provided after the system development has been completed. The widespread use (and re-use) of general-purpose applications separates the application developer from the software developer. It is in this separation that the design intent of the data component may be misinterpreted or even lost.

The term used in this paper to describe the processes associated with the provision of data is a 'data supply chain'. The supply of data within the Air Traffic Control sector is described by the RTCA standard D0 200A. This standard describes 'aeronautical data chain' that contain components which it terms 'phases'. In order to preserve the distinction between system components and elements of the supply chain this paper will also use the term 'phase'. Using the

data supply chain, data is transported from its origin, transformed and adapted before finally being presented to the consuming system. To assist in the management of data integrity the authors advocate treating data as a separate component (ref. 2, 3). Data would therefore attract an allocation of the system integrity requirements and should be treated as a peer component alongside the hardware and software components of the computer-based system.

The strategy for the data component should recognise that data may be created by a number of organisations, groups, or political bodies. This data supply chain may also cross the boundaries of a number of organisations, groups, or political bodies. The data strategy should contain policies to manage the propagation of data errors and contain error detection schemes to increase the likelihood of error detection.

In separating the system development activities from data provision, it is common for the acquisition cost of the system to reflect the initial installation of the system, without taking into account the costs associated with the maintenance of data integrity through the life of the system. In the course of its working life, a data-driven system may be used by one or more organisations, which change and evolve over time. Organisational responsibilities and boundaries also change. The data supply chain must respect these organisational changes and make provision to identify changes of responsibility, ownership and liability for the data.

An assessment model for data

The assessment model described below is based upon the concepts and terminology used in the UK CAA document Air Traffic Services Safety Requirements (CAP 670) section SW01 (ref. 4). The relevant software concepts from CAP 670 SW01 have been adapted for data to give:

1. the concept of Assurance Evidence Level (AEL) to express the level of confidence that a data component will possess the integrity according to its specification on the basis of the strength and depth of the available evidence;

2. the five fundamental safety objectives which a safety-related system must fulfil, namely *requirements validity*, *requirements satisfaction*, *traceability*, *configuration consistency* and *non-interference* with safety functions by non-safety functions;

3. the concept of Direct and Backing evidence; and

4. the requirement to define the integrity of a data component in terms of a defined set of attributes.

The assessment model is developed for each of the phases of the data supply-chain. The Assurance Evidence Level as defined in SW01 is not a measure of reliability, but a measure of confidence that a component satisfies its requirements. A safety-related system will have specific safety reliability requirements, which may be expressed in the form of numeric failure rate targets, or as a Safety Integrity Level (SIL).

Data definition

This assessment model requires that a number of requirements specifications are required not only for the data but also for each element of the data chain, and the tools that they contain. These requirement specifications form the data definition. Data definition also requires documented guidelines. These guidelines are concerned with the exchange of data between sub-

systems and applications that comprise the system. A majority of these guidelines are derived from practices common to many safety related standards. Data should only be shared amongst these systems when the data integrity requirements of each consuming system are fully satisfied. These requirements are:

1. that the data integrity requirements of all sub-systems or applications within the system are documented;

2. data may be passed from a higher integrity system to a lower integrity system (provided that the data from the higher integrity system exceeds the data integrity requirements of the lower integrity system for each of the data elements passed across the interface, including failure rates and failure modes);

3. data may be passed between systems of the same integrity requirements if and only if these data integrity requirements are compatible, including failure rates and failure modes;

4. data may not be passed from a lower integrity system to a higher integrity system unless data integrity requirements are compatible, including failure rates and failure modes, for each of the data elements passed across the interface; and

5. that the hardware and software components of these systems meet the integrity requirements for each system.

These guidelines may then be used establish a framework for a data dictionary. It is common when developing large information systems to create such a data dictionary, particularly where the proposed system uses one or more data models. The data dictionary usually contains a description of the data model, its structures and the data elements and their attributes. Data-intensive systems require that an extended data dictionary include:

1. the origin of each data element, relationship and attribute (this may also provide a description of the data supply chain requirements);

2. the owner of each data element, relationship and attribute (Ownership may itself be a complex issue as data may originate from a number of organisational and political bodies and include any consolidation to produce a higher data abstraction. For example, operational considerations may allow ownership to change based upon pre-defined criteria);

3. a *register of interest* of all those systems which will use (consume) each data element, relationship and attribute together with the integrity requirement of each consuming system;

4. the system responsible for the updating of each data element, relationship and attribute;

5. a set of rules for the validation of each data element, relationship and attribute; and

6. a set of default values to be used in the event of failure to acquire data of the appropriate integrity for each data element, relationship or attribute for each system which will consume the data.

