Developing Data Management Processes for Safety Critical Systems

Ken Frazer, Duncan Dowling, Mike Ainsworth
Praxis Critical Systems

Keywords: data management

Abstract

Many safety critical systems use data to configure their functionality for a particular application. The approach to data preparation has traditionally been seen as an adjunct to the software development environment, and many industries have developed perceivably robust methods and techniques to ensure that the data meets necessary quality criteria.

However, the increasing use of standardized components and subsystems has led to ever more dependence on configuration data to define a system's functionality. The development of safety-related systems by integrating and adapting existing systems, rather than bespoke development, means that not only must the hardware and software engineering approach change, but also the strategies adopted for the management of safety related data.

This paper describes work being undertaken as part of the development of the European Rail Traffic Management System (ERTMS), a computer-based control and protection system for trains which is being introduced across Europe, to ensure that the data required by ERTMS is managed appropriately. This is of particular concern because ERTMS poses new challenges to managing the data, partly as a result of the integrity, accuracy and volume required, but also due to the growing dependence by systems for their correct and safe operation on data. This paper discusses the problems encountered and the development of a data management framework to help alleviate these problems.

Overview of ERTMS

ERTMS is a new computer based control and protection system that has the capability to improve both safety and performance. It is required, under European Law, to be introduced on all high-speed train lines. The system also introduces a degree of interoperability between equipment from different suppliers.

ERTMS comprises a number of discrete sub-systems all essentially configured by data to achieve interoperability in a safety critical environment. Figure 1 presents a high level view of the key ERTMS components and interfaces to the supporting interlocking and control center systems. The driver's interface, the man machine interface (MMI), displays a Movement Authority (MA) to the driver, consisting of a safe speed and a safe authorised distance that the train can travel. This MA is enforced by the on-board control system which applies the brakes if the MA is violated.

Data is a significant component of the ERTMS system. A wide range of data is used in calculating and enforcing the MA and is obtained from:

- the driver – who supplies journey specific data which may include the weight and length of the train;
- the train – which is configured with data specific to the locomotive such as wheel size which is used to calculate distance travelled;

- the trackside – which is configured with route specific data, such as speed limits and track layouts. This is communicated to the train either from static trackside equipment (balise) or by radio, using GSM-R, from a Radio Block Center (RBC).

