Cross-industry remote condition monitoring programme: Phase 2

Overview report

RESEARCH AND DEVELOPMENT

Copyright

Published: July 2015

Cross-industry remote condition monitoring programme: Phase 2 Overview report

Executive summary

The T1010 programme

RSSB research project T1010 'Cross-industry remote condition monitoring programme: Phase 2' is intended to put in place an enabling framework for cross-industry sharing of remote condition monitoring (RCM) data. This is part of Phase 2 of the Cross-industry RCM programme managed by the Cross-Industry Remote Condition Monitoring Strategy Group, a subgroup of the Vehicle/Vehicle System Interface Committee.

T1010 is in four parts:

- T1010-01, the current project, is to define a data architecture to support cross-industry RCM data sharing.
- T1010-02 is to define a commercial and legal framework to be used by parties wishing to share RCM data.
- T1010-03 will update industry standards and guidance to promote and simplify the use of the data sharing architecture.
- T1010-04 will update the business case assessment tool previously prepared under project T857 'Detailed review of selected remote condition monitoring areas' for RCM data sharing initiatives.

This project

Work done

The current project, T1010-01, has involved several tasks leading to the definition of a set of requirements for a data architecture for cross-industry sharing of RCM data. These are:

- A review of relevant work, covering RCM developments in the rail industry in the UK and elsewhere, previous research on cross-industry RCM, relevant initiatives in other industries such as oil and gas and aeronautics, and developments and trends in information technology and data management.
- Consultation with creators and users of RCM data in the UK rail industry.

- Collation and analysis of the findings of this work and the production of a set of principles to be obeyed by the Data Architecture.
- Development of the architectural principles into a set of requirements which define the form and contend of the Data Architecture and how it should be set up and managed.

Documents produced

The results of the work are contained in these products:

- Review of Relevant RCM Developments [1]. This document contains the review of previous work, work in other fields and recent IT developments, and the results of the industry consultation. Appendices give details of the data collected and a tabulated set of the resulting business goals, dependencies, recommendations, constraints and use cases which inform the definition of the Data Architecture.
- Architecture Requirements [2]. This document contains the architecture principles and the definition of the key requirements of the architecture and of its management and governance.
- Final Report (the current document). This is a narrative summary of the project, its work and the Data Architecture, with conclusions and recommendations. It draws on the work in the other two documents.
- Data Repository. This is a formal record of the architecture requirements, tracing each back to the original observation, goal, constraint or dependency which gives rise to it. As well as the textual requirements, the record includes draft data models and data interchange formats. The data repository is in the form of an Enterprise Architect™ database.

Audience

For a general audience, this Final Report gives a narrative overview of the work done, the principles of the data architecture and the recommendations for how to take the work forward. It provides cross-references into the other documents for the underlying detail.

The Review of Relevant RCM Developments document is also broadly narrative in form, though there is technical detail in some of the sections on IT developments.

The Architecture Requirements document is quite technical in nature, though the introductory sections including architecture principles (Sections 1 to 5) are the most accessible to the general reader.

The data architecture

Rationale and purpose

Although large amounts of data are gathered about railway assets by a wide portfolio of projects, the full potential of these data to help improve whole-system reliability and performance is not yet being realised. One of the reasons for this is the lack of standard ways to represent and interchange RCM data.

The realisation of the value of the collected data is greatest where the location and owner of the data collector are the same as of the data user. Conversely, the maturity of approach is lower where the rail/wheel boundary is crossed: where the data are gathered on vehicles about the infrastructure or vice versa. In these circumstances the costs of data gathering are met by one industry party whilst the benefits are realised by another.

An industry-wide framework with central support can help to address this asymmetry and the impact it has on participants' incentives to co-operate. An agreed data architecture is an important element of that framework.

Influences and principles

The rail industry's current technical and commercial state and its future strategic goals define a set of influences on the shape of the data architecture. Current and emerging developments in IT and project management also suggest ways of defining and implementing the architecture.

In terms of the rail industry, the influencing forces are:

- A regulatory framework which mandates the storage and transmission of particular types of data about infrastructure, rolling stock, safety and network capability.
- Strategic commitment to significant improvements in network capability, availability and reliability.
- Strategic requirement to reduce costs.
- The fragmented nature of the industry which means that top-down solutions cannot easily be mandated.
- The existence of many legacy IT systems and data formats which will be expensive to adapt.
- Reluctance of IT project managers to accept extra cost for conforming to standards not directly required by their projects.

- Variations in the legal ownership of data and the intellectual property in methods of processing it.
- The presence of proprietary, rather than open, approaches to RCM data storage and processing.
- Rapid growth in the number of sensors in use, the types of data they capture and the volume of data so generated.
- The need to automate higher-level asset management processes to include the forecasting of future asset performance and the planning and optimisation of interventions to minimise life-cycle cost and the level of disruption to passengers and freight customers.

Themes and changes in the IT industry have an impact on the type of data likely to be generated and interchanged; and the methods available for managing them:

- The emergence of 'big data' as an idea, with the tools to deal with it.
- "The development of open web-based standards for communication and data interchange.
- The rise in the use of 'service-oriented' approaches to connecting together disparate IT systems to support shared business processes.
- The challenge of maintaining security with the growing interconnection of systems and the use of off-the-shelf equipment.
- The growth in the use of intelligent systems based around ontological descriptions of data.

These influences shape the type of data architecture and management process required to address the cross-industry RCM requirement. The architecture will have the following key characteristics:

- It will be based on existing and emerging industry data models and formats to minimise the amount of work to set it up.
- It will offer different levels of compliance to suit different types of RCM project.
- It will support progressive increase in data integration as business needs develop, without extensive re-working.
- It will be service-based and message oriented, to minimise the extent of coupling between different IT systems and thus cost and effort of connecting them together and keeping them in step with each other.

- It will recognise master data sources and support the use of them as lookups to help drive up data consistency and quality across the industry.
- It will use published open standards for data interchange, to enable data to be liberated from proprietary formats.
- It will use standard definitions of the processing and data transfer aspects of RCM-driven asset management, to enable existing suppliers' intellectual property be protected whilst opening the field up to innovative new suppliers.
- It will use standard and well-understood techniques for achieving cyber-security.

Scope and use

The data architecture supports all types of RCM data. Standard representations and interchange methods are defined for the common types of data: sensor readings, waveforms, images, video and sound, statistical summaries, alerts and alarms, health assessments, prognoses and confidence levels, advisories. It can be extended to support other novel types of RCM data.

The architecture supports data about all types of rail asset: fixed, point, linear or areal; moving. Standard methods of identifying assets are supported.

RCM data gain value when they are integrated with other data about the assets and rail operations affected. The architecture defines standard ways of referencing operational trains and locations on the rail network.

The architecture defines a standard layered approach to separating out the processing and data elements of RCM driven business processes. This covers the whole spectrum of processing sophistication from the gathering of raw data from sensors, through the processing of the data and generation of statuses, alerts and alarms, to sophisticated optimisation of responses based on a range of data inputs.

The data architecture requirements are intended to be used:

- To guide project managers and technical staff on data formats and interchange techniques to use in their projects so that they are compliant with the architecture and thus able to take advantage of its capabilities in future.
- To inform industry data suppliers (such as Network Rail) on how to set up web services to make industry reference data lookups available to data interchangers.

- To define elements of shared IT infrastructure which would need to be established and managed by a central authority.
- To identify the management processes this authority would need to carry out.

Elements of the data architecture

The data architecture requirements cover these areas:

- Representation of basic data items such as dates, times and locations.
- A shared data model for RCM-driven data, based on existing industry data models and international RCM data interchanges.
- Ways of representing railway locations and infrastructure assets.
- Ways of representing rolling stock and its elements: locomotives, multiple units, vehicles, components.
- Timetabled trains: train formations, timetables, resourcing.
- Functions and levels of RCM-driven information: raw sensor data, processed sensor data, alerts and alarms, condition assessments, estimates of future condition and capability, interventions.
- Influencing factors: weather, passenger loads, freight volumes.
- Lookup services for reference data on assets and operations.
- Ways of interchanging data using open standards.
- Ways of securing and verifying data being interchanged.
- Ways to supplement standard data interchanges with application specific additions if needed.

Implementing the data architecture

Progressive approach

The Data Architecture is intended to be implemented gradually, in a series of stages which correspond to increasing levels of data and process integration. Each stage will generate business benefit on its own, though the benefits are greater for the later phases. Each stage is in some ways a pre-requisite for the ones that follow, though the sequence is not a strict one.

The stages are:

1 Standard representations for common data items.

2 Standard structure and representations for RCM data types.

3 Web-based data lookup services for railway assets using a data bus approach.

4 Standardised data interchange packets which share the same data bus.

5 Standard mechanisms for requesting and supplying data packets across the data bus.

6 Shared asset relationship ontology.

Supporting investment needed

At each stage, some investment is required to enable the architecture to support RCM data sharing. The investment is in the form of work to set down and manage the data standards, to provide shared IT infrastructure, and to connect existing and proposed reference data systems to the infrastructure. The main investments are:

- Stages 1 and 2: communication and management of the data standards; collaborative work with early adopters to ensure standards are fit for purpose.
- Stage 3: creation of facades on existing IT systems to support web-based lookup queries in a standard manner; creation of necessary enabling technologies to support this type of lookup (chiefly Automatic Vehicle Identification and Rail Position Location services).
- Stage 4: introduction of a data bus.
- Stage 5: enhancement of the data bus to support message-based interactions.
- Stage 6: development and management of a shared railway ontology and referencing mechanism.

Integration of existing systems and processes

The data architecture enables the communication of RCM data, where the source or destination of the data or supporting reference data comes from existing industry IT systems. Connecting these together economically without threatening their current function or their security is a prime concern.

The architecture envisages this being done by the use of a now standard set of IT techniques collectively known as 'enterprise application integration' (EAI). These have been developed to handle the problem posed by cross-industry RCM data transfer - system wide business processes involving the participation of existing IT systems that were not originally designed for this type of integration. The main characteristics of the EAI approach are:

- A 'bus' style of interconnection, where each connected system only has a single interface to the 'bus' but this enables it to communicate with all other connected systems.
- Communication via standard web service protocols. These are well understood and comparatively cheap and simple to implement.
- The use of messages. This approach decouples systems from each other, using the bus as a message transfer medium, so limits the impact on the individual systems and enables new ones to be added without disruption.

Conclusions and recommendations

The conclusions comprise practical next steps and engagement with the other parties who will need to cooperate to bring the data architecture into being. They are:

- Validate the principles of the proposed data architecture by consultation.
- Gather key information from contributing IT systems as they develop - chiefly ORBIS, LINX-TM, R2 and AVI.
- Set up a body to oversee the Data Architecture specification and management.
- Establish the 'top-down' elements of the Data Architecture through a Rail Industry Standard (RIS) or changes to Railway Group Standards (RGS). These are the overall principles of the architecture and the duties of those that participate in it.
- Start the main process of development of the architecture, the bottom-up process, by applying it to two pilot projects. Candidate projects are:
  - Data sharing between two different trackside bearing condition detection systems
  - Publishing of data from an existing train-mounted data gathering system using web services and a data bus
  - Start the development of the rail ontology with a view to applying it to these pilot projects.

Glossary of terms

Term	Definition
Advisory Generation	The sixth stage in the six-level processing model for condition monitoring defined in ISO13374. Work orders, recommendations etc. are issued based on the prognostic assessment.
AIXM	Aeronautical Information Exchange Model - framework to enable management and distribution of Aeronautical Information Services data.
Algorithm	A method of calculating a result based on input data.
Architecture	From TOGAF®:
a)	A formal description of a system, or a detailed plan of the system at component level, to guide its implementation (source: ISO/IEC 42010:2007).
b)	The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.
ATOC	Association of Train Operating Companies
AVI	Automatic Vehicle Identification
CCS	Control and Command Systems
Conceptual Model	A data model showing how the core concepts of an area of interest are defined and are related to each other, independently of any particular way of storing or exchanging the data.
Data Acquisition	The first stage in the six level processing model for condition monitoring defined in ISO13374. Data are captured from sensors.
Data Manipulation	The second stage in the six level processing model for condition monitoring defined in ISO13374.

	Data from a sensor undergoes basic summarising cleansing operations such as averaging or signal processing.
Data Model	A way of expressing the data items associated with an area of interest and the relationships between them.
Deterioration Profile	A graph describing the expected rate of deterioration of some characteristic of an asset from an assumed level of usage or passage of time.
DRACAS	Defect Resolution and Corrective Actions System
EIA	Enterprise Integration Architectures - an approach to integrating IT systems to support cross organisational business processes.
Encapsulation	The property of a system whereby only the elements of any object or data item that need to be made accessible to the outside world are visible; all other elements are kept private. Systems exhibiting this characteristic are easier to extend and develop.
Enterprise Architect	Data modelling tool used in T1010-01 to model the various artefacts of the project (such as sources, recommendations, dependencies, use cases) and traceability between them.
ESB	Enterprise Service Bus - a component of Enterprise Integration Architectures whereby all participating IT systems interact via a single 'bus' with standard data formats, rather than directly with each other.
ETL	Extract, Transformation, Load. The component of a data warehousing system responsible for capturing, cleaning, organising and storing the data to be reported or analysed.
EVN	European Vehicle Number. 12-digit unique identifying number for a rail vehicle used in the EU.
Extensibility	The property of a system which enables its capability to be enhanced easily after initial construction by adding new functions without disrupting existing ones or their users.
HABD	Hot Axle Box Detection System

Health Assessment
The fourth stage in the six level processing model for condition monitoring defined in ISO13374.

The health of the asset or component asset is assessed using the state from the previous stage and any supplementary information.

INSPIRE
Infrastructure for Spatial Information in the European Community - European directive aimed at creation of data infrastructure for spatial data.

InteGRail
European research project into information integration and management across the rail industry

Intelligent Infrastructure
Network Rail strategy to deliver infrastructure management improvements through intelligence design and maintenance through prediction and prevention rather than fixing.

ITPS
Integrated Train Planning System. Network Rail IT system for managing and distributing train timetables.

JSON
JavaScript Object Notation. A simple text-based way of representing structured information.

LINX TM
Network Rail initiative to enable open interchange of information about the control of moving trains using EIA components to handle interfaces between traffic management and other industry systems. See EIA.

Logger
A device which records, stores and passes on the data from one or more sensors.

Metadata
'Data about data'. Information about the format and type of data; possibly also including information about how it can be used.

MIMOSA
Operations and Maintenance Information Open System Alliance - association for development of open information standards for operations and maintenance in manufacturing, fleet, and facility. Enables collaborative asset lifecycle management.

Modularity
The property of a system in which the functions and data are organised in groupings (modules) by purpose, with a small number of clear relationships between the modules. This

	property makes the system more extensible and maintainable because it limits the spread of impact of any single change.
NETEX	Network Exchange - XML schema based standardisation initiative for data formats and interchange methods for railway data, used widely in Europe.
NR	Network Rail
OLE	Overhead line equipment
Ontology	A schema where objects are defined in relation to other objects to allow understanding of a new concept by a machine.
Ontology Engineering	A set of processes associated with constructing and maintaining an ontology.
ORBIS	Offering Rail Better Information Services - a Network Rail programme to improve its approach to acquisition, storage and usage of asset information
OSI	Open Systems Interconnect. A layered architecture for organising the networking together of computers.
OWL	Web Ontology Language - a language used to create ontologies. OWL is recommended by the internet's governing body World Wide Web Consortium (W3C).
Prognostic Assessment	The fifth stage in the six level processing model for condition monitoring defined in ISO13374. An estimate of the future health of the asset is made using the health assessment and any supplementary information about environment, expected usage etc.
R2	An IT initiative intended to replace the Rolling Stock Library and the Rail Vehicle Record System (RAVERS)
railML	Railway Markup Language - data standard for the interchange of railway industry, in the form of XML schemas. See XML.
RDF	Resource Description Framework. [1] A method of exchanging data attributes and relationships suitable for exploitation by ontology-based software
RCM	Remote condition monitoring -the activity of monitoring the condition of assets remotely: via a sensor at or on the asset which sends data to another location.