The general requirements of a data dictionary must also be maintained. The data dictionary should be complete and unambiguous. A key objective of the data dictionary is to capture information only once in such a way that it is available for all. The use of the data dictionary

provides a single point in which the data requirements for all the systems may be managed. The data dictionary also contains an explicit statement of interrelations, and data dependencies within the overall system.

Provision of data

Within this paper data provision is used to describe the techniques and measures to create and transport the data to the consuming systems. Data provision has two distinct components; firstly the design, creation and management of the data supply chain; secondly data origination.

Design and construction of the data supply chain

The design and construction of a data chain will be based upon the phases of the D0 200A aeronautical data chain. In order to simplify and focus the discussion upon design and construction the following assumptions are made:

1. Data integrity (including D0 200A data quality) requirements are documented for each system that uses data from the supply chain;

2. Data requirements specification, identifying each data element, its structure, content and any references to other data (in other data elements and data records), for the data supply chain are documented for each system using the data;

3. Data verification and validation requirements are documented for each system using data from the supply chain. These may include rules, and default data sets; and

4. Organisational and political boundaries are documented, including responsibilities, ownership, intellectual property rights, legal constraints, liabilities and restrictions on use.

The purpose of the data supply chain is to transport data from its source, apply a number of processes (phases) and deliver data of suitable integrity as required by all the systems using the data. Data supply chains may become long and complex containing many interrelated phases.

The design process for the data supply chain should:

1. Identify data origins
   This will be the start of the data supply chain. As a general rule the first phase will be a receive phase. Where data has a known integrity this data may be stored for use by subsequent phases. If additional confidence in the integrity of the data is required then the this additional confidence may be attained through test, analysis and where necessary simulation.

2. Identify boundaries (organisational, legal, political):
   Changes of responsibility will require that the data supply chain recognises the boundary and creates a suitable distribution media. As a general rule the select, format and distribute phases are used to create this media. The exchange need not be a physical media but may be an automated exchange. The important requirement is to recognise the change of responsibility, and possibly ownership. On the other side of the boundary will be a receive phase.

3. Identify process and adaptation phases
   The major interface issues are established. The next step is to plan the required processing and adaptation of the datasets. This may be achieved with the assemble, translate, select and transform phases.

4. Apportion the integrity requirements
   The integrity requirements for the data are specified. These integrity requirements are then apportioned between the data source and the phases of the data supply chain.

5. Identify evidence requirements
   Having apportioned the integrity requirements between the data source and the phases of the data supply chain, it is then possible to establish the evidence requirements for the data source and the phases of the data supply chain to satisfy the assurance model. This will identify the verification criteria required by each phase.

6. Specify corrective action process
   Failure to satisfy verification criteria of each phase will create error reports, which require corrective action. The final step is to identify the corrective action process for each phase; group of phases and the data supply chain as a whole.

7. Assess the design of the data supply chain; repeat steps 1 to 7 as necessary to attain required goals (integrity organisational responsibilities, liabilities, ownerships).

Data origination

Data origination is perhaps the most difficult issue concerning the design, development, data provision and maintenance of data-driven systems. The production of data of an adequate integrity (sufficient for meeting the requirements of all systems using this data) may take a number of forms. Small-scale systems may use data entry to create a validated dataset, possibly with limited tools support. These toolsets may include graphical entry and also incorporate verification rules. The traditional view of data entry is based upon data entry into forms either realised as paper records or automated on computer screens. In such systems a form-based data entry is arranged so that data is a collection of related items to support the operator and his (or her) perception of reasonability whilst performing the data entry task.

As the scale of these systems becomes larger, the volume and nature of the data required changes. Where systems co-operate with supervisory or subordinate systems, data within this hierarchy may be described in terms of its vertical use within the hierarchy. Data production may require vertical datasets, which describe the infrastructure and the abstractions that control the use of the infrastructure. Where systems co-operate as peers these datasets may be considered as horizontal sections across the system hierarchy. Horizontal datasets may extend the span of control through the extension of existing datasets to describe a greater geographical area.