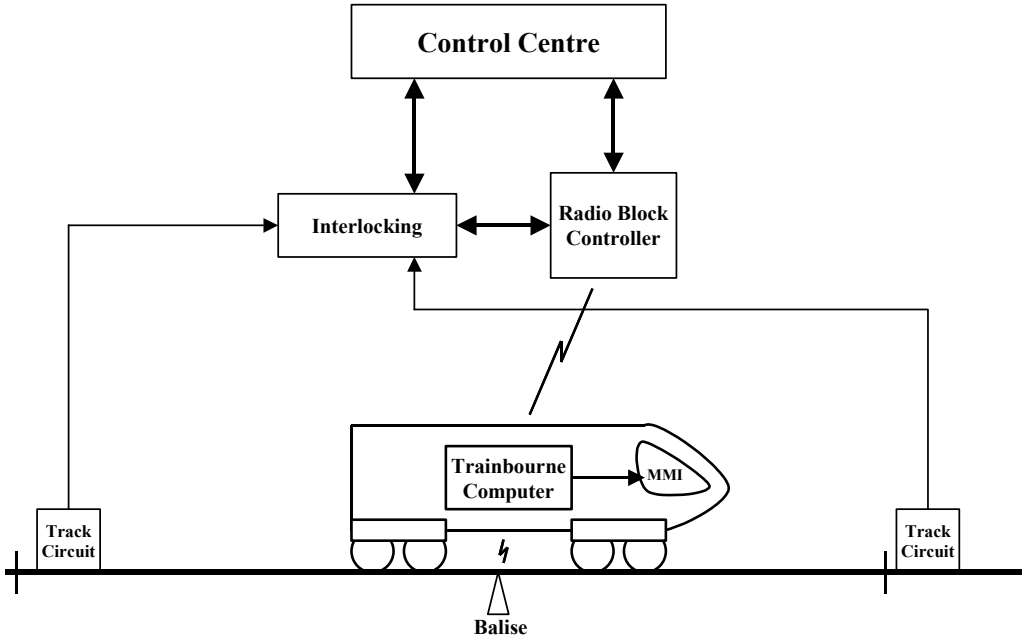

Figure 1 – Overview of the ERTMS system components

The ERTMS system is much more dependent on accurate data than previous generations of railway signalling systems. Managing ERTMS data is particularly complex because the data comes from a wide variety of sources. Figure 2 provides an indication of the number of stakeholder groups that own and maintain configuration data for ERTMS.

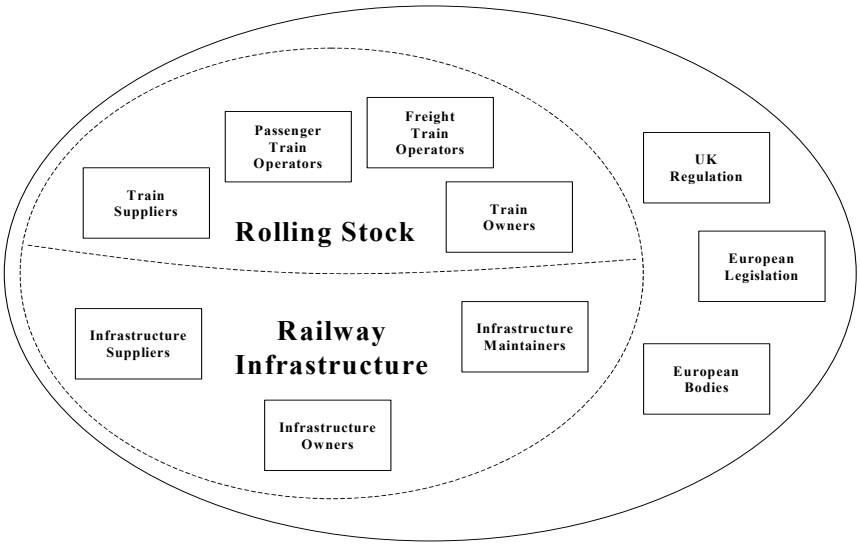

Figure 2 - Sources of ERTMS Configuration Data

Data Management Lifecycle

As with any complex problem or process it helps to break it into smaller more manageable components. This approach has been (successfully?) applied to system development, software development and can equally well be applied to data management.

The data lifecycle phases presented below, Figure 3, cover the complete lifecycle for a typical system application. The initial two phases focus on initial application design and delivery. The next three phases relate to the operation of application where minor operational changes will be necessary, except in cases when an enhancement, e.g. technology refresh or infrastructure change, requires data changes. The final, often overlooked phase, of replacement/decommissioning calls on consideration of data management support.

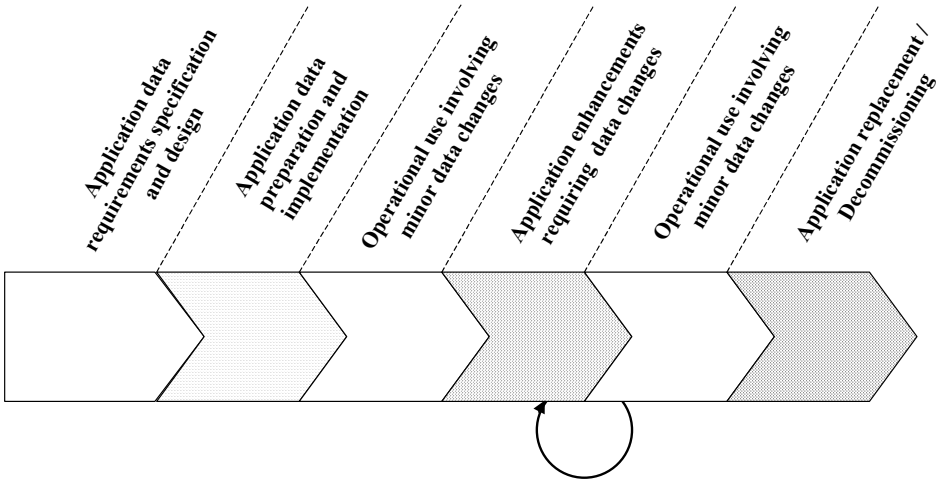

Figure 3 – Data Lifecycle Phases

The following sections look at the different phases of the lifecycle in more detail.