REST	Representational State Transfer. A style of web application architecture in which the history and state of a user's interaction is shown in the web pages themselves rather than being remembered by the server.
RFA	Railway Functional Architecture. A top-level view of the whole UK railway created by RSSB research project T912.
RFC	Request for Comment. An internet standard maintained by the Internet Engineering Task Force.
RFID	Radio frequency identification - a system whereby tags wirelessly transmit identification data to a sensor.
RIS	Rail Industry Standard - a definition of technical or functional requirements. In this document it is the Traction & Rolling Stock RIS which is relevant.
RSL	Rolling Stock Library. Computer system that currently holds the master reference data about UK rail vehicles.
Schema	A formal definition of a set of data structures.
	A specific type of schema that will be used is an XML Schema [1] which enables the structure and valid content of an XML file to be defined formally in a way that can be checked automatically in software.
Semantic	Pertaining to the meaning of data: what it represents and how that affects its relationship with other data.
Semantic metadata	Metadata which describes a way of defining the meaning of data. An ontology is a type of semantic metadata.
Sensor	A device which converts a physical property of the environment or of an asset into an electrical signal.
Server	A computer which hosts a software service which can be used by other connected computers.
SOAP	Originally 'Simple Object Access Protocol'. A standard for distributed processing over the internet. [1]
SPARQL	SPARQL Query Language for RDF [4]. A language for querying data sources that provide RDF results.
SQL	Structured Query Lanquage. Standard language for querying relational databases.

State Detection The third stage in the six level processing model for condition monitoring defined in ISO13374.

Processed data from sensors is assessed in relation to a threshold or band of acceptability and alerts or alarms raised if exceeded.

TLS - OB and MT Train Location Strategy - RSSB project to enable provision of train location information to application providers to an industry standards through collation, consolidation and distribution of information from trains and signalling.

TLS OB refers to on-board systems which send location information to the central system.

TLS MT (multi train). MT will enhance the accuracy of this information based on data from many trains and send it back to the train OB system.

TOGAF The Open Group Architecture Framework (http://www.opengroup.org/togaf/) A standard methodology for defining software architectures.

TOPS Total Operations System

TRUST Train Running System on TOPS

TSI Technical Specifications for Interoperability - European Standards for the interoperability of data including TAP (Telematics Applications for Passenger Services) and TAF (Telematics Applications for Freight)

UJG Uninterrupted Journey Group

UOMS Unattended Overhead Line Management System

UUID Universal Unique Identifier. A standard type of identifier generated by a computer algorithm, (nearly) guaranteed to be completely unique. Can be represented by a 128-bit binary number or a 36-character string of hexadecimal digits with spacers.

URI Uniform Resource Identifier. [3] Standard way of referencing an entity uniquely on the World Wide Web.

W3C World Wide Web Consortium. International community that sets standards to be used on the World Wide Web.

WSDL Web Services Description Language. [3] W3C standard XML-based method for specifying the structure and capability of web services.

XiRCM Cross-industry remote condition monitoring (in the rail industry) - remote condition monitoring occurring in: a) any of the quadrants where train based sensors monitor train, or infrastructure, and infrastructure based sensors monitor train, or infrastructure; b) across different parties and sections of the industry - different groups may be involved in the operation of an asset, provision of a sensor, monitoring of that sensor…

XiRCM Programme Phase 2 Cross-industry Remote Condition Monitoring Programme Phase 2 - a research project to facilitate the introduction of more cross industry remote condition monitoring, comprising packages to deliver architecture (01), commercial recommendations (02), review of standards (03) and decision support tool extension (04).

XiRCMSG Cross-industry Remote Condition Monitoring Strategy Group - a sub group of the Vehicle/Vehicle System Interface Committee with the responsibility to deliver the XiRCM Programme.

XML Extensible Markup Language - a widely used language used to define data structures (see Schema) whereby elements and attributes and their structure can be declared and assigned values.

Table of Contents

Cross-industry remote condition monitoring programme: Phase 2 Overview report

1 Introduction

1.1 The project

In 2013 the Cross-Industry Remote Condition Monitoring Strategy Group, a sub group of the Vehicle/Vehicle System Interface Committee, launched Phase 2 of the Cross-Industry Remote Condition Monitoring Programme. The purpose of the programme is to support the industry's high-level objectives of improving reliability, capacity and value for money, detailed in reports such as the Rail Technical Strategy [1] and Rail Value for Money 2011 [2].

The RSSB managed research programme T1010 is intended to set up the enabling framework for this initiative. T1010 is in four parts:

- T1010-01 – the current project – is to define a data architecture to support cross-industry RCM data sharing.
- T1010-02 is to define a commercial and legal framework to be used by parties wishing to share RCM data.
- T1010-03 will update industry standards and guidance to promote and simplify the use of the data sharing architecture.
- T1010-04 will update the business case assessment tool previously prepared under project T857 for RCM data sharing initiatives.

In 2013 Halcrow (now part of CH2M HILL) were awarded contract T1010-01 to develop the Cross-industry Remote Condition Monitoring data architecture for the UK rail industry. The data architecture is intended to remove some of the technical barriers to cross-industry data sharing caused by incompatibilities of approach between stand-alone RCM projects. It will do this by defining standard methods for structuring, representing and interchanging RCM data.

Remote condition monitoring is an aspect of asset management where the condition of an asset is monitored using data output from a sensor, so it can if necessary be monitored remotely from the asset itself (e.g. a person or system in a base station receives data directly from an asset on the railway).

Railway remote condition monitoring activities can be partitioned into four quadrants, defined by which types of asset, trains or infrastructure, are being monitored; and where, on trains or on infrastructure, the monitoring is being done. In this view, the two cross-interface quadrants, trains monitoring infrastructure and infrastructure monitoring trains, are of particular focus in this research as they involve different industry parties.

Figure 1 - The remote condition monitoring quadrants

train monitors train	train monitors infrastructure
infrastructure monitors train	infrastructure monitors infrastructure

However, the 'cross-industry' element of RCM data sharing is not limited to this view. Sharing of RCM data on the same side of the rail boundary can deliver benefits by breaking down information silos. One of the visions for the rail industry discussed in the Rail Technical Strategy is the concept of a whole system approach, with aligned asset management practices. The greater availability of shared data through the industry will enable a greater level of condition monitoring to take place with resulting improvements to reliability. This is in contrast to the current situation where many RCM systems in use and being developed operate as stand-alone systems where the data types, communication protocols and architecture do not necessarily conform to any agreed standards and can be tied to the equipment vendor.

1.2 Context

Work done so far on T1010-01 has produced a Review of Relevant RCM Developments [3], which has surveyed previous work on data integration in the rail industry, on rail RCM data integration in the UK and elsewhere, on relevant developments in the IT industry, on RCM data integration initiatives in other industries, and on the legal and contractual background to rail RCM activities.

Also in that document is a review of consultation with relevant parties: current RCM activity sponsors, academics, RCM system suppliers and the people responsible for new IT and information services in rail asset management.

The results of that work have been codified into a set of Influences on the RCM data architecture which inform its future shape and design. Based on these, a set of RCM data architecture Requirements have been defined which inform the development of an industry wide RCM data architecture and guide those responsible for implementing and managing it and participating in it.

The Architecture Requirements document sets out the following pillars on which the architecture is built:

- Use of existing ISO standards for referencing standard data items such as dates, times, locations, and engineering units.
- Use of standard representations of railway assets: track and fixed asset locations; vehicles, multiple units, train formations, trains.
- Formal recognition and use of the ISO 13374 segmentation of the RCM and Asset Management domain into six 'maturity levels': DA: data acquisition; DM: data manipulation; SD: state detection; HA: health assessment; PA: prognostic assessment; AG: advisory generation. This standardises and clarifies the business process elements at each layer, the data content and communication requirements between and within layers, and the location of intellectual property.
- Use of the MIMOSA EIA framework to represent RCM data at all the 6 maturity levels and for all asset types.
- Specialisations of this general framework for specific rail-based RCM functions.
- A set of standard queries and responses with industry systems to look up train or location information.
- Provision for a shared ontology of the rail infrastructure and rolling stock to facilitate mappings across different conceptual views of the network.

Building on the Principles is a set of architecture Requirements which adds a layer of detail and specificity into the Principles and gives additional guidance on how it should be implemented and governed.

These documents are technical and detailed in nature. This overview report summarises their findings in a narrative form and focusses on conclusions and recommendations to be taken forward by the GB rail industry.

1.3 Rationale and purpose of the architecture

The role of the data architecture is to provide an easily adopted and stable way for RCM data to be interchanged across the UK rail industry. To make the architecture stable, it will be based on open and common standards; to make it easily adopted, it will be closely mapped to the types of data interchange used and foreseen for the UK industry.

It is fair to ask why RCM data need to be interchanged in this way i.e. how the industry will benefit from removal of the technical barriers to data integration and how a data architecture will enable this. Previous work for RSSB under research projects T853 ‘Mapping the Remote Condition Monitoring Architecture’ [5] and T857 ‘Detailed Overview of Selected RCM Areas’ [6] has identified the key benefits:

- Reduced costs of asset data collection, processing, storage and retrieval through the use of standard data formats, referencing and interchange methods
- Improved understanding of asset degradation causes and relationships through merging of data from different assets in difference places, leading to more efficient interventions to maintain asset reliability
- Improved knowledge of the state of health of individual assets through merging condition / operations data from different sensors or measurement methods, leading to fewer in-service breakdowns
- Improved ability to predict the future health trajectory of assets through merging data on their current state with data about their predicted future usage or exposure to degrading factors
- Improved ability to support linkage of operations and maintenance business processes through the use of advanced software such as ontologies and reasoners, leading to reduced operational impact of asset failure or maintenance activity
- Ability to support optimisation of maintenance interventions through the linking of now separate IT systems and business processes managing maintenance planning and scheduling, operational planning and customer relationship management, resulting in reduced disruption and optimised overall system cost.

1.4 Scope of the architecture

As described above, RCM data are shared between parties in order to support business processes which have goals such as reliability improvement, maintenance of safety and cost reduction. These processes always involve data from other sources, such as asset management systems, operational systems, and work planning systems. They are informed by RCM data, but also involve data of their own which are not directly RCM-related.

The RCM data architecture described in this document covers the following scope:

- Data directly associated with remote condition monitoring: sensor data in raw and processed forms; alerts and alarms; health assessments and estimates of usable life or future condition.
- Data needed to identify the assets described by or affected by the RCM data.
- External data on usage or other influences on future life or state such as weather.

The architecture does not restrict the type of rail asset being considered (moving or fixed, point or linear), the type of data being collected (analogue and digital, time-based, frequency-based or event-driven, alerts, alarms and health assessments), or the level of analysis (raw, processed, assessed against tolerance criteria, merged with other data). It will therefore potentially work with any current rail RCM initiative and with any reasonable future one.

The architecture does not demand large scale investment in hardware or software to be brought into being, though some of the more sophisticated applications at the higher maturity levels will benefit greatly from some shared IT infrastructure, for example to manage shared reference data and to mediate message transfers between data originators and data consumers.

1.5 Uses of the architecture

It is envisaged that the architecture will have the following uses:

- As a set of specific requirements on data sharers which can be included in agreements between data providers and data users, which will simplify their own task of interchanging data while furthering the broader goals of industry data availability.

- As requirements on the developers and managers of any shared IT infrastructure set up to support data interchange in an industry-standard way
- As guidance on the principles to be followed in managing the architecture over the longer period to ensure that its goals are realised.

1.6 This document

The remainder of this document is organised into these sections:

Section 2 summarises our Review of RCM Developments and the conclusions and guidance taken forward from it to the data architecture.

Section 3 outlines the principles of the data architecture and how they contribute to the goals of the Cross-Industry RCM Strategy Group.

Section 4 considers the implementation of data integration, the roles of the various parties and the ways in which the data architecture can be retro-fitted to existing industry IT systems.

Section 5 covers the governance of the architecture and the management of the physical elements which are shared by all participants and so require central control.

Section 6 contains recommendations for future cross-industry RCM work and for the industry at large.

Section 7 is a list of references to other documents and sources cited in the text.

2 Review of recent RCM developments

2.1 Introduction

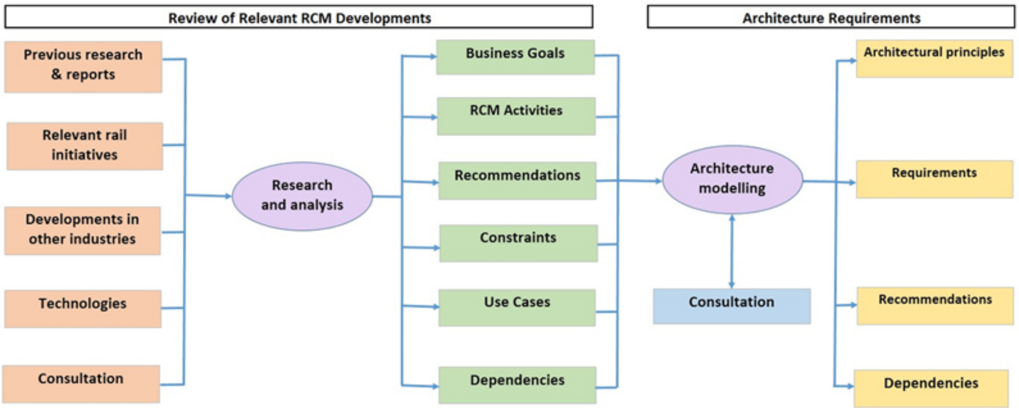
The first part of the T1010-01 remit was to carry out a review of relevant documentation and developments. The scope was to include:

- Previous work in the area of cross-industry RCM
- Other RCM-related activity in the rail industry in the UK and elsewhere
- Initiatives in other industries that might be relevant
- IT developments, standards and initiatives
- Views of significant stakeholders and influencers in the UK rail industry: Network Rail, Association of Train Operating Companies (ATOC), train builders and maintainers, software and hardware suppliers

The output of the work was a set of factors which influence the principles and specification of the data architecture, expressed in the form of:

- Architectural Principles: guidance on the type of architecture most suitable to support the business needs of RCM data interchange
- Requirements: firm functional items which the data architecture must deliver
- Recommendations: suggestions on features of the architecture from stakeholders or other similar initiatives
- Dependencies: requirements on other parties or IT systems outside the direct scope of the data architecture, necessary to make it work.
- The process followed to get from the input documentation to these outputs is shown in Figure 2.

Figure 2 - Review of Relevant RCM Developments Process Flow



The key tasks were research and analysis of the inbound documentation to produce a set of interim findings, followed by modelling and synthesis of these into the outputs.

Care has been taken to ensure that each relevant finding is represented in the output influencing factors i.e. that nothing important has been missed out; and that each influencing factor can be traced back to an initial finding – that there is no 'gold-plating': requirements which have no business justification.

The full results of this analysis are presented in the document T1010-01 Review of Relevant RCM Developments [3]; a summary of the key findings is presented in the sections below.

2.2 Industry context and binding standards

2.2.1 Industry context

The driving force behind the work of the Cross-Industry RCM Strategy Group is the aspiration set out in the Rail Technical Strategy 2012 (RTS) [1] to drive improvements in maintenance cost and system performance by the use of RCM. This document identifies a cross-industry technical and commercial framework as an enabler of better exploitation of asset condition data; and encourages a whole-system approach to breaking down process and data silos which hamper cross-industry process improvement.

The data architecture represents a key part of that technical and commercial framework. The way in which it is defined in the architecture Requirements supports the whole-system approach by using open standards and data interchange methods not tied to specific IT systems or industry processes.

The data architecture supports the RTS's goal of improving asset knowledge through the capture and use of life cycle degradation information and application of advanced theoretical modelling. It takes cognisance of specific recommendations:

- The adoption of international standards for RCM, data processing and asset management
- Consolidation and linking of existing asset management systems
- Integration of internal and external information
- Use of open standards and COTS equipment
- Setup of shared data types, data dictionary and ontology.