The widespread availability of general purpose computing has produced a lot of data. Many companies have vast data stores, which are locked into proprietary systems. These systems restrict access through constraints based either in hardware or software. In many cases obsolescence has rendered access to this data difficult. A recent trend has been to provide access to this data through third party products. These products act as a brokers or agents translating the request into a form that the original system can satisfy. These products are commonly known as middleware.

Having gained access to this data, data quality tools may then be employed to provide assessments as to the consistency of the data, using a range of metrics. Where data quality metrics demonstrate the poor quality of data, this data should not be considered for use by high integrity systems.

Data may also be derived from specialist tools. In considering the ATM system and terrain data it uses, terrestrial survey data may be compared with satellite images to provide diverse sources of data. The altitude data may be compared with satellite radar images to confirm the altitude of geological structures (such as mountains).

A paper-based approach to data.

Paper-based documents are a medium supported by general-purpose tools ranging from simple text files to office integration packages. Simple documents that do not contain active fields are simple to produce, although they may be labour intensive, and may be very flexible. Stable environments with long change cycles may adequately be managed using such simple documentation. Paper-based systems are becoming less common as organisations seek competitive advantage by becoming more agile and able to react to market changes.

Any proposed changes to a safety-related system should respect the safety functions that they contain. Many regulatory authorities require written submissions to demonstrate that risk has been adequately controlled. One of the mayor issues with a paper-based system is that as the number and size of documents increases, changes become more difficult to control and implement safely. This problem is exacerbated by traceability requirements, which are established and maintained manually.

Paper based systems are suited to small-scale implementations, which cannot justify the cost of tools to support the data provision process. Many of these applications are supported by configuration tools supplied by the vendor and are either stand-alone applications or are part of a range of applications or vendor specific products. These applications may consume existing datasets and are probability based on updates to existing systems. Even changes to existing systems require a safety justification to demonstrate the risk is adequately managed and controlled. Any change may require that data provision be documented through computer-based tools to demonstrate additional traceability, control and to provide references to evidence of test and analysis.

The accommodation of the flexibility required by a data-driven implementation may be difficult to implement using a paper-based system. Its implementation will be labour intensive and may prove to be neither flexible nor cost effective. The logistics and management overhead of providing the required adaptability to implement the multiple stage works may not be feasible using a paper-based system.

Common data dictionary and the use of an integrated suite of computer-based tools

The advantage of using computer-based tools to support data design is consistency and the management of change. Large-scale systems are likely to consist of many computer-based systems from a variety of vendors. Each system will require data of the appropriate integrity. The issues of change are compounded in large infrastructure change projects, which are delivered in a number of phases each containing a number of stages.

Where the infrastructure is modified for upgrade the constraints and capabilities of the system may change significantly. It is common for the capability of the system to fall whilst a component is replaced. The capability is then increased to reflect the capabilities of the new component. The management of these changes requires strong configuration management to enforce baselines for each change in capability. Change on this scale is best managed with the automation provided by computer-based tools.

These tools are complex and usually expensive. Their use should be justified through their ability to reduce both for project risk and safety risk. However misuse of these tools may increase both for project risk and safety risk. A significant factor in the use of these tools is the operational procedures and work instructions required to formalise their use and to realise their potential. Mere possession of a tool cannot be used as any form of justification.

Highly integrated toolsets are common in the manufacture of automobiles, which combine computer aided design (CAD) with computer aided manufacture (CAM). In such systems the design of the car is created within the same computer environment as design of the production processes and manufacturing tooling. Changes to the design are planned for introduction after a specific point in the production cycle. It is common in such integrated systems to use a series of catalogues to describe collections of items. One such catalogue is the bill of materials used to describe the kit of parts, at each level of the design. In the same way in which the kit of parts describes the system as a hierarchy based upon assembly and sub-assembly finally terminating in the description of a component. Data catalogues may also be used to describe the structure and relationships of the data.

Discussion

The scale and potential for change of the application has a significant influence upon the strategies for data provision. Infrastructure change projects are probably the most visible examples of large-scale systems. An example of such a system is the UK railway project for the West Coast Route Modernisation. Ambitious changes were planned to the track, electrification and control systems together with the introduction of new trains. The proposed degree of change, the interrelated nature of train, track and control system presents significant systems engineering challenges.