Data Requirements Specification and Design

In this phase the detailed data requirements specification, together with necessary data structure design is undertaken. The design includes the processes for the initial application data preparation and implementation. At the design stage it is also important to define the data processes associated with:

- data changes during operation
- enhancements that require data changes
- replacement /decommissioning

In ERTMS, two areas are particularly complex:

- Data categories
- Identifying the required integrity for data

Data Categories

The data used in a system often covers a number of different categories. The categorization needs to consider a number of different factors:

- the time and resources available to produce the data;
- the processes for integrating the data into the system;
- the sources of data;
- the way the data is used by the system;
- the change management requirements;
- the longevity of the data.

In ERTMS, possible categories are:

- rolling stock – to define the train's characteristics;
- scheme – to define the route's characteristics;
- fixed – to include data that changes infrequently (i.e. years)
- operational – to include data modified by users as part of system operation;
- logged – to include data recorded for later analysis to support business decision making and incident investigation.

In order to design an appropriate data management process, the integrity required for each category must be understood. In this paper we focus on safety integrity, but in practice other factors such as security and maintainability must also be considered.

Identifying the required integrity for data

The integrity required is determined by the hazards and risks that arise if the data is incorrect. A simplistic approach would be to decide that allowing the train to enter an unsafe state is a catastrophic situation and so train protection is a SIL4 function. Anything, which can therefore cause this system to fail, must be engineered to SIL4, and this includes virtually all the data.

However, if we look at the system and its usage in more detail, we begin to see that things are rather more complex, and that there are different types of risk associated with the use of data in different situations.

Category of data	Example Consequences (worst case)	Other factors required to cause an accident	Preparation time	Data lifetime
Fixed scheme data e.g. track gradient, distance between points etc	Collision between two trains or derailment.	None (driver/signaller cannot avoid an accident)	Offline, typically prepared over several weeks	Years
Operational trackside data e.g. temporary speed restrictions	Collision between two trains or derailment, as a result of application of incorrect speed limit.	Driver is briefed about speed restrictions and has lineside signs for reminders. An accident therefore requires a driver error as well as an ERTMS fault.	Has to be prepared rapidly (within minutes of a situation being reported).	Days - Months
Fixed rolling stock data e.g. braking and acceleration profile	Collision or derailment, as a result of delayed ERTMS braking.	Miscalculated position could result in an unavoidable accident.	Offline, established in the train development life cycle	Years
Operational train data e.g. details related to the load being hauled	Collision or derailment, as a result of delayed ERTMS braking.	Requires a driver error	Has to be prepared and entered rapidly to avoid train delays	Hours

We have two basic types of risk from ERTMS data:

1. A data error can result in an incorrect command to the ERTMS system (and hence the driver). In this situation the driver cannot avoid making a mistake, and the ERTMS system will not stop the train (since it is working to the same information). Although there are still some procedural mitigations in place, such as avoiding scheduling conflicting train movements, this situation represents a hazard where ERTMS can directly cause a catastrophic accident.
2. Data errors which reduce the effectiveness of the ERTMS protection functions, but where the driver has other information on which to base decisions. In these situations, the data error cannot directly cause an accident, it can only fail to prevent an accident triggered by driver error.

To assign SILs we have to consider the tolerability of the risk and the level of risk reduction required. Part of this assessment has also to consider what is reasonably practicable – often much of the data will follow the same process, and so it makes sense for all of the data to be treated as the same SIL, even if some of it represents a lower risk. Alternatively, if what is reasonably practicable doesn't make the risk tolerable then the process or system needs to be redesigned!