Network Rail's response to the RTS is its own Technical Strategy 2013 [7] which picks up the themes established in the Rail Technical Strategy and defines a portfolio of research and development initiatives with specific goals. Several of the themes and research portfolios overlap directly with the scope of the cross-industry RCM data architecture; and the architecture will generate indirect benefits for several others:

- Under the Whole System Approach theme, the data architecture directly impacts the short-term goals to develop modelling capability to understand trade-offs in operation and to improve whole-life costing; to turn data into information for engineers; to adopt worldwide best practice; and to improve safety of trackside workers by increasing the use of remote monitoring of fixed assets. In the longer run the architecture will support the need to double passenger capacity in 30 years by improving network availability and performance and improving the precision and level of integration of railway operations and maintenance processes.
- Under the Information theme, the data architecture directly supports the goals to automate the collection of asset information, the interchange of messages across the industry and international borders, associated processes such as analytics, reporting and augmentation. It will also indirectly support the improvement of data management and governance and the application of integrity standards by demanding high-quality reference data to support cross-industry processes.
- Under the Command and Control theme, the data architecture supports the goal of improved recovery from perturbation. It could also drive adoption of the Compass train positioning programme.

- Under the Energy theme, the data architecture will support moves to improve the reliability of the overhead line equipment by automating the assessment of its condition and by reducing time to fix.
- Under the Infrastructure theme, the data architecture contributes to remote condition monitoring of earthworks, to the real-time monitoring of assets, building on the Intelligent Infrastructure programme.
- Under the Rolling Stock theme, the data architecture supports the remote collection of condition data, particularly at the system interfaces: wheel/ rail, pantograph/OLE, shoe/3rd rail, train to ground telecommunications. It will also indirectly impact the standardisation of on-board equipment and the move to reliability-centred maintenance of rolling stock.

The Railway Functional Architecture, resulting from RSSB research project T912 [8], sets out an overall view of the entire railway system in terms of a number of different perspectives. The most relevant are the 'Concept' and 'Solution' perspectives. Concepts respected by the data architecture are shown in Table 1.

Table 1 - Railway Functional Architecture views relevant to RCM

Ref	Title	Elements incorporated in view:
CV03	Condition monitoring	asset, condition monitoring agents and maintainers
CV01	Maintenance	asset, condition monitoring agents and maintainers
SV04	Rolling stock condition monitoring	functions such as HABD and systems and standards
SV04	Signalling and control infrastructure maintenance	condition monitoring and analysis, scheduled maintenance and corrective maintenance
SV04	Track condition monitoring	various automated track inspection functions
SV04	Train service location	various functions involved in determining velocity and location of trains

Table 1 - Railway Functional Architecture views relevant to RCM

Ref	Title	Elements incorporated in view:
SV01	Condition monitoring	the various physical elements and systems involved in train-borne and trackside monitoring
SV01	Hot axle monitoring	track-mounted and train-borne
SV01	Track Inspection	automated track inspection
SV01	Automatic train location service	train and track based systems

The T912 report also sets out some desirable characteristics of system architectures which the data architecture respects:

- Modularity (defined as the degree to which a system's components may be separated and recombined, the tightness of coupling between components, and the degree to which component relationships may or may not be allowed).
- The importance of eliminating 'sub-optimal modular interfaces': cases where the same item or interaction element exists in multiple relationships.

2.2.2 Guiding standards

Data interchange in several railway contexts is governed by existing regulations and standards. Those considered for this work are:

- Technical Standards for Interoperation (TSIs) for Telematics Applications for Passengers Services and Freight (known as TAP-TSI and TAF-TSI). These are binding documents adopted by the European Commission. These TSIs set out details of data items and their formats describing trains and their routes in a messaging format. The data architecture respects these data items.
- UK RGSs and RISs relating to rolling stock, infrastructure and monitoring equipment data storage and interchange. A total of 13 RGSs and 1 RIS were reviewed and their data requirements picked up in the data architecture.

2.3 IT developments

Recent and ongoing developments in the IT industry have greatly improved the prospects for implementing a data sharing architecture which meets the rail industry's requirements. These relate to new standards and new technologies. The most significant of these, covered in depth in [3] and explicitly recognised in the data architecture, are:

- Enterprise Integration architectures - a set of principles and techniques to assist in the integration of business processes in a robust and economical using existing IT systems and a 'data bus' structure.
- Web Services and REST – a way of providing data and processing using simple existing open standards already commonly in use on the World Wide Web.
- Web Ontologies – a set of technologies based on internet standards, helping to move the concept of Ontologies from the academic domain into everyday application in industry.
- Big Data standards – new and emerging standards for the storage and referencing of bulk data such as that generated by video or by large numbers of sensors.
- InteGRail – a European Union project which developed a proof-of-concept for cross-industry data and process integration based on EIA and Ontology principles.

These are described in the following paragraphs.

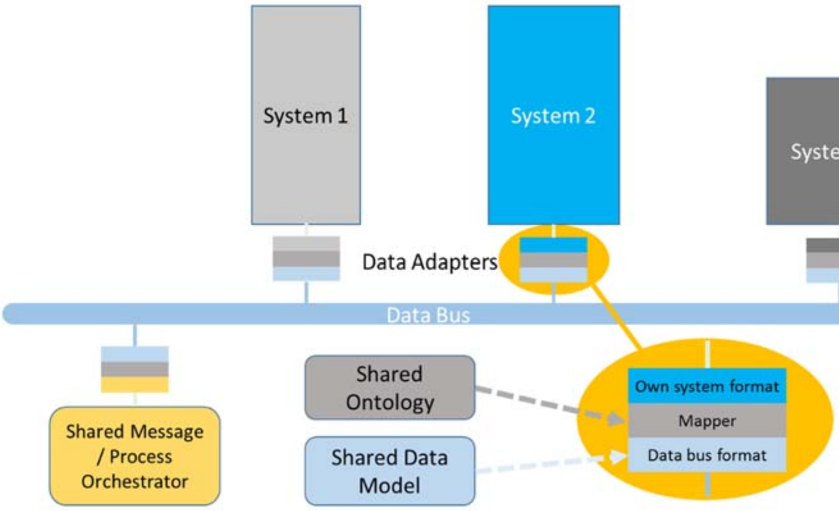
2.3.1 Enterprise integration architectures, web services

The need has arisen in recent years to automate business processes which cross organisational boundaries, whether between organisations or between functions within organisations. Typically, existing IT systems are department or function-based and so unsuited for this type of automation.

Enterprise integration architecture techniques have been developed to address this need. The key techniques are:

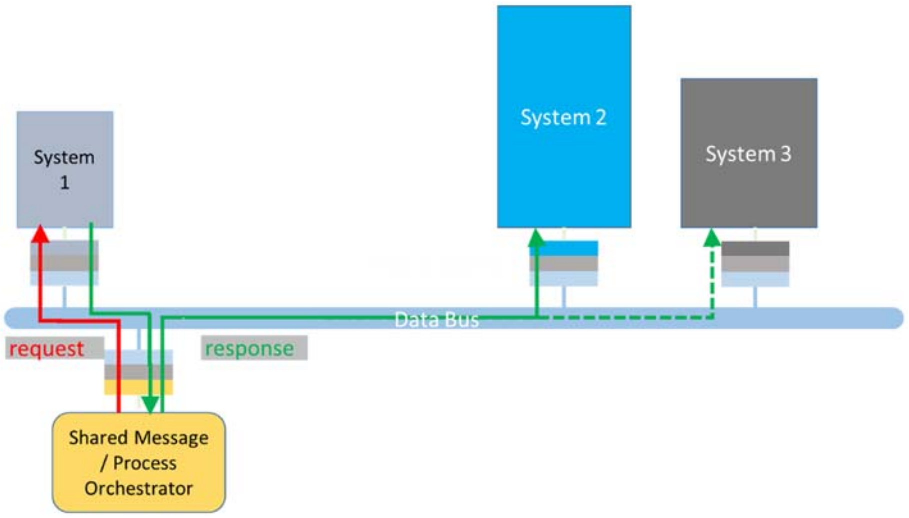
- 'Bus-style' structure to reduce the number of point-to-point interfaces. The bus uses a standard shared data model. Each participating IT system connects to the bus using an adapter which maps the data formats and structures from that system's own to those of the bus. (Figure 3 shows the essential elements of a bus-style architecture. A shared data model defines the data formats used on the bus; each adapter maps from its system's own formats to the bus formats; an ontology layer can be used if required to handle semantic differences between the system and bus representations).

Figure 3 - Bus-style architecture



- Interaction via web services. Each participating IT system responds to web requests to perform processing and/or return results via web messages. Standard internet protocols and data formats are used to handle the requests and responses.
- Use of messaging. Services are invoked and responses returned using messages, which enable systems to co-operate asynchronously and for different paradigms of data sharing to be employed (such as direct point-to-point messaging, publish or subscribe). (Figure 4 shows a representative message-based interaction, where a messaging service requests data from one system and distributes the responses to several others using a publish or subscribe approach).
- Use of open standards. All data are interchanged using data packets defined using standard methods such as XML or JSON; web services are defined using standard methods such as SOAP or WSDL.

Figure 4 - Message-based interchanges



The benefits of these architectural approaches are:

- Existing systems can be linked together without major adaptation. Existing data outputs can be used where appropriate; existing functions can be wrapped with a services layer to make them more accessible.
- Systems are loosely coupled, so they don't make major demands on each other's availability nor require them all to upgrade or be modified in step with each other.
- New systems can be connected together easily and the barriers to entry gradually lower as adapters are developed and extended.

Significantly, several major system developments in the rail industry are using an EAI style approach to their architecture. This both validates the technical approach and will tend to simplify the setup of the necessary supporting industry services such as vehicle identification, asset reference or co-ordinate translation:

- LINX TM uses a commercial ESB product and a message-based architecture.
- R2 will use a message-based architecture based on a data bus for integration with mainframe systems.
- ORBIS is conceived as a web services based application suite.

2.3.2 Web ontologies

For several years, academics and specialists have been making the case for the use of ontologies as enablers of process and data integration in the rail industry. The development of new web-based standards and tools for creating, managing and applying ontologies has opened up possibilities for applying the approach in real data integration scenarios.

An ontology adds a layer of meaning on top of a shared data model because it is capable of representing the concepts implicit in the data model and the relationships between them. This means it can address 'semantic heterogeneity', different interpretations of concepts, between different IT systems, on top of the 'structural heterogeneity', different data structures, addressed by a shared vocabulary or data model. This simplifies the process of integrating these IT systems.

An ontology also allows relationships of meaning to be defined between different data representations, allowing the ability to 'reason': to draw conclusions about data items which are implied by the semantic relationships contained in the ontology, without them being explicitly stated. This enables a new class of software tools to be developed which can interrogate the ontology to drill into these relationships or chain them together.

Some areas where the power of the ontological approach can add value in the RCM domain are:

- Understanding the detailed structure of complex assets. This supports, for example, deriving health assessments for a composite asset (such as an operational train) made up of a collection of simpler assets (such as a train made up of two units, each comprising 4 vehicles, each with 2 bogies, each with 2 axles, each with 2 bearings, each of which we have a condition rating for).
- Bridging operational and engineering domains. This supports, for example, calculating the impact in passenger lateness minutes for a morning peak for a section of route based on the inability to operate a single switch and thus a reduction in the number of possible routes through an interlocking; or the calculation of the expected usage of a single section of OLE based on the future train timetable.
- Representing fault trees and failure modes in a shareable and extensible way, including with fuzzy logic. This will enhance the ability to build up best-

practice fault-finding and diagnostic algorithms and link asset state to likely future life.

All these areas occur at the higher levels of the ISO 13374 maturity hierarchy. An ontological approach can thus be seen as a key enabler of these levels.

The key technologies associated with web ontologies are RDF, SparQL and OWL.

Resource Description Framework (RDF) is a simple method for expressing facts about concepts or objects using a simple 'triple' arrangement of subject - predicate - object (such as: wheelset IsPartOf bogie, where wheelset would be the subject, bogie the object and IsPartOf the predicate describing the linkage or relationship). SparQL is a query language for data expressed in RDF triples, similar in concept to SQL used to query relational databases. An RDF data source queryable using SparQL is called a SparQL Endpoint, it is usually represented as a URL web address.

OWL is the Web Ontology Language, a way of specifying ontologies using web standards. It includes all the ideas in RDF, and adds to them some key features such as the ability to define classes of objects by rules on other objects (for example: that a train set with a pantograph or with a 3rd-rail pickup is an electric train set); to combine these rules using the ideas of set theory (UNION, INTERSECTION); to have rules based on the number of objects in a class; to verify the correctness or validity of the ontology by detecting contradictions (where the rules defined are logically inconsistent) or unsatisfiable classes (where the rules mean that no actual individual object could be in the class).

The data architecture does not demand or require the use of an ontology, but it does offer features which will enable them to work effectively when the business need for them arises:

- A proposed method to expose data from connected systems as SparQL Endpoints, making their data available for ontology-aware software.
- A data adapter structure which allows for an ontology-driven middle layer (as shown in Figure 3) to carry out any advanced semantic mapping required between existing IT systems and the shared data model.
- A metadata structure which allows any data being transferred using the data architecture to be augmented with additional data items describing its source, owner, quality… These are expressed using standard terms from well-known ontologies such as the Dublin Core [9].

We also propose actions and recommendations to facilitate the creation of shared ontologies to support cross-industry data integration.

2.3.3 InteGRail

InteGRail was an EU-funded project to generate a proof-of-concept and define a roadmap and best practices for data interchange across the rail industry. Its scope was broader than RCM, since it also included interest in other integration challenges such as network capability analysis and fleet management.

The key elements of the InteGRail architecture were:

- An 'Integrated Service Grid', conceptually a data bus architecture with adapters as described in Section 2.3.1.
- An Ontology to represent the shared conceptual model of the railway.
- Distributed computing elements, including 'reasoners', communicating using services.

The project resulted in the creation of a number of working applications based on this architecture, plus a very useful body of knowledge and experience which has been drawn on for the current work.

The same basic IT architecture used for InteGRail is used in the data architecture since it successfully enabled the sample applications ('Development Scenarios') to be built; and it exhibits many of the good practices already outlined above under Enterprise Integration architectures and Web Ontologies.

2.4 Existing and emerging standards

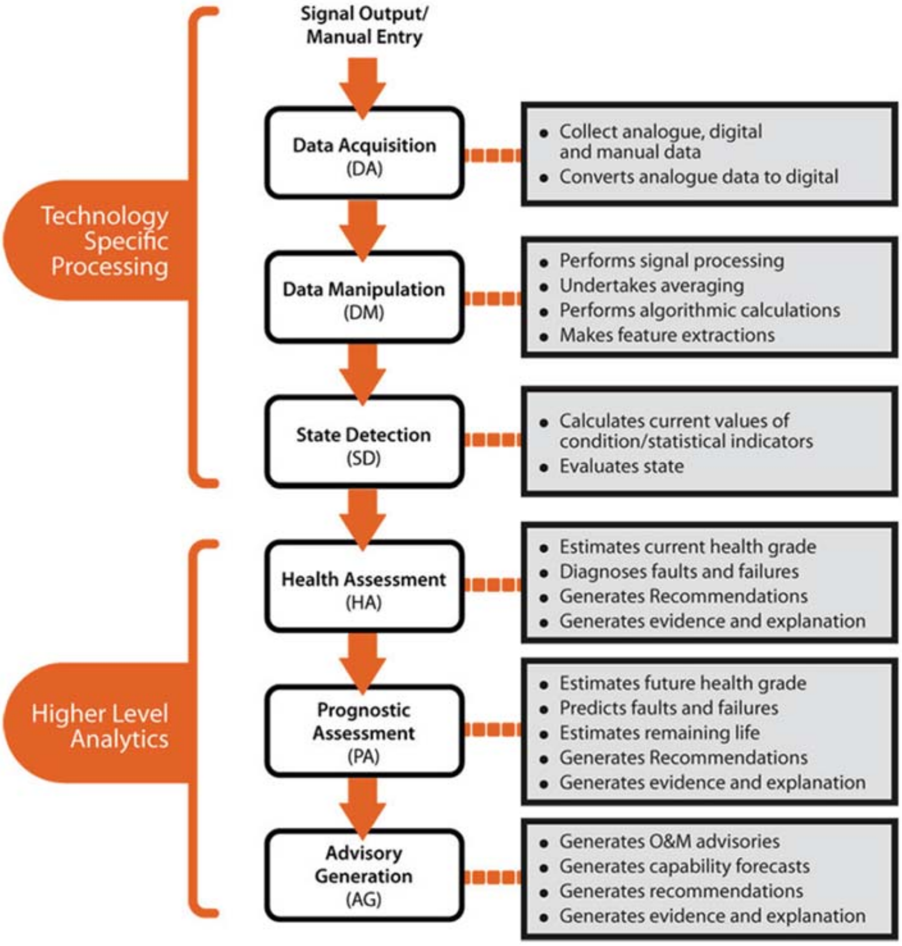
Within the arena of IT integration, RCM and data sharing in the rail industry and similar industries, a set of overlapping and complementary standards has been evolving in recent years. Many initiatives have been made to try to bring order to this very complex set of domains, resulting in a plethora of standards and guidelines of different levels of generality, acceptance and applicability. From this large field, a set of genuinely open standards has been identified which are compatible with each other and which are most likely to stand the test of time. These are described in the paragraphs below.