Such projects require that an adequate data definition be created. This data definition identifies the data required by each system together with its integrity requirements. Where data is passed between sub-systems specific rules are required to control the propagation of data errors across the system. The level of control required is dependant upon the integrity requirements of the data. The higher the integrity requirement the greater the control required.

This paper has outlined one method of the design and creation of a data supply chain. It is important to recognise that the data supply chain may cross a number of organisational and political boundaries involving several agencies. The data supply chain should recognise these boundaries to ensure that a clear demarcation between responsibilities can be established and maintained.

All data is not equal. Data integrity requirements should be used to target resources towards the integrity requirements of the system and the data required. One means of gaining confidence that the data integrity has been attained and maintained is through the use of an evidence-based assurance model. The model proposed in this paper is adapted from CAP 670.

The choice of using an integrated computer-based toolset or to use document-based controls is dependant upon a number of factors. The scale of the application is a significant factor as large-scale systems are better managed with the flexibility offered by computer-based tools. Small-scale projects may not be able to justify the expense of computer-based tools nor the significant investment in process and procedure to make effective use of them.

Conclusions

The choice of a data provision strategy is multi-facetted, based upon a diverse range of factors including complexity, scale, maturity of application and cost. Significant investment in resources is required to effectively manage data-driven systems. This investment is not only based upon the initial provision of the system but must take account of the cost of maintaining data integrity for the life of the system. Data provision requires extensive resources to ensure that the data integrity requirements are attainted and maintained. These integrity requirements extend beyond issues of data quality into the evidence that is required to support the integrity claim. This evidence is based upon the verification of the data through testing or analysis. The re-use of data may be constrained by the integrity that may be claimed for it. Low integrity data should not be used by a high integrity system.

The case for treating data as a separate system component has been made. This case recognises that data has different properties from both hardware and software. Although some parallels are apparent with the development of software components, these similarities are limited, particularly for data *provision*. Data provision has two main aspects; firstly the design and construction of the data supply chain, and secondly data origination.

One approach to the design and construction of a data supply chain has been presented in this paper. However data origination remains a difficult issue. The acquisition of high integrity data may require several diverse sources that are compared to enhance the probability of error detection. The key issue is to control the propagation of data errors. The data supply chain may span many organisations, agencies and even countries. This raises issues associated with responsibility, ownership, and liability for data errors.

Data provision demands a systematic, ordered design, based upon the requirements of the system and the ability of the organisation to support data provision. It may be possible to design robust data supply chains, but if the organisation does not or cannot support the provision of data of suitable integrity, the integrity of the operational system will be compromised. In extreme cases this may lead to system failures, harm or significant loss.

The strategy for system design should be influenced by the ability of the organisation to provide data of the required integrity. Whilst many large-scale systems require the flexibility offered by a data-driven implementation, little consideration is given to the through life cost of data. The pressures to reduce the acquisition cost of such systems whilst retaining the adaptability to implement the multiple stage works required by large-scale infrastructure changes draws the systems designers to consider data-driven designs. However this flexibility and adaptability require extensive support, as the influence of any individual data error on the behaviour of the system may be extensive.

As these systems become larger the use of paper-based systems to control and manage data integrity becomes both labour intensive and hence expensive. Large infrastructure change projects require the flexibility and control that may only be available through the support of computer-based tools to facilitate the required traceability and ensure consistency.

References

1. IEC 61508 Functional Safety of electrical / electronic / programmable electronic safety-related systems Geneva: International Electrotechnical Commission, 1998.

2. Storey N., Faulkner A. "*The Role of Data in Safety-Related Systems*", Proc. 19th International System Safety Conference, Huntsville 2001.

3. Storey N., Faulkner A. "*Data Management in Safety-Related Systems*", Proc. 20th International System Safety Conference, Denver 2002.

4. UK Civil Aviation Authority, Safety Regulation Group, Air Traffic Services Safety Requirement, Document CAP 670 section SW01 "Regulatory Objective for Software in Safety Related Air Traffic Services", Issue 6. London : October 2002.

5. RTCA: DO 200A Standards for Processing Aeronautical Data, Washington: Radio Technical Commission for Aeronautics, 1998.

Biographies

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. 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.

Neil 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 British Computer Society 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.