For the first type of risk (ERTMS misleads the driver and causes an accident) we have to start by considering a situation without ERTMS (and without conventional signals). The risk in this situation is clearly unacceptable since there is no way to prevent trains exceeding their safe limits of movement. This is also a type of hazard (computers killing people) which is less tolerable to society as a whole. The tolerable hazard rate in this case is likely to be around 10^{-9}/system/hour which implies SIL4.

Since a data fault can directly cause this hazard, the data will inherit the same integrity requirement of SIL4. SIL4 is probably achievable for this data because the data capture and preparation process can be spread over a long period of time, and the data can be thoroughly validated.

For the second situation (ERTMS fails to mitigate a driver error) the risk before we add the new protection functions is already at a tolerable level, but we want to reduce it still further. To set an appropriate SIL, we need to estimate the likely driver error rate – estimates will vary but 10^{-3}/hour is probably a pessimistic estimate. The overall tolerable hazard rate for a train exceeding its movement authority will be 10^{-9}/system/hour as in the previous case, so if the driver error rate is 10^{-3}/hour, the ERTMS failure rate for this situation needs to be 10^{-6}/system/hour. This is equivalent to SIL2. It is probably impractical to try to increase the integrity further given the operational constraints involved.

This second situation is actually even more complex, since with operational data there is also a risk involved in not being able to change the system quickly. The final risk assessment of the data management process must take this into account.

Application Data Preparation and Implementation

Once the initial requirements are understood, we must then define a data supply process which can gather the data from its various sources, prepare it appropriately for the application, and deliver the data into the operational system. A typical Data Supply Chain (DSC) is shown in the diagram below, Figure 4.

A prescriptive process is not described, because any process needs to be developed while also considering the attributes of the organisation and people involved. For example, some organisations may prefer to achieve high integrity through the use of appropriate tools and only nominal human involvement, whereas others may prefer to rely on a small group of experienced engineers.

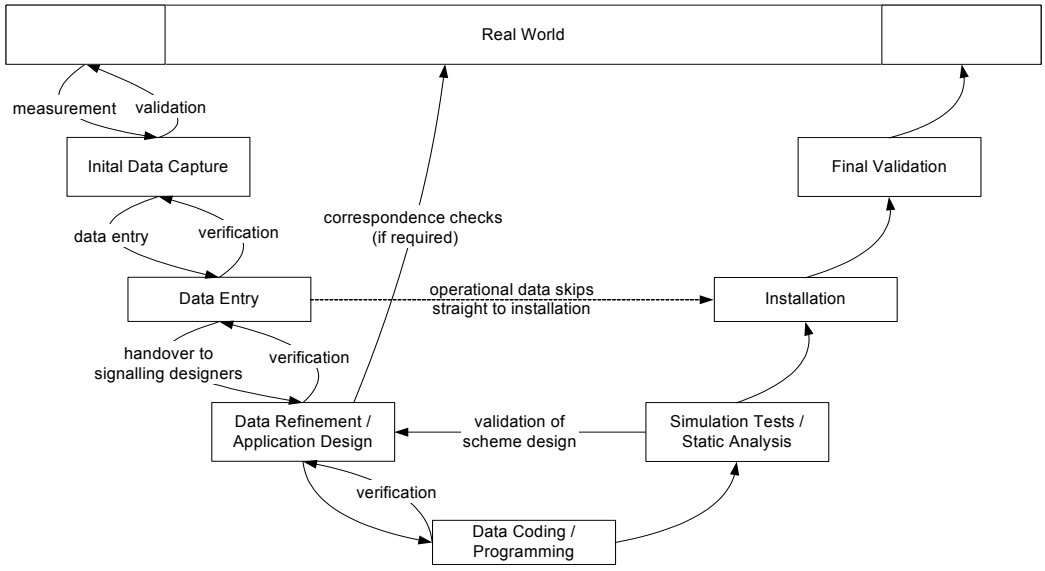

Figure 4 – Data Supply Chain (DSC)

Two issues that often cause problems are in determining:

- a suitable Data Supply Chain;
- the integrity needed from software tools.