2.4.1 ISO 13374

This ISO standard ‘Condition monitoring and diagnostics of machines – Data processing, communication and presentation’ [10] is intended to enable condition monitoring data of all types to be processed and shared by heterogeneous IT systems. At its core is the organisation of the IT functions associated with RCM into a 6-level hierarchy, with standard data formats and representations identified as the input and output of each layer.

The 6 layers of the hierarchy and their basic functions are shown in Figure 5 (taken from T844 [11]).

Figure 5 - ISO 13374 layers


Signal Output/ Manual Entry
Technology Specific Processing
Higher Level Analytics
Data Acquisition (DA)
Collect analogue, digital and manual data
Converts analogue data to digital
Data Manipulation (DM)
Performs signal processing
Undertakes averaging
Performs algorithmic calculations
Makes feature extractions
State Detection (SD)
Calculates current values of condition/statistical indicators
Evaluates state
Health Assessment (HA)
Estimates current health grade
Diagnoses faults and failures
Generates Recommendations
Generates evidence and explanation
Prognostic Assessment (PA)
Estimates future health grade
Predicts faults and failures
Estimates remaining life
Generates Recommendations
Generates evidence and explanation
Advisory Generation (AG)
Generates O&M advisories
Generates capability forecasts
Generates recommendations
Generates evidence and explanation

In terms of railway RCM, an example of application of these levels would be (also taken from T844):

1 Data Acquisition (DA) example – A 16.2mA signal represents an instantaneous point motor current of 12.3A.

2 Data Manipulation (DM) example – Peak point motor current was 14.7A with an average of 11.4A.

3 State Detection (SD) example – A peak point motor current of 14.7A is above the alert threshold of 13.8A (but below the alarm threshold of 15.7A) and is an alert condition.

4 Health Assessment (HA) example – A peak point motor current of 14.7A with the standard deviation of swing time from normal to reverse exceeding 1.5 seconds indicate points are out of adjustment and require readjusting. The need to re-adjust has been caused by track voiding. These points have a Health Index of 7.

5 Prognostic Assessment (PA) example – Points with a Health Index of 7 which require adjustment will reach a Health Index of 4 in 6-8 weeks based on a forecast traffic volume of 50 EMGTPA and 68 points swings per day. Adjustment will require 2 maintainers for 2 hours and adjustment kit SK154. Resolving track voiding will require track tamping.

6 Advisory Generation (AG) example – Three work orders to be issued:

   a Temporary adjustment of points to mitigate deterioration in Week 1.

   b Track tamping to take place in Week 7 as tampers will be operating in adjacent possession and two signalling maintainers will also be available.

   c Immediately after tamping, the faulty points should also be adjusted using the SK154 adjustment kit which will be available from week 4.

The data architecture formally recognises this structure of processing layers. Its value comes in two distinct ways:

- It enables the plethora of potential types of RCM data to be described by a simple set of relatively straightforward data types, thus enabling the creation of a maintainable architecture.
- It clarifies where intellectual property lies and what it covers in the RCM domain. This is within the processing elements. The data passing between the elements need have no IP restriction and can be in open formats. This opens the way to an open architecture of collaborating software elements.

2.4.2 ISO 15926

This ISO standard ‘Industrial automation systems and integration—Integration of life-cycle data for process plants including oil and gas production facilities’ [12] defines methods for representing the types of data that occur in RCM applications so that they can be transferred and understood across system, function and organisational boundaries. Though originally set up for the oil industry, its concepts are sufficiently general that it can be applied elsewhere. It covers:

- A shared Conceptual Data Model – terms on which all parties can agree
- A shared Reference Data Set – common lookup and reference data
- A shared Ontology – descriptions of the meanings of terms
- A method to define and build Templates – application-specific implementations of the general data model for particular uses.
- A way to request and receive data – an interaction protocol, based on RDF, or SPARQL.

All these concepts are reflected in the data architecture. Important aspects of the architecture for handling RCM data are defined using MIMOSA standards (see Section 2.6.1). MIMOSA is an implementation of the ISO 15926 concepts.

2.4.3 RailML

RailML is an open XML schema standard for the interchange of rail data. Whilst it started life in mainland Europe, it is gradually finding use in the UK as well, with work being done at Network Rail to make it consistent with the ORBIS infrastructure asset information programme. RailML can represent rolling stock, infrastructure, timetables and metadata.

A key recent development in RailML is the effort in the latest version 3.0 to make the infrastructure element compatible with the RailTopoModel [13] representation of railway network topology managed by the UIC. This means that there will be a standard way of representing not just the physical structure of the rail network, but also common higher levels of abstraction of the network such as the timetabling view, the passenger journey view and the corridor view. Other views are also possible; and standard methods are available from moving from one view to another.

This consistent topological model will be of great importance in supporting the ontological and reasoning processes which will be needed to calculate, for example, the impact of failures of specific infrastructure assets on the passengers using the routes they are part of.

The data architecture uses RailML concepts and data structures to reflect the physical infrastructure of the rail network.

2.4.4 RSSB research project T990 Train location strategy

This research project sets out the principles and architecture for a comprehensive Train Location service which combines information from all available sources to give a single most accurate view of a train's location. The strategy has both train-borne and ground-based elements. It sets out the data content of train location datagrams, including geographic and track positions and confidence or data quality indicators for position. Optional data items would include train consist, length and direction.

The RCM data architecture will use train location information and so interact with any realisation of the Train Location Strategy in a number of ways:

- As part of a data service to link raw train-gathered RCM data to train and track location information. This will be relevant where trains are monitoring the infrastructure and therefore need to specify its track location.
- As part of a data service to identify the train passing a particular point on the infrastructure. When combined with RFID-based vehicle identification, this would give a comprehensive train and consist service enabling low-tech detectors such as HABDs to contribute useful shared RCM data.
- As part of a comprehensive train consist data service which would enable a full consist of a train to be retrieved given some basic information.

The work on this standard also introduces the useful concept of grades of data quality for a data service, wherein the same basic service can offer responses at different levels of precision, accuracy or detail depending on the type of query or the quality of the source data available at that time or place. This concept is used in the data architecture in some of the Reference Data services such as a Train Identification service.

2.5 Rail RCM projects

Several studies have been carried out by RSSB for the Cross-industry RCM Strategy Group, looking at RCM and other areas where easier or deeper integration of data would benefit cross-industry business processes. For this study we have reviewed the following:

- RSSB research project T844 'Mapping Current Remote Condition Monitoring Activities to the System Reliability Framework' [11]. This document identified the types of RCM activity that could offer the biggest potential savings in delay minutes and set out some key developments that would need to take place to bring them about: AVI, better predictive algorithms and a shared industry-wide ontology to facilitate data sharing. The data architecture advances and recognises all of these.
- RSSB research project T853 'Mapping the Remote Condition Monitoring architecture' [6]. This document examined the system and technical architectures in use for current RCM activities and benchmarked them against the 6-level RCM processing maturity hierarchy established by ISO 13374 Condition Monitoring and Diagnostics of Machines [10]. Nearly all of the systems work at the State Detection or Health Assessment levels; none so far reaches the Prognostic Assessment or Advisory Generation level. The report identifies other ISO standards and precursors that should provide input to the data architecture: the architecture takes forward the ISO 13374 hierarchy, the ISO 15926 data integration stack and the basic architectural approach used by InteGRail.
- Data Framework Feasibility Study 2011 [14]. This investigated the feasibility of a data-sharing architecture with a shared ontology at its core and concluded that there was a clear case for such an approach. Barriers to be overcome included the effort to create the core ontology, the need for clear industry-level leadership in hosting and managing the ontology, sensitivity about data ownership and access, and the need to enforce participation where immediate interests of parties and the industry at large are not aligned. The data architecture enables the ontology approach and considers the data ownership aspects and governance arrangements.
- RSSB Cross-Industry Information Systems Workshops 2012 / 13. These considered current major IT initiatives in the UK rail industry and what might be gained by harmonising them under the umbrella of an industry-wide information architecture. Key overlaps between initiatives were identified, as were some important enabling technologies and pointers to suitable future architectures. The data architecture takes forward the importance of Ontologies as a unifying technology, the recommended Enterprise Integration architecture approach and common data model, and

recognises how the key enablers of AVI and Location Definition would be applied to RCM data.

2.6 Similar activities in other industries

Several data integration initiatives from other industries have been studied to identify useful standards for RCM data and best-practice approaches to employ in designing and managing the architecture.

2.6.1 MIMOSA

MIMOSA is a standardisation organisation set up to enable harmonisation of data across a number of process-related industries. It has set up a reference data model and set of interaction standards based on the ISO 13374 hierarchy of RCM functions and the ISO 15926 concept of shared data model and ontology. The reference model is called the Open Systems Architecture for Condition Based Maintenance (OSA-CBM) [15] and the interaction standards are called the Open Systems Architecture for Enterprise Application Integration (OSA-EIA) [16]. There is also a specification for a method of managing and issuing unique identifiers for assets and components, called MIMOSA Structured Digital Asset Interoperability Register (SDAIR) [17].

OSA-CBM is a definition of RCM data types based on the layers of the ISO 13374 RCM hierarchy (Section 2.4.1) and the types of data that are transferred into and out of each layer. The data types are completely general in terms of the sensor type, asset type and data values being described. It includes a highly-generalised definition of a data model for assets, the relationships between them, measurements that can be taken about them and the types of data needed to describe all of these. This data model is capable of representing, for example, elements of a railway network or of items of rolling stock.

OSA-EAI is a definition of interchange formats and methods based around a layered architecture of data structures, each of which builds upon the one below. The structure maps on to the ISO 15926 structure (see Section 2.4.2), has parallels in the InteGRail architecture (see Section 2.3.3) and can be mapped on to other well-known railway data structures such as those implied by RailML (Section 2.4.3). At the bottom is a shared vocabulary of terms and concepts (a domain ontology). Based on that is a conceptual data model, which is then used to define implementation-level data models used for information exchange. Supporting these is a set of reference data; and at the top are a set of defined document types for data interchange management.

SDAIR defines a data structure and set of processes to issue, query and manage unique identifiers for functional locations (= assets in common parlance) and equipment (serial-numbered items performing the functions of the functional locations at a given time). The goal of SDAIR is to bridge across different systems and business functions which have different coding schemes and conceptual views of the same physical item.

All of these are highly relevant to the data architecture and have been adopted as its founding structures. The RCM data interchange elements are based closely on the RCM data types of OSA-CBM; the shared data elements for assets and equipment use the concepts and hierarchy of OSA-EAI; and the concepts described in SDAIR are used to recommend ways of managing the uniqueness and translation of asset identifiers.

2.6.2 INSPIRE

INSPIRE is a European spatial data infrastructure focussed on the interoperability of network services such as transport, utilities and geographic features, with the objective of combining spatial data without specific computer or human action, to enable sharing and open access to data by organisations and the public.

Its relevance for the data architecture comes from two aspects of the programme:

- Railway infrastructure data and its representation come within the scope of INSPIRE, so the data architecture needs to recognise and be compatible with its formats.
- The process followed to gather the requirements for INSPIRE and set up an architecture to meet them closely mirror that being followed for cross-industry RCM.

These have both been taken to account in the definition of infrastructure data types and in the recommendations in the *Principles of the data architecture* and *Data models* sections, that cover implementation and governance, in the data architecture report.

2.6.3 Aeronautical Information Exchange Model

The Aeronautical Information Exchange Model (AIXM) is an information exchange model used in the aeronautical industry to enable machine-to-machine or service-to-service management and distribution of data. It is based on a data schema primarily concerned with airspace definitions and communications of those definitions, rather than on assets. However, there are a number of ways in which it is comparable to the objectives that are being worked towards in the rail industry. It thus furnishes guidelines and principles which the data architecture can usefully follow:

- The demands of the user community are similar: disparate existing IT systems needing to merge data from different sources.
- Each user of the architecture is only interested in elements of it, not the whole thing. A modular design approach is therefore suggested.
- Each data interaction carried out by users will need to add its own data to that mandated by the architecture, similarly to the case with the data architecture. Extensibility is thus a key requirement.
- The use of open standards is heavily recommended.
- A clear concept of asset identity and the importance of the time dimension are necessary.

2.7 Stakeholder consultation

A series of consultations was held with a variety of industry stakeholders with an interest in RCM data from one or other perspectives. These were held as a series of face-to-face meetings, attendance and follow-up on the two case studies used in the parallel T1010-02 work on commercial agreements, and stakeholder group workshops.

Through the consultation, a number of clear themes has emerged which the data architecture must address. These are described briefly in the following sections.

2.7.1 Identification

The ability to quickly and accurately identify a vehicle or component is critical to realising the full benefits of an RCM data framework. In practice this ‘identification’ means the matching of data gathered by one component (for example, a trackside monitor) with the identity of the component to which the data corresponds (such as a passing vehicle); or accurate identification of location, being able to match the location of a fixed asset to a piece of data gathered by a moving vehicle.

2.7.2 Improved analysis and information

A clear point arising from research indicates that simply making the data available will not deliver the sort of improvements to reliability and costs that are envisaged. It must be accompanied by improvements in the analytical capabilities of those using the system, realised through the improvement of algorithms upon which the analysis is based. Alongside this there is an additional dependency for a better information base on all aspects of asset management. For example, asset configuration data will be important in component identification, deterioration profiles are needed to support assessment of an asset's health or lifespan and so on.

2.7.3 Integration with existing systems

RCM-based data can enhance and improve the efficiency of existing business processes for train operations, maintenance and asset management. These processes are mediated by existing information systems operated by different stakeholders. It is clear that the RCM data architecture therefore needs to enable communication between these existing systems and those offering RCM data. There should be no wholesale replacement of asset management, work management or rail operations systems.

2.7.4 Reuse, standards and interoperability

The RCM data architecture will comprise definitions of data items, data types, and their associated parameters. In many cases these definitions will already exist and may be standardised. For example, asset management concepts are already clearly defined in ISO standards and rail terminology in the RailML schemas. Although this will save work in the definition stage of the architecture the clear benefit is of course the ease and consistency with which RCM can integrate with any system which uses the same definitions, in other words, interoperability.

2.7.5 Modularity and extensibility

The overall scope of the RCM data architecture is broad, covering many different types of RCM data for many different types of asset, generated by different types of equipment and IT system and offered in different data formats, being consumed by different types of operations and asset management users and systems. The RCM data architecture will therefore comprise many parts and will need to be constructed and evolved over time as uses for it become clear. Modularity and Extensibility are two architectural concepts which will help make this possible. Modularity means that the architecture should consist of parts which are as far as possible independent of each other. Any given RCM project should only need to know about the parts directly relevant to its needs and should not need to drag in a large number of unnecessary extras.

Extensibility means that the architecture can be augmented or enhanced over time or for particular purposes, without forcing users who do not need the extensions to upgrade unless they wish to.

2.7.6 Web ontologies

Many of the sources researched and several of the contributors to work so far on rail data integration have indicated the potential value of semantic metadata to achieving the higher-level goals of data and process integration in rail operations and asset management. This value has been demonstrated in prototype in, for example, InteGRail. Web Ontologies and their attendant standards and technologies are seen as the most likely way in which semantic metadata will be implemented in future. However, the technology still has some way to go before being ready for full-scale rail operation.

The RCM data architecture will need to be defined in such a way that it can accommodate future web ontology developments. The impact will be in methods of structuring and querying data from existing systems in a compliant way, and in the use of common ontological methods for representing standard information types.

2.8 RCM systems and data

An overview was carried out of existing UK rail RCM systems and the types of RCM data they generate or use in terms of the ISO 13374 stack. This was done for two reasons:

- To assess whether the RCM data types referenced in the standard and in the MIMOSA implementation of it were adequate to cover all the data actually expected in UK rail systems and cross-industry interchanges
- To identify classes of system and their users which would feed forward into the analysis of Use Cases for the data architecture.

41 RCM systems were evaluated. For each, the following data items were gathered:

- Which of the industry quadrants were covered (See Figure 1)
- The level(s) of the ISO 13374 hierarchy at which outputs were available
- Characteristics of the data generated, including: binary, text, multimedia; structured, continuous, event-based, real-time; in open or proprietary formats.
- Quantities being measured, including: acceleration, speed, pressure, voltage, and temperature.
- Asset types being monitored, including: track, OLE, plant, structures, bearings, and wheels.
- Type of location data involved, including geographical (lat/long), odometer, track-based, and topological.
- Type of vehicle identification used or needed.

The data architecture has been defined to ensure that all of the data types and characteristics can be represented.

2.9 Outputs of the review

All of the findings from the research and review activities have been captured in a requirements tracking tool and analysed into a set of formal requirements which have been fed forward to inform the data architecture Principles and Requirements. They are listed in Appendix D of [1].

An analysis was also done of the Actors and Use Cases to be supported by the data architecture. These have been used to validate that the data architecture contains the elements necessary to support the tasks it will be needed to help with.

2.9.1 Research findings

The findings of the research, document review and stakeholder consultation have been summarised into these groups of requirements:

- Business Goals. These are high-level industry objectives which the data architecture will be supporting or enabling. They cover aspects such as system reliability and safety, best-practice asset management and cross-industry operation decision support.
- RCM Activities. These are actual RCM functions, systems and associated processes that the industry presently carries out or wishes to in future. They have the status of requirements on the data architecture directly to support.

- Recommendations. These are suggestions or guidance given by previous work or learnt from observing other similar activities. They typically relate to how the data architecture should be designed or managed.
- Dependencies. These are activities, systems, organisations or processes external to the RCM data architecture itself which must be present in order for it to be able to fulfil its requirements. They typically relate to the ability to call upon external information on assets, their location, their identity, their structure, their future usage and causes for their degradation and failure.

2.9.2 Actors and use cases

Based on all the RCM activities and the industry business processes where shared RCM data has a bearing, a set of Actors and Use Cases was put together.

Actors are humans or IT systems fulfilling a role. They are based on actual people and tasks, but abstracted so that they do not depend on specific individuals or job descriptions, but rather describe the functions needing to be performed.

Use cases are instances of an Actor interacting with the RCM data architecture to fulfil a business need. They were grouped into the following categories:

Table 2 - Use Case Categories

Use Case Category	Description
Content	Use cases relating to the different types of data content that actors might require or supply. The data content types are defined in terms of the ISO 13374 stack.
Delivery	Use cases relating to the different ways actors may provide or receive RCM data
LifeCycle	Use cases relating to the life cycle of individual assets: addition, movement, disposal.
Process	Use cases relating to the gathering and processing of RCM data. We have used the layers of the ISO 13374 stack to distinguish the different types of data processing and usage that take place.

Table 2 - Use Case Categories

Use Case Category	Description
Reference	Use cases relating to the lookup of valid descriptive data about the assets and the railway.
Responsibility	Use cases relating to ownership, responsibility and commitments to supply and use the data as agreed.
Security	Use cases relating to the need to maintain security of railway IT systems and the integrity of the RCM data handled by the architecture
System	Use cases relating to the management of the RCM data architecture and the connection of external systems to it.

The purpose of the use cases is to give guidance on where the data architecture is required and where not. All aspects of the data architecture should be present because they support one or more of the use cases.

3 Architecture principles

3.1 Introduction

Following the work to survey the literature and consult with stakeholders described in Section 2, the requirements were analysed in depth and the guiding principles for the data architecture defined and described in detail in the Architecture Principles report [4]. These principles are intended to ensure that the data architecture is stable over time, well-grounded in best-practice, able to be implemented easily and at an appropriate level of complexity and sophistication by conforming parties, and compatible not only with binding legislation and standards but also with system developments elsewhere in the rail industry and in the broader IT and asset management domains.

RCM data are shared between parties in order to support business processes which have goals such as reliability improvement, maintenance of safety and cost reduction. These processes always involve data from other sources: asset management systems, operational systems, work planning systems etc. They are informed by RCM data, but also involve data of their own which are not directly RCM-related.

The RCM data architecture described in this document covers the following scope:

- Data directly associated with RCM: sensor data in raw and processed forms; alerts and alarms; health assessments and estimates of usable life or future condition.
- Data needed to identify the assets described by or affected by the RCM data.
- External data on usage or other influences on future life or state such as weather
- Metadata describing aspects of the exchanged data such as its provenance, ownership, usage conditions, currency and dependability.

The architecture does not restrict the type of rail asset being considered (moving or fixed, point or linear), the type of data being collected (analogue /

digital, time-based, frequency-based or event-driven, alerts, alarms and health assessments), or the level of analysis (raw, processed, assessed against tolerance criteria, merged with other data). It will therefore potentially work with any current rail RCM initiative and with any reasonable future one.

The architecture is presented as a set of components which build progressively on each other to support greater levels of data and process integration. The most basic components do not demand large-scale investment in hardware or software to be brought into being, though some of the more sophisticated applications at the higher levels will need investment in shared IT infrastructure, for example to manage shared reference data and to mediate message transfers between data originators and data consumers.

The architecture Principles and Requirements will be used:

- As a set of specific requirements on data sharers which can be included in agreements between data providers and data users, which will simplify their own task of interchanging data while furthering the broader goals of industry data availability.
- As requirements on the developers of any shared infrastructure which is set up to support data interchange in an industry-standard way.
- As guidance on the principles to be followed in managing the architecture over the longer period to ensure that its goals are realised.

In the paragraphs below, we describe the key elements of the architecture:

- The overall vision of the architecture as a set of layered components
- Ways of structuring RCM data, processes and interactions
- The Conceptual Data Model
- Asset Identification and Reference – Services from the Wider Industry
- Data Interchange methods
- Relationships between assets and functional views
- Metadata.

3.2 Overall vision of the data architecture

The data architecture will be a layered one, where the layers correspond to increasing levels of data integration and thus increasing opportunities for business process improvement. The layers are shown diagrammatically in Figure 6 and described in Table 3. The lowest level of integration is at the bottom of the figure.

Figure 6 - Data architecture overview

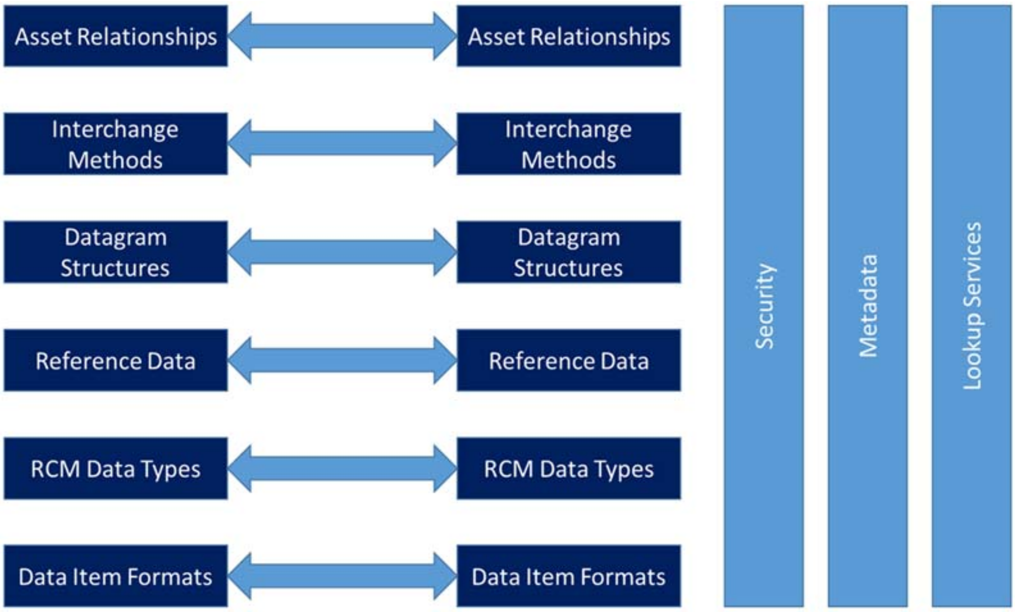


Table 3 - Proposed data architecture layers

Data architecture Layer	Standardisation level	Integration capabilities; dependencies; limitations
1 Standard Data Item Formats	Basic: common understanding of standard quantities.	Simpler coding and more reliable interpretation of data from other parties. No standardisation of asset or RCM data, or of file formats or interchange mechanisms.
2 RCM Data Types	Structured: RCM data at all levels of the ISO 13374 stack	Standard representations for all types of alert / alarm, sensor, health, prognostic data. Enables RCM services to be set up with clear process/data interfaces.
3 Standard Reference Asset Nomenclature	Asset-level: data from different sources about the same asset can be merged easily.	Significant step forward enabling a number of new processes and new data integrations. Will need support in terms of referencing services / asset ID repositories and lookup functions (such as AVI or location referencing). No standardisation of file formats or interchange mechanisms.
4 Datagram Structures	Shared understanding of formats for data interchange: files, messages, other.	Standard datagram formats for all kinds of RCM data. Significant level of integration possible at this level, at least for asset health and history; and the ability to drill into RCM data from multiple sources in a standard way.

Table 3 - Proposed data architecture layers

Data architecture Layer	Standardisation level	Integration capabilities; dependencies; limitations
5 Interchange Methods and Transaction Protocols	Shared mechanisms for requesting and transmitting data.	The ability to request and receive data in real time and by a number of asynchronous methods. This enables integrated business processes involving RCM data to be set up. The more real-time aspects require a message-based infrastructure such as an ESB to mediate the interactions.
6 Asset Relationships	Standard methods for expressing and enquiring of complex asset structures.	Enables roll-up of single assets into groupings (such as point motors -> interlockings -> route sections; or vehicles -> multiple units -> formations). Enables drill-down into assets' components (such as vehicle -> bogie -> wheelset -> bearing). This enables higher level RCM functions such as the assessment of network capability in the presence of faults or restrictions of some assets.

This layered structure maps well on to the conceptual layers set out in ISO 15926 and embodied in the MIMOSA OSA-CBM implementation.

In addition to the data integration layers described above, the data architecture considers the following themes which affect all data interchanges:

- Security: who has access to the data and to connected systems; how data and system integrity is guaranteed
- Reference: getting access to accurate reference and master data from external systems.
- Metadata: data about the data: ownership, responsibility, licensing, quality.
- Extensibility: the ability to augment the data items defined in the data architecture with user-specific ones.

- Implementation: what needs to be written, built and supported for the architecture to work.
- Governance: how is the architecture managed and compliance with it encouraged, regulated or mandated.

In order to be generally applicable, the data architecture must have an abstract conceptual data model that can be used for any type of asset, any type of measurement or data and any type of RCM-related application.

However, in order to be useful to actual data interchangers, it must also be rooted in the types of asset, standard representations and data formats that they understand and use regularly.

Figure 7 - Hierarchy of data models

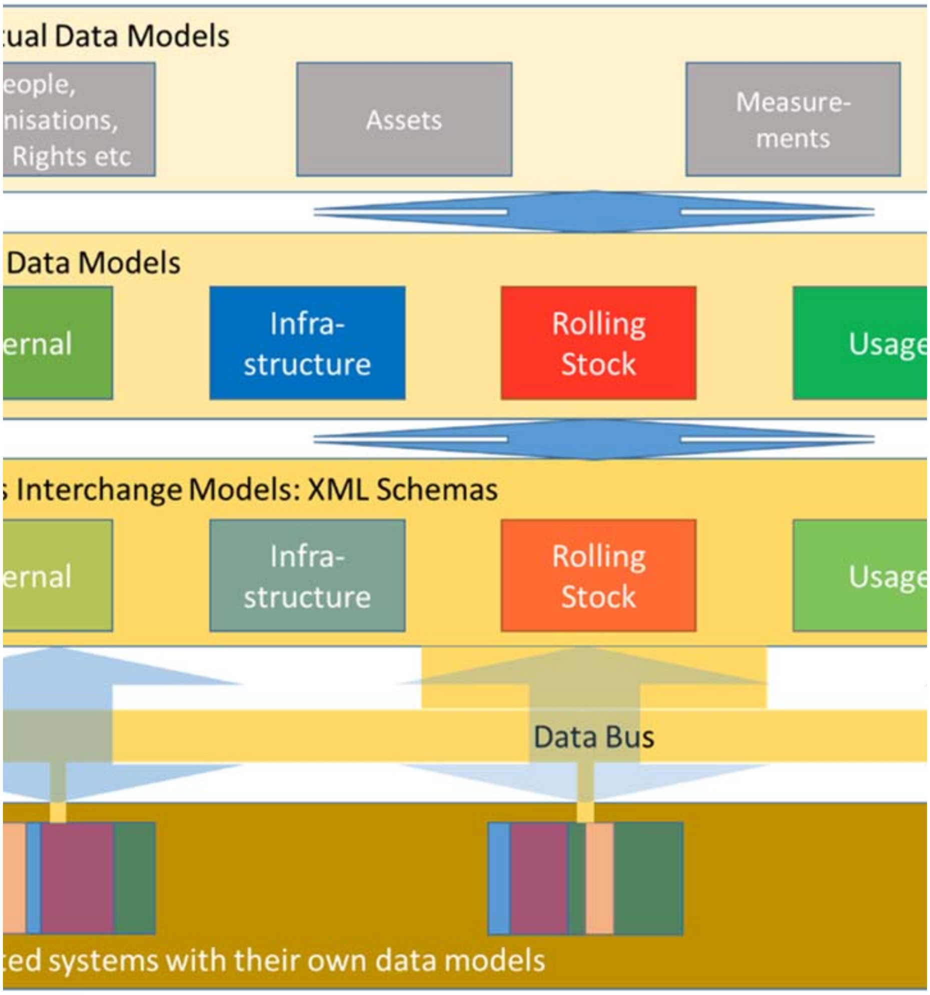


Figure 7 shows the hierarchy of data models in the RCM data architecture. It also shows an important possible role of a Rail Ontology in managing the mappings between the layers.

The four layers of data model in this figure are, from the top down:

- Conceptual data models. These deal in the abstract entities associated with asset remote condition monitoring – assets and measurements about them, as well as more general concepts required to manage data about places, people, rights and roles. At this level there is nothing specific about the rail industry in the data model. Key sources for these models are ISO standards and global standard data models and ontologies such as the Dublin Core.