Determining a suitable data supply chain

It is generally perceived that the task that introduces most uncertainty to the Data Supply Chain (DSC) is the Initial Data Capture stage. There are two main reasons for this:

1. The task is labour intensive because the records are generally paper based. Faults can be found through (independent) peer review. This may include checking the data entered into the DSC against the source records, checking entered data against the data's requirements, and the application of rules e.g. continuity of route gradient data.
2. The available records do not always accurately describe the current state of the infrastructure. Comparing different sources for a data item can increase the likelihood of identifying faults. This provides extra confidence if the same value is obtained from both source but is still susceptible to a common problem if neither source was updated following an infrastructure change. Additional measures require validation of available data against the actual infrastructure, which can be done in a number of ways depending on the required data.

The extent of verification and validation within the DSC needs to be considered. Some general rules and useful guiding statements are:

- Fixing faults early in the lifecycle is most cost effective;
- Not all faults are detectable early in the lifecycle, therefore additional testing is required later in the DSC;
- Use of good configuration management and quality management is essential to prevent degradation of the data through the DSC;
- Preventing faults entering the DSC is cheaper than removing them later, therefore development in tools to facilitate the entry of data is likely to be justified;
- Formal specification of data structures reduces the chances of faults within the specification and when interpreting the specification. The development and use of diverse tools developed from a common specification are less likely to suffer from a common mode failure;
- Verification only checks that the data is as it was when it was originated;
- Validation checks that the data correctly represents its intent e.g. the infrastructure;
- Informing people involved in managing the data about its criticality so that they are motivated to apply due care and attention;
- Diversity of tools should be strongly considered for high integrity data;
- Apply techniques, such as MD5, CRC and other hashing algorithms to enable detection of corrupted data;
- Safety and operational risk can both be reduced by increasing the confidence in the correctness of the data supplied by the DSC.

Organisations may decide to develop a single DSC for data items that require different integrities. The decision to do this depends on the degree of automation in the process and also the ratio of high integrity data items to low integrity data items.

Determining the integrity needed from software tools

Data in modern projects is almost always stored in databases and processed using a variety of software tools. The design of the data management process must consider the level of integrity that is required from any such tools.

Most safety standards (IEC61508, EN50128) give little guidance on this topic – possibly the most useful is the guidance in DO-178B on qualification of software tools. This states that:

- Tools require validation when the output is not fully verified by downstream lifecycle activities;
- Tools are generally used to aid or replace software development processes; as such they should be validated to provide at least the same level of confidence as the process replaced;
- Only tools whose operation, or input to output mapping, is deterministic can be qualified. Tools that do not exhibit this property must have the output validated by a separate verification process.

Data preparation should be treated in a similar way to software development – the integrity comes from using a combination of tools and manual reviews in a well-defined process. Each individual step is of fairly low integrity, but in combination the likelihood of an error surviving undetected into the final product is very low. The type of activities and guidance listed in standards such as IEC61508 for different safety integrity levels can often be interpreted for data preparation (e.g. can you write a formal specification of the data language? Are there static analysis techniques that could be applied to check the structure or contents of the data?). Process safety analysis (described in Def Stan 00-55) can be used to check that a process provides suitable defences against the possible errors that can be introduced at each stage.

Operational Use Involving Minor Data Changes

After commissioning the system will be put into operation. A number of events will necessitate changes to the data.

- Temporary or emergency conditions often require changes to speed restrictions on the track, which with ERTMS will need data to be changed.
- Maintenance activities may require equipment to be replaced, in which case the data resident in that system will need to be reprogrammed.
- Changes to the infrastructure layout may necessitate data changes to the trackside systems.
- Faults in data may be detected during use and require correction.

Each of these will require procedures to be defined to ensure that the data is changed in a controlled manner. In the last two scenarios the process followed will be a modified form of the initial data process. The first two scenarios require processes that manage and control change in a safe prescriptive manner. It is important that any data process to be used by operational staff is

designed to fit in with their existing methods of working to reduce the potential for human error due to misunderstandings.