- Domain data models. These represent the railway and other signification application areas such as RCM in a way which is independent of any specific IT system, business process or data interchange requirement. They are informed by existing shared standards such as RailML and RailTopoModel, TSIs and Railway Group Standards.
- Data bus interchange models. These models reflect data structures appropriate for data interchange. They are geared to clear and concise expression in terms of XML Schemas for message types, which can themselves be expressed physically in different standard formats such as .csv or JSON. The key driver for these is that they are easy to use: they contain familiar concepts in a clear structure, supporting the types of data interchange required by users. These data models form the "lingua franca" for a data-bus model of data interchange through data services.
- Connected systems' data models. These are the data models used by actual IT systems which will provide or consume RCM data. They therefore sit outside the scope of the data architecture, but their design should inform that of the data bus interchange and higher models. Each such data model will be an amalgam of elements of infrastructure, rolling stock, usage and RCM data, using its own concepts and data formats.

3.2.1 Mapping between data models

There is a conceptual linkage between the layers. Each lower layer is a 'realisation' of the layer above; each higher layer is an 'abstraction' of the layer below. The entire structure is built by a combination of top-down processes, in which a starting abstract model is applied to a particular domain; and bottom-up processes, in which real data models are compared and shared overlapping concepts identified and resolved.

The choice of abstract models at the top layer is based on previous work in this domain and others, where shared concepts and relationships have been identified, applied and agreed.

The shape of the models in the lower layers is based on understanding of the structure of the rail industry, its business processes and the need for data interchange to support them.

Each system which will provide or use data from the data architecture will need its sponsor to carry out the mapping between its own data model and that of the data bus interchange model. The code elements which do this mapping are known as Data Adapters.

In the example data models and datagrams presented in this document, which represent aspects of the top three layers described above, this mapping has been done 'manually' by the project team based on their understanding of the rail domain and of the data that is transferred. Any data interchange project will have the data elements it needs represented at the data bus Interchange level. The bottom-up mapping process and reconciliation with higher-level models will therefore need to be repeated each time a new type of data or interchange is encountered. There is thus likely to be considerable benefit to systematising and automating the process as far as possible.

The data architecture comprises shared data and shared IT elements. Each of the integration levels requires its own such elements, starting from simple requirements at the 'Data Item Formats' integration level, going up to a considerably higher requirement at the Asset Relationships level.

Figure 8 - Architecture data and IT components

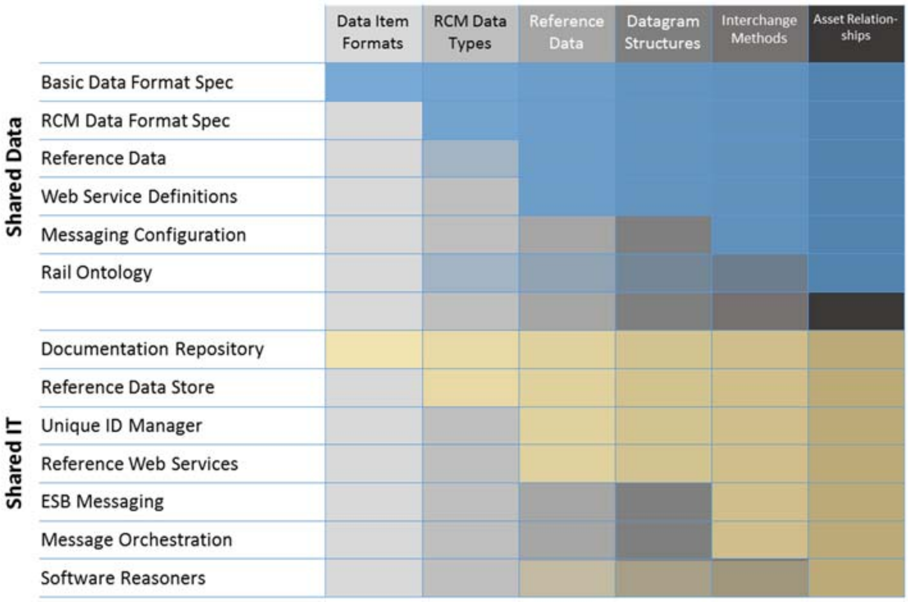


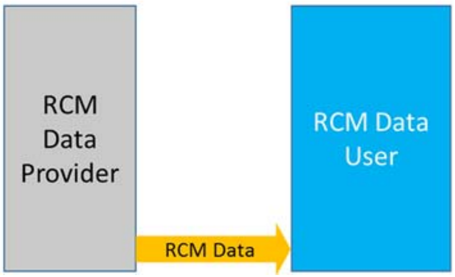
Figure 8 shows the main data and IT elements required for each of the six integration levels. The heavily shaded blue and yellow squares indicate where the element is necessary to deliver an integration level; lightly shared squares indicate where the element would add functionality but is not a pre-requisite.

The main message from this figure is that data integration efforts can be supported at the lower levels of integration with relatively little central investment in IT and relatively little top-down standardisation / data modelling work. Features can be added progressively as the demand for integrated data and standardisation increases, based on successful implementation of the simpler elements.

The application of the architecture can be seen in the following sequence of diagrams which show how its elements can contribute to a gradual increase in the level of data and process integration in the industry.

3.2.2 Unstructured interaction

Figure 9 - RCM data provider and data user; unstructured interaction



- Figure 9 shows a basic data flow between a provider of RCM data and a user. At this stage there is no structuring of the interchanges between them.

3.2.3 Use of standard data types

Figure 10 - RCM data provider and data user; structured interaction

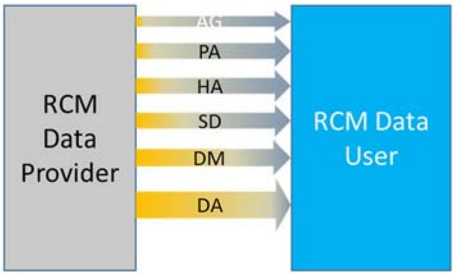


Figure 10 shows the result of adoption of the lower levels of the data architecture, Standard Data Types and RCM Data Types. Each data flow can be positioned at one of the levels of the ISO 13374 stack; and the data interchange can be carried out with standard and open data formats. No central IT infrastructure will be needed to support this level of integration. Both the Data Provider and the Data User will derive benefit from the use of the architecture in terms of ease of data processing, the ability to standardise across other similar interactions they may have with others, and the clarification of the location of intellectual property and ownership of data that will occur.

3.2.4 Data bus: standard requests and responses

Figure 11 - Data bus with shared data model

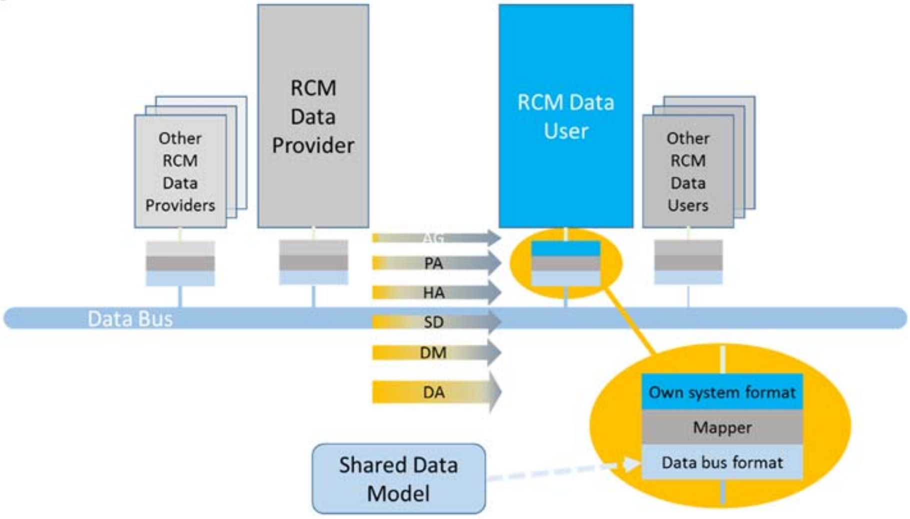


Figure 11 shows similar types of interaction, now mediated by a data bus. This corresponds to levels 3 and 4 of the data architecture described in Table 3. The key element of the data bus is the Shared Data Model which means that all data interactions between Providers and Users are done with a common view of the structure of data items and common ways of representing elements of the railway network. Each data contributor and data user must build a Data Adapter to convert data from its own internal format to data bus format and/or back again. The common data format is used to perform the data interchanges, rather than a bilaterally-agreed format.

Data interchanges between different providers and the same user; or between a provider and all its users, will now all be done using the same set of data formats as each other. This opens up the possibilities for specific RCM service providers such as optimisers, degradation calculators, data smoothers, data warehouses to enter the market.

The Shared Data Model needs to be maintained centrally, which generates a governance requirement on an industry body. Guidance needs to be given to adopters of the architecture on the development and management of their adapters.

At this stage there is no fundamental need to provide an IT infrastructure to carry the data, existing channels will work perfectly well. However, the tools and features available on standard Enterprise Service Bus products, for example to manage user access and security and to log and audit transactions, may be beneficial.

3.2.5 Standard reference queries/responses

Figure 12 - Request/response reference data lookup

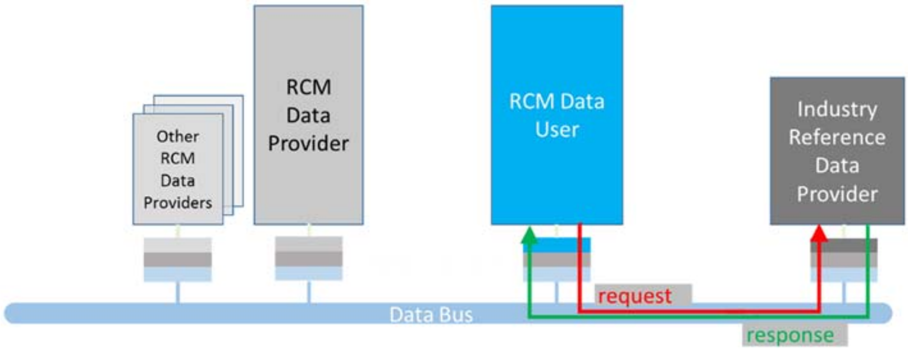


Figure 12 shows an enhancement built on the data bus: the ability to query standard industry reference data sources using standardised request and response formats. This is an aspect of level 3 of the data architecture described in Table 3, with messaging facilities made available by level 5.

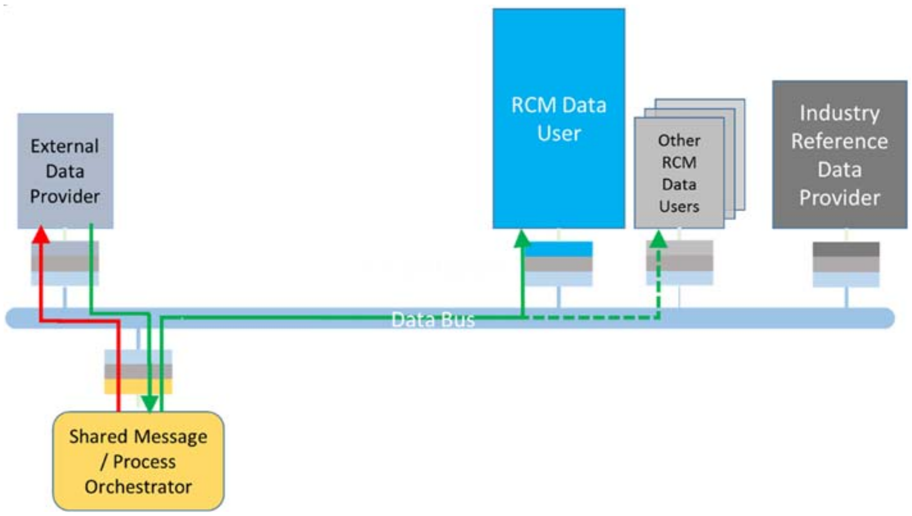
The Reference Data Providers would be facades built on to existing industry IT systems using the standard Data Adapter mechanisms already described. The queries and responses would use standard Web Service protocols. Typical queries might be (of a rolling stock register system): list all the component parts of vehicle with EVN 937034512342; (of a network model): give the track location nearest to Lat 51.2315N, Long 1.5432W; (of an AVI service): give the full consist of the train that passed the point at mileage 23:46 on ELR ECM1 track 2100 heading northbound at 12:22 on 5 May 2014, for which one RFID tag said 937034512342L.

The owner or sponsor of each reference system would need to provide the Web Services. The exact format of the data returned by the service would depend on this sponsor, but it should conform to the Shared Data Model defined by this data architecture.

The benefits to the industry of this type of reference data service will be great and will not be limited to the demands of RCM data interchange. They will result in a massive improvement in data currency and quality across the rail network and will enable the sharing of reliable data on a much greater basis than now.

3.2.6 Message-based data distribution

Figure 13 - Message-based data integration



External Data Provider
RCM Data User
Other RCM Data Users
Industry Reference Data Provider
Data Bus
Shared Message / Process Orchestrator

Figure 13 shows one example of the further possibilities offered by a message-oriented data bus using facilities at level 5 of the data architecture (See Table 3): the use of a message manager function to handle the acquisition of data from an external source (such as a weather service) and the distribution to several data consumers using a 'publish-and-subscribe' mechanism.

3.2.7 Complex mappings using ontologies

Figure 14 - Ontology to facilitate complex conceptual mapping

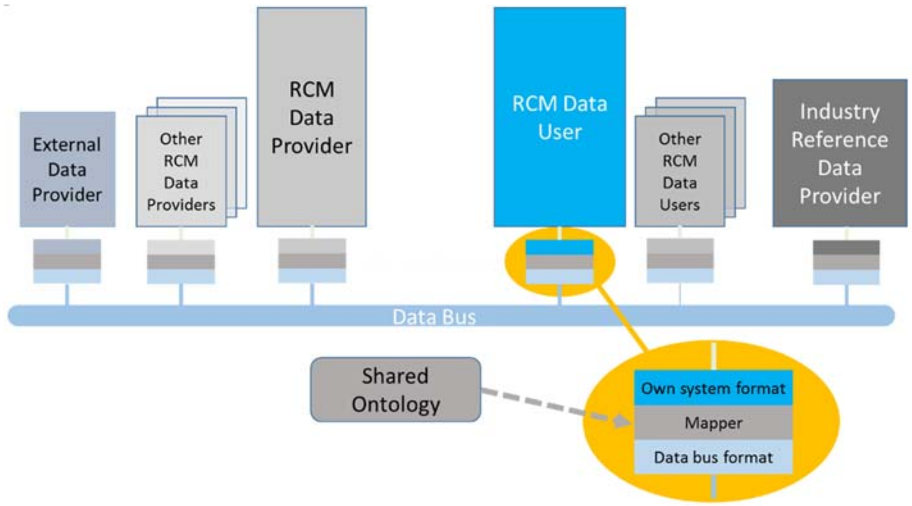


Figure 14 shows one aspect of data integration that will be facilitated by the use of a Shared Ontology (level 6 of the data architecture described in Table 3): the connection of users or providers that have a different conceptual view of the structure of the rail network, trains or data items than the Shared Data Model of the data bus has. The ontology will enable the conceptual views to be mapped to each other in a maintainable and extensible way that does not require hard-coding.

The Shared Ontology will need to be managed centrally. It is an extension of the Shared Data Model that adds semantics to the model. Other aspects of the level 6 architecture would be the extension of Data Adapters to enable data querying using SPARQL and result rendering using RDF.

There are other important applications of the Shared Ontology. A significant one will be the ability to query and understand the relationships between groups of assets or of assets from different functional domains. This would, for example, enable the automation of processes such as calculating the impact on passenger journeys of the unavailability of an infrastructure component such as a point motor.

3.3 Organising the rail RCM world

3.3.1 Introduction

The data architecture needs to represent a sound framework into which future RCM data sharing projects can be fitted. This framework needs to reflect the types of RCM data to be handled, the industry processes for which it is used and the levels of system integration they demand, and the surrounding technical and data environment.

Possible levels and scales of data / process integration have been discussed above in Section 3.2. In this section we consider the framework in terms of RCM process, data type and monitored rail asset type.

3.3.2 RCM data and processes: ISO 13374

The data architecture reflects the 6-layer hierarchy of RCM-related processes set out in ISO 13374 and described in Section 2.4.1. This means that data transferred using the data architecture will abide by the following principles:

- All data elements can be characterised as being the outputs of one of the ISO 13374 processing layers / blocks. Each layer / block has a characteristic output:
    - A Data Acquisition (DA) block generates a scaled digital version of sensor data from a specified sensor, in one of a number of standard formats, with timestamps and quality indicators.
    - A Data Manipulation (DM) block generates descriptors of the sensor data, for example: extracted features, conversions, integrations, filtered versions, normalisations, averages, statistical summaries, with timestamps and quality indicators.
    - A State Detection (SD) block generates indicators of the asset or sensor state, indications of proximity to, or exceedance of, threshold boundaries, identification of data or sensor anomalies, with timestamps and quality indicators.
    - A Health Assessment (HA) block generates a health grade or a diagnosed failure mode for an asset, optionally with probability assessment and justification.
    - A Prognostic Assessment (PA) block generates a projected future health grade or time to failure in a given mode for an asset, associated with a set of future load / exposure values and with an optional explanation.

- An Advisory Generation (AG) block generates a work request or operational advisory for action by external Operations / Maintenance systems.

- A block in each layer can request data from blocks at the same or lower levels.
- Each block has a defined protocol – an Application Program Interface (API) - by which data can be requested from it, and a defined set of formats in which the data can be provided.

This approach means that both existing and new IT systems can be categorised according to where they fit in this layer / block model.

Figure 15 - Systems, APIs and the ISO 13374 layers

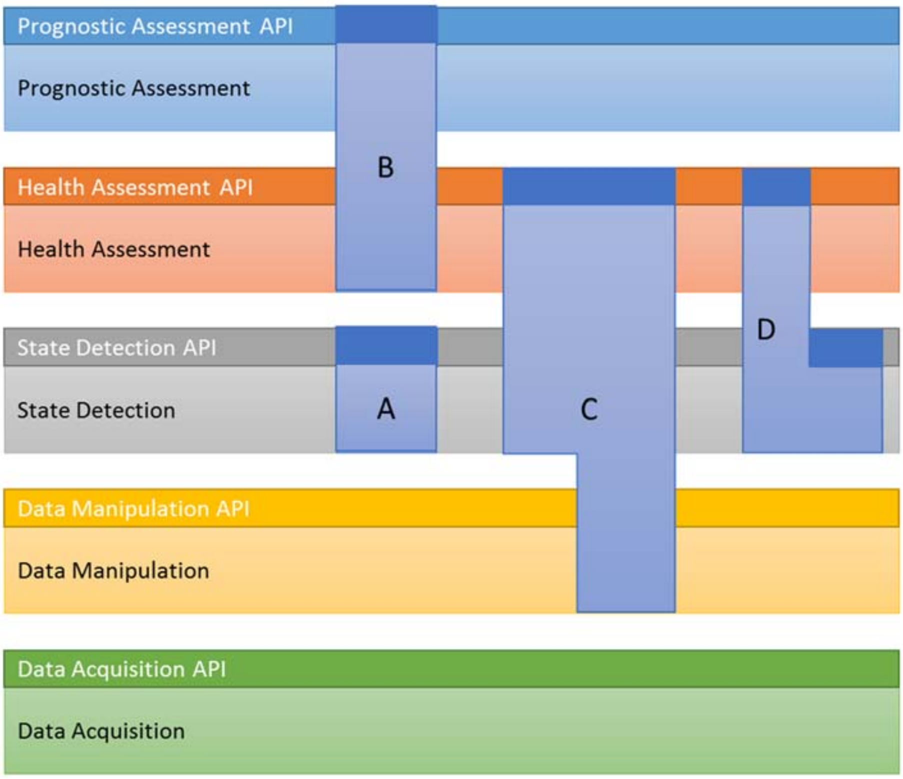


Figure 15 (taken from ISO 13374-2) shows how different IT systems in the RCM domain can carry out functions and offer data about different ISO 13374 processing layers:

- System A only does State Detection functions and exposes just a State Detection API. This may, for example, represent a specialist algorithm which carries out a sophisticated composite analysis of different sensors to come up with an overall asset state.
- System B operates at both the Health Assessment and Prognostic Assessment levels but only outputs Prognostic Assessment data. This may, for example, be a forecasting module which calculates current asset health (though doesn't output it) and generates forecasts of time to live based on expected future loadings. System C has a similar structure, though operating over the DM, SD and HA levels and emitting Health Assessment data only.
- System D carries out both State Detection and Health Assessment duties and can output data from both these levels.

3.3.3 Types of RCM data: MIMOSA

The data architecture needs to support all the types of data collected by all the RCM activities, current and foreseen. There are many different types of sensor and many different formats and types of data that they gather and transmit.

MIMOSA OSA-CBM [15] defines a set of standard data types appropriate to the outputs of each of the ISO 13374 processing levels. These are listed below. From the survey of rail RCM systems carried out for this project, it is clear that these types enable all current and expected rail RCM data to be expressed in a compliant way. The data architecture therefore requires RCM data to be expressed in these formats. It defines standard representations of them in XML.

3.3.3.1 Data Acquisition

For the Data Acquisition layer of ISO 13374 (sensor data):

Table 4 - MIMOSA Data Types - DA Layer

Data Type	Format	Typical Usage
Integer	A single integer	Count of occurrences e.g. revolutions
String	Alphanumeric string – no length restriction	Description e.g. 'Overheating'
Real	Double-precision floating-point number	Temperature in degrees Celsius
Boolean	True or False	Status = OK
Vector	A sequenced list of Real values	A set of temperature readings from a group of sensors
Waveform	A regularly-spaced list of Real values.	A set of current readings sampled at 100Hz for a point motor for a short period of time such as a point activation.
Data Sequence	A set of x, y pairs of Real values	A set of records of point operation, showing the start time and duration of each activation.
BLOB	A binary file identified by a MIME type	A photograph of an anomaly on an OLE neutral section; the sound file associated with a train passing over a microphone array.

3.3.3.2 Data Manipulation

For the Data Manipulation layer of ISO 13374 (processed data):

Integer, Real, Boolean, Vector, Waveform, Data Sequence and Blob as for DA, plus:

Table 5 - MIMOSA Data Types - DM Layer

Data Type	Format	Typical Usage
Ampl	Amplitude and Phase as Real values	Power Factor
Complex Waveform	List of Real and Imaginary components as floating point (Real) values	Electrical voltage and phase over time
Constant Percentage Bandwidth Spectrum	List of Real values and band start / width parameters	Noise spectrum
Real Frequency Spectrum	List of Real values by standard increment of frequency	Frequency spectrum such as that generated by rotating equipment
Complex Frequency Spectrum	List of Real and Imaginary values by standard increment of frequency	Phase response of acoustic equipment

3.3.3.3 State Detection

For the State Detection layer of ISO 13374 (state / alerts / alarms):

Integer, Real, Boolean as for DA, plus:

Table 6 - MIMOSA Data Types - SD Layer

Data Type	Format	Typical Usage
Enumeration	Value from a list of valid values	Alert status: Normal / Alert / Alarm / Out of Service / Test
Enumerated Set	List of enumerated values	As above, but for multiple statuses
Integer Test	Integer Value and Enumeration	Measured value and the matching looked-up state: 16 clicks – OK

Table 6 - MIMOSA Data Types - SD Layer

Data Type	Format	Typical Usage
Real Test	Real Value and Enumeration	Measured value and the matching looked-up state: 27.3V – Alert: Low

3.3.3.4 Health Assessment

For the Health Assessment layer of ISO 13374:

Table 7 - MIMOSA Data Types - HA Layer

Data Type	Format	Typical Usage
Health Assessment	Health level and type, expressed as Real (0-1) or Integer (0-100)	0 = terrible health -> 100 = perfect health 0.0 = terrible health -> 1.0 = perfect health

Associated with the data items at this level can be additional diagnosis information which describes the reasoning for the Health Assessment.

3.3.3.5 Prognostic Assessment

For the Prognostic Assessment layer of ISO 13374:

Table 8 - MIMOSA Data Types - PA Layer

Data Type	Format	Typical Usage
Remaining Useful Life	Estimate of life, confidence level	Time till asset will fail
Remaining Useful Life Distribution	Set of estimates of life and a cumulative probability distribution with error bars	Graph showing time till likelihood of failure exceeds a threshold
Future Health	Estimated health level at a specified future time, confidence level	Forecast health at a specific future time
Future Health Trend	Set of estimates of future health level for a sequence of future times, with error bars	Assessing when asset health will fall below a threshold

Associated with the Prognostic Assessment data can be additional diagnosis data which explains the assumptions behind the forecasts.

3.3.3.6 Advisory Generation

For the Advisory Generation layer of ISO 13374:

Table 9 - MIMOSA Data Types - AG Layer

Data Type	Format	Typical Usage
Recommendation	Priority, chosen from a list of levels, with explanatory remarks	Prioritisation of work on a large group of assets
Request for Work	Maintenance or Work Package Definition	Automatic request to work scheduling system to perform maintenance or correction on an asset.

The MIMOSA standard also allows other data formats to be used if required, though it does not encourage this practice.

For each of the data types listed, MIMOSA also defines a set of standard metadata which define the Engineering Units of the data being represented, if applicable, a time stamp and any configuration data that are relevant. These provide information to enable users of the data to interpret it correctly.

3.3.4 Types of rail asset

The data architecture could be used for any type of asset on the rail network for which Remote Condition Monitoring may be carried out. It does not therefore build in any restriction as to the exact type of asset for which RCM data could be gathered. For the sake of consistency of representation and to simplify common usage scenarios, some specific asset types are defined.

In considering the maintenance and condition of rail assets, it is important to distinguish between the designed object (such as outer left-hand wheel bearing of the front axle of the front bogie of a rail vehicle) and the specific item fulfilling that role at a point in time (SKF bearing, type XYZ/1, serial number 3022315, fitted 23/01/2012). The designed object persists for the life of the whole ensemble (vehicle in this case); the specific item may be changed over time, may be maintained or overhauled, and may be moved to fulfil the role on a different designed object.

MIMOSA OSA-CBM uses the term Segment for the designed object, and Asset for the specific item fulfilling its role at a point in time. In the words of MIMOSA:

'An object is an Asset if it meets one of these criteria:

- *Could be depreciated in a financial system*
- *Could be tracked by serial number*
- *Could be transferred/sold and utilized/installed at a different Segment possibly associated with another Site at another Enterprise'.*

In most current RCM systems, RCM data are gathered for a Segment (for an identified location such as point motor on switch SN210, or outer left-hand bearing on the front axle of the front bogie on vehicle 27154). The identification of the specific asset, serial numbered item, is not known at the time the measurement is gathered. However, making this connection is an important enabling factor for the higher-level RCM functions such as prognostic assessment and advisory generation: it is impossible to track the degradation of an asset over time in an RCM system if the system is unaware that the asset has been replaced during the time period.

The MIMOSA nomenclature is at odds with common nomenclature in the rail industry, where typically a MIMOSA 'Segment' would be referred to as an Asset; and a MIMOSA 'Asset' would be referred to as a Component. To avoid confusion in the data architecture documentation, the terms *mimSegment* and *mimAsset* will be used if the MIMOSA interpretation is intended.

3.3.5 Conceptual Rail Network Model

The rail network can be viewed at a number of different levels of abstraction, from the lowest physical level of sections of track, switches and crossings and buffer stops, through intermediate layers to a high-level corridor view. Each railway function tends to have its own view of the network and to express data about the network at that level. This would include RCM-driven condition, usage and impact information.

There are several schemes for representing the rail network in common use, each with its own concepts of nodal points for locations and links for connections between locations. Various attempts have been made to harmonise them. The data architecture will use the UIC's RailTopoModel (see Section 2.4.3 where RailTopoModel is discussed in connection with its application in RailML) which can deal with all the current UK representations of topology and also enables application-specific ones to be defined.

RailTopoModel supports multiple methods of identifying linear position on the route (such as the ELR / Track ID / Miles:Chains method, or a metres-fromdatum method); as well as different geographic location schemes for the same point (such as WGS-84 Latitude / Longitude, or OS grid reference).

It also supports different location naming schemes for locations at different levels of abstraction (such as STANOX / TIPLOC / NALCO commonly used in the UK industry).

The key element of a topology is that it represents the connectivity or routing possibilities between sections of route at each level. This means that it can be used to analyse and represent, for example, the number of distinct routes through an interlocking depending on whether a particular switch is operational or locked in its normal or reverse state.

Crucially, RailTopoModel defines a clear algorithm for mapping between different layered views of the network. This enables automated methods, particularly those based on ontological principles, to be able to apply RCM data gathered at one level (say at a specific switch / crossing) and work out its impact at a route or corridor level. This will be vital for higher-level RCM-driven applications which need to optimise maintenance or repair interventions in terms of impact on passenger journeys, for example.

3.3.6 Fixed linear elements

The data architecture defines[1] a standard method of referring to the location of an extended linear asset such as a section of track, foundation, cutting, tunnel or OLE which can be identified by a start and end track location and, optionally, a track ID. Any such asset can be represented.

From a data architecture point of view, the key characteristic of this type of asset is that the components making it up may have a complex relationship with the asset itself. For example, a 200m section of track between two switches may have several different pieces of rail in it, all of different types, ages and conditions.

[1] In draft: the actual representation will depend on Network Rail's ORBIS programme

3.3.7 Fixed point elements

The data architecture defines[2] a standard method of referring to the location of assets which occupy a single location on the rail network, such as a bridge, switch/crossing, or signal cabinet. Any such asset can be represented.

The asset location will be defined in terms of a track location (in one of the recognised formats) and an optional transverse offset.

Any such asset may be composed of other assets. The data architecture has a general method of representing such structures which can cope with any number of levels of nesting. It also has a mechanism for defining standard structures which are often found in the UK railway.

3.3.8 Moving elements: vehicles and their components

The central concept in the data architecture's representation of rolling stock is the Vehicle. There is a standard nomenclature for rail vehicles which the architecture reflects. There is also a standard set of information about each vehicle mandated or recommended by TSIs, RGSs and RISs which reflects a standardised view of a vehicle's component hierarchy. The data architecture reflects this structure[3].

The data architecture also enables a more general multi-layer structure of a rail vehicle to be defined, enabling data to be associated with very low-level components if required while preserving the key understanding of which vehicle the data belongs to.

3.3.9 Moving elements: formations and physical trains

Vehicles may be associated with other vehicles in quasi-permanent formations such as Multiple Units or rakes of coaches or train sets; and these formations may be temporarily associated with each other through coupling to form the consist of an operating train. In such formations, the order of formations, the orientation of a formation and the orientation of each vehicle within the formation are of key importance since they enable the data from trackside sensors to be associated with the correct component of the vehicle (such as a

[2] In draft; the actual representation will depend on Network Rail's ORBIS programme

[3] In draft; the actual way vehicles and their data are represented may depend on the replacement for the Rolling Stock Library.

wheel bearing). The data architecture supports the representation of vehicle orientation for this purpose.

3.3.10 Operational trains

The UK railway has several different concepts for an operating train, depending on whether an operational, engineering or commercial view is being taken. Decisions on the impact of RCM data on the railway may need to represent a train and its condition and expected future performance to be expressed in any of these ways.

Associated with a train or elements of its formation may be future work plans (diagrams) which may impact or be impacted by decisions

The data architecture supports all the standard representations of operational trains[4].

3.4 Data standards

The data architecture defines standard ways of representing common data items to reduce the (surprisingly common) inconsistencies and uncertainties that occur when interpreting data from other sources. Where available, ISO standards are used; otherwise, existing UK rail industry standards are used.

3.4.1 Basic data items

The data architecture requires standard formats for basic data items:

- Dates: must use the ISO 8601 format. Legal forms are YYYYMMDD or YYYY-MM-DD, with truncated forms YYYY or YYYY-MM also valid.
- Times: must use the ISO 8601 format [18]. Legal forms are HH:MM:SS or HHMMSS, with an optional partial second added, such as HH:MM:SS.dddd, which can have arbitrary precision. Times must be in UTC (indicated by Z at the end 'Zulu time') or have an explicit time offset from UTC specified.
- Geographic locations: must use the ISO 6709 format. The reference system must be specified: it will typically be WGS-84 (for latitude / longitude) (EPSG [19] 14) or OS grid reference (EPSG 27700), but other legal ones listed in EPSG are also allowed.