Application Enhancements Requiring Data Changes

ERTMS systems will have a typical life of some 20 years or more. During this period a number of system enhancements may be necessary due to technology refreshes or changes to the infrastructure layout of the route area controlled by ERTMS.

At the initial system design stage provision should be made to define the processes required to satisfy these types of substantial change. The approach is likely to entail much of the data supply chain processes used for the initial production of the configuration data. Note that the early stages of data capture will not be necessary.

Application Replacement/Decommissioning

This phase of the lifecycle is critical for these railway control systems because the replacement or decommissioning has to be accomplished in stages. During this changeover period it is likely that several different versions of the data will be necessary to reflect the stage alterations. It may also be possible to transfer data into the replacement systems. This phase also needs to be considered in the design of the data management process, although in practice it is unlikely to add any extra requirements over those already introduced by the various maintenance activities.

Conclusions

Developing data management processes for safety related systems is a non-trivial task, and one that is often overlooked in the early stages of system development. In this paper we have tried to highlight a number of common problems that have been experienced with ERTMS:

- Allocation of safety integrity levels (SIL) to data should to be based on detailed risk assessment and not just determined by applying a blanket SIL to an entire system. For operational data, the risk assessment has to consider what is reasonably practicable given the time constraints involved and the risks that arise if data cannot be changed quickly.
- Development of a data supply process should concentrate on identifying and fixing faults as early in the process as possible. It is far easier to correct a measurement when it is being taken than to discover an error some time later, e.g. during system installation.
- Data and systems will change over time, and the design of the data management process should make use of strict configuration management throughout the whole data lifecycle, to take this into account. Thereby providing confidence in the data and reducing the need to repeat costly data collection and validation when updating railway system components.
- The initial stages of identifying the data requirements and safety targets determine the design of the data supply chain.
- The whole data life cycle has to be considered, and appropriate processes and measures developed so that the required integrity is maintained to ensure safe operation in normal and degraded conditions.
- Data preparation should be treated in a similar way to software development, where the integrity is obtained from using a combination of tools and manual reviews in a well-defined process.

Acknowledgements

This paper is based on discussions and experience over a number of different projects in the UK rail sector. The contents do not reflect the views or design details of any particular manufacturer or industry body. We would like to thank the numerous individuals at Network Rail, Railway Safety and Standards Board, Alstom, Bombardier, Siemens, First Great Western and our colleagues at Praxis Critical Systems with whom we have had many useful discussions.

References

1. Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC61508, IEC 1998.
2. Railway applications — Communications, signalling and processing systems — Software for railway control and protection systems, EN50128:2001, CENELEC, 2001.
3. Software Considerations in Airborne Systems and Equipment Certification, RTCA/DO178-B, December 1992.
4. Requirements for Safety Related Software in Defence Equipment, Defence Standard 00-55, UK Ministry of Defence, August 1997.

Biographies

Mike Ainsworth, Ken Frazer, Duncan Dowling
Praxis Critical Systems Limited, 20 Manvers Street, Bath BA1 1PX, UK.
Tel: +44 (1225) 466991, Fax: +44 (1225) 469006
Email: Michael.Ainsworth, Ken.Frazer, Duncan.Dowling@praxis-cs.co.uk

Dr. Mike Ainsworth is a senior safety engineer with Praxis Critical Systems. His experience covers 10 years of working on a variety of safety-related systems in both aerospace and rail. Recently he has been working with the Network Rail/Alstom Joint Project Team developing the Train Control System for the West Coast Main Line in the UK.

Ken Frazer is a senior consultant with Praxis Critical Systems, specializing in project management and process improvement. His recent projects have included work with Network Rail and Railway Safety Limited (now Railways Safety and Standards Board) to establish requirements for new data management processes for signalling systems.

Duncan Dowling is a systems engineer with Praxis Critical Systems. He has over 10 years engineering experience in rail, aerospace and manufacturing. He was the technical lead for a research project on ERTMS data management for Railway Safety Limited, which identified requirements for the components of a data management framework.