[4] In draft; the actual representations may well change depending on Network Rail's LINX-TM programme and similar developments.

3.4.2 Railway data items

The data architecture requires that standard railway items are referenced using well-known codings and formats. Some of these are provisional since new system developments in rail infrastructure, rolling stock and operations areas may result in new standards emerging which the data architecture should respect.

- Track positions: should reflect Network Rail standards, if present, or can be in the form ELR/Mileage/Track ID. Transverse offsets must be in metres to the right of the rail bed centre line in the direction of increasing mileage.
- Locations must use one of the coding schemes appropriate to the topological network layer being considered. Valid location codings are STANOX, TIPLOC, NALCO and CRS code.
- Vehicles must be identified by their European Vehicle Number. Other common identifiers (such as class / number in the form 43 110) may also be shown.
- Multiple Units must be identified by class designator and unit number, in the form 395 001 (with space).
- Operational trains may be identified by Train Reporting Number, ITPS UID or Retail Service ID as appropriate. Train Reporting Number must be augmented with starting location and/or departure time to guarantee uniqueness.

3.4.3 Unique identifiers

The data architecture must enable every entity described, whether representing a rail asset, a sensor, an organisation or person, an IT system or any other quasi-permanent thing, to have a unique identifier which is a UUID as defined in RFC4122 [20]. This is a globally-unique number which can easily be issued by suitable software.

A UUID is a 128-bit integer usually represented in text in the form 'f81d4fae-7dec-11d0-a765-00a0c91e6bf6'.

The data architecture may be required to maintain a register of UUIDs and the common entity references to which they refer, to support the mapping of item references from one scheme to another. The MIMOSA framework offers one way to do this, in the form of its Structured Digital Asset Interoperability Register [17] – this is described in Section 3.5.

3.5 Reference data

The data architecture defines some shared data stores which will be needed to support RCM data interchange beyond a simple bilateral approach. Certain data items need to be shared and to have a consistent meaning across the UK rail industry.

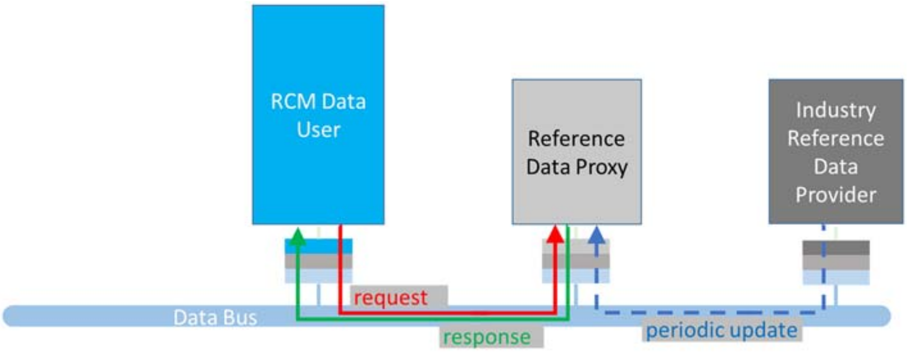
3.5.1 Stored reference data

There are a number of sets of reference data which will need to be available to data architecture users to enforce shared concepts. These will include such things as health statuses, alarm conditions for particular asset types, organisations and authorities and so on.

3.5.2 Asset reference data proxies

It is not envisaged that the data architecture itself will need to store information about all the rail assets and the RCM equipment attached to them: these will be made available via a lookup mechanism referring to the master systems for each type of data. This mechanism is described in Section 3.8.5. It may be desirable, however, for proxy data stores for this type of information to be managed centrally to reduce the demand on the master systems or to provide the analysis necessary to support more complex data services such as those that need an ontological approach (see Section 3.9) to satisfy them. Figure 16 shows the basic setup. The Reference Data Proxy gets a periodic refresh of the actual reference data from the industry master source; RCM data users do lookups on the proxy rather than the industry reference data source itself.

Figure 16 - Reference data proxy



3.6 Metadata

Metadata describe data, thus enabling users of the data to apply it in an appropriate way. Lack of a way to communicate the limitations, characteristics and context of RCM data is one of the main limitations to its effective cross-industry supply and use. The data architecture therefore provides some standard methods for supplying this additional information. It recognises the following specific aspects of metadata, but also provides a mechanism for data providers to enhance these with their own metadata items if required:

- Data source. The originating organisation or IT system must be defined.
- Time or data sequence. The creation date of a data item must be specified. In some contexts, exact correct creation time is not known (for example because of sensor clock drift or inaccuracy). In these cases, data items must be given a unique sequence number.
- Data owner. The legal owner of the data can be identified. This is done using the 'Dublin Core' standard representation [9].
- Data licensing or usage restrictions. Data providers may restrict the use of the data to specific parties or specific purposes; and there may be license terms associated with the data. These can be specified, again using Dublin Core methods.
- Data Units. Every data item must have its engineering unit and scaling factor specified.
- Data Quality indicators. The data architecture uses the MIMOSA four-level indication of the status of the data (OK / FAILED / UNKNOWN / NOT_USED) and confidence percentage (0 % = no confidence → 100 % = full confidence) which may be used as applicable; other data quality indicators (such as precision and standard deviation) can be added as needed.
- Configuration. Any RCM data processing element at whatever level of the ISO 13374 hierarchy will have stored setup parameters which defines how it behaves. The data architecture provides a standard method for interrogating and supplying these based on MIMOSA principles.

3.7 Data interchange formats - datagrams

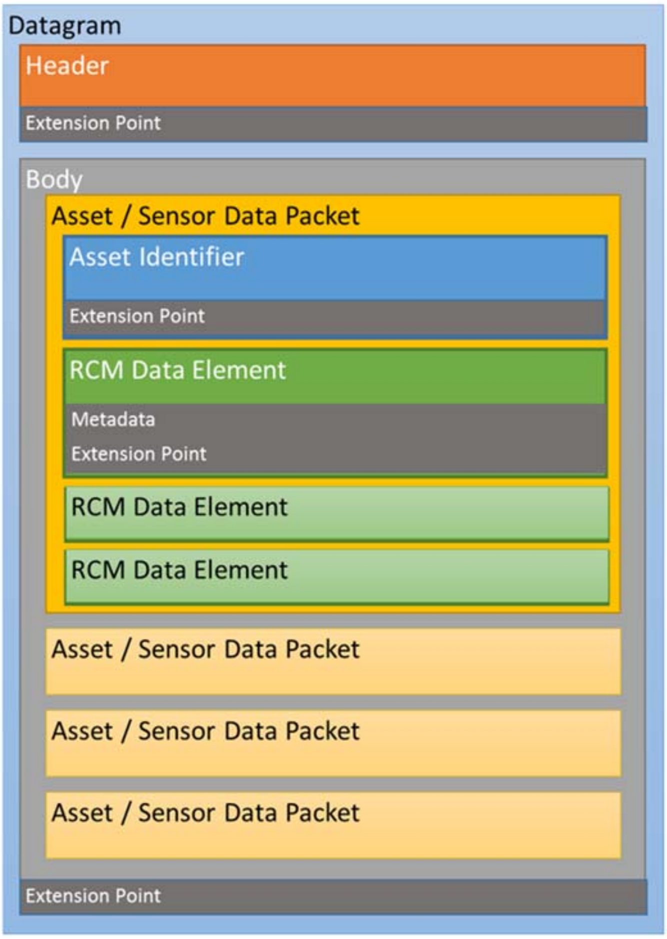
As well as specifying the formats of individual data items, the data architecture defines ways in which the data items must be assembled for transfer between parties. This is done by means of datagrams, defined in XML schemas [21]. The fact that the datagrams are defined in this way does not mean that the data themselves need to be interchanged using XML, though there are advantages to doing so which are discussed in Section 3.8. Any standard data interchange format is permissible provided that the overall structure of the data is compatible with the datagram format and the data items themselves are formatted as the data architecture requires (Section 3.4).

The number of types of legal datagrams the data architecture needs to be able to support is very large, since it is the intersection of all the asset types, RCM data types, event types and ISO 13374 processing layers. The data architecture thus specifies datagram templates standard layouts for data packets which apply to many different actual datagram layouts.

3.7.1 Datagram structure

The basic structure of a datagram is shown in Figure 15.

Figure 17 - Datagram structure



A datagram consists of a Header and a Body.

The Header contains details of the version of the architecture and data format in use, a source indication and a timestamp (mandatory), plus any other (optional) Metadata about ownership, rights, and provenance, as discussed in Section 3.5. It also has an extension point which enables additional optional data items to be added as required by the providers and users of the data. These are not validated or restricted by the data architecture.

The Body is optional, but if present will contain one or more data packets relating to an IT system, railway asset or a sensor (depending on the RCM data and the level of the ISO 13374 stack concerned). It also has an Extension Point allowing interchange-specific data to be added.

Each data packet comprises a system, asset, or sensor identifier and one or more RCM data elements relating to the system, asset, or sensor.

The System / Asset / Sensor Identifier defines which system, sensor or asset the data relates to. It will identify the asset or sensor using one of the methods described in Section 3.4.2; and may optionally also have other data items as required by the MIMOSA framework for a measurement on that asset or sensor type. It has an Extension Point which allows the parties to the data transfer to add any other data items they need.

The RCM Data Element contains data items as required by the RCM data type, which will be one of those described in Section 4.3.3. This will contain the actual sensor or asset data. Optional Metadata items can include additional useful information relating to, for example, the quality, usage, provenance, dependability of the data, as described in Section 3.5. There is also an Extension Point which enables additional data items to be included as required by parties to the data transfer (see Section 3.7.3).

3.7.2 Mandatory and optional elements of a datagram

Although a datagram has placeholders for all of the components listed above, very few of them need to be used in any given interchange. This is to provide maximum flexibility as well as compatibility as far as possible with existing data interchanges, allowing them to be developed gradually to full compliance with the data architecture.

The only mandatory element of the datagram is the Header, which must have a timestamp and a source identification. This may be encoded in a filename if required.

The body of the datagram is optional: it is perfectly valid that a datagram may contain no data, such as to report no events. Once a datagram body is included, the asset or sensor identifier is mandatory but data content is optional.

These rules mean that data can be interchanged in familiar means such as .csv files or industry standard formats with little or no modification to the file naming convention or content format, and still be compliant in a basic sense with the data architecture.

3.7.3 Extensibility

At all potentially-useful places within the standard datagram of Figure 17 are Extension Points. These are places where individual users of the data architecture may include additional data items for their own purposes. These data items will not be validated or checked by any of the data architecture's validation mechanisms. They may be present or absent as the users require.

All the data formats and transfer mechanisms specified in the data architecture will support the use of the extension points in some form. For data formats such as XML, this is straightforward, as XML parsers can be set to ignore data items they are not expecting. For other data formats such as .csv, care needs to be taken that data consumers are aware of the potential for data to be present or not in these places.

For any given usage of this extension mechanism, there may be a case for 'promoting' it to the data architecture proper, so that it becomes available to all users and subject to the same validation and quality control as other included data items.

3.8 Data interchange methods – requests, responses, files and messages

The datagram formats described in Section 3.7 define the format and structure of data items to be interchanged using the data architecture – the 'what' of data interchange. In this section the 'how' is described: methods for requesting and transferring data.

Data providers and data users will benefit from adopting the standards for data format and asset identification defined in earlier sections, even if they use project-specific methods for requesting and providing data. However, larger gains to the broader industry will be realised by the use of the standardised approach described here.

The methods described in this section support the data bus concept described in Section 3.4.2 and following and conform to now well-established Enterprise Application Integration principles for linking previously-separate IT systems.

3.8.1 The service-oriented approach

The key to the data bus approach is the concept of software services. Each connected data source IT system offers a number of such services to any other connected system. The services are invoked by means of a standard request from a connected system; and the results of the services communicated back via a response.

The software service may do any of these types of task:

- Acquire and provide data from a sensor or logger.
- Get data from an external source such as a weather forecast service.
- Carry out an operation such as send an email, update a database or create an audit trail.
- Process data included or referenced in the request and return the answer.
- Store data in a repository or retrieve data from a repository.
- The technologies of the World Wide Web have proven themselves very useful as a medium for the handling of requests and provision of responses, to the extent that 'Web Services' are now a standard mechanism for providing this type of service. Indeed, the World Wide Web itself is an excellent example of the possibilities opened up by the application of a simple request / response data protocol. The main reasons for this success are:
- The underlying protocol (hypertext transmission protocol, http) is an open standard, very simple and well-defined.
- There is no dependence on any specific hardware or software platform.
- It uses well-known and common networking technologies and is therefore reliable.
- It is scalable to high data volumes and usage rates.

3.8.2 Requests and responses

Each interaction of any data user with the data architecture will comprise a series of requests to data providers and a response to each from each provider. The request will take the form of an http request to a specific URI which will define the service required and the parameters or input data required by the service; the response will be an http message containing result data, result status or error message and possibly some additional data items indicating other related services now available.

The request will typically include details of the IT system, asset, sensor or logger from which the data are required, the time band for which data should be provided, other restrictions such as data type, sensor type or location, and stipulations on how the results should be returned.

Result data may be returned in the body of the response or (more likely for large returned data sets) as a URI link to the location of the data. Elements of the response will indicate the format of the returned data.

Linked result data may be stored in a different network location than where the service itself is hosted. This may particularly be true for high-volume data such as streaming video or audio, which are likely to be made available on specialist servers.

In some cases where processing the request may take some time, the initial response will be an acceptance of the request, typically containing a link to the final results or to another status service. The requestor will query this link to see if the results are ready; if so, they will be returned; if not, a 'pending' response will be sent.

All requests and responses using the data architecture must use the data formats and concepts described in Sections 3.3 to 3.7. It will be the task of the data requestor to make sure that requests are formatted correctly; and of the data provider to do the same for the results.

3.8.3 Request and response style

There are several competing methods for implementing a service-based API using WWW technologies. They differ in the approach they take to defining the format and content of the requests and responses and have different characteristics which suit them to different types of application.

The two main approaches appropriate to the cross-industry RCM data architecture are 'Message API' (of which an implementation based on the Simple Object Access Protocol (SOAP) would be typical) and 'Resource API' (sometimes, incorrectly, called 'REST-based'). The key differences between the two are outlined in Table 10.

Table 10 - Web Service API Approaches

Consideration	Note	Message API	Resource API
Basic operation	How does the service work?	A request for the service (name + parameters, including asset / sensor referred to) is encoded as an XML datagram and sent to a URI. The service returns another XML datagram which is unpacked by the user.	A request is sent to a URI which identifies the 'thing' about which data is requested (e.g. an asset or a sensor). HTTP methods such as GET or PUT are used to query or update data. The service returns a web response including the data or a reference to it.

Table 10 - Web Service API Approaches

Consideration	Note	Message API	Resource API
Ease of Setup	How quickly and easily can a new service be added?	All the components of the API need to be defined fully before the service can be published and used.	The API can be built up gradually and can change during development without causing too much disruption
	How quickly / easily can a new data user be added?	Coding the call and decoding the response requires some programming effort. However, programming tools can often generate the code automatically based on the specification of the service.	Standard http queries and responses mean that data users can start simply and service is easy to understand. However, there is usually no simple automation of user-side code.
Security	How can transactions be made secure?	Standard web security protocols can be used. There are also advanced security standards such as WS-Security.	Standard web security protocols apply: https and web authentication.
Flexibility of Return Data	What formats can data be returned in?	Each service typically has one data return format, usually XML.	Data return types can be negotiated using standard http 'content negotiation' and can be very flexible.
Error Management	What types of error can occur and how are they described?	Each service typically returns its own error messages, which may be difficult for end users to understand.	Standard http status messages such as 200 OK, 202 Accepted, 404 not found or 503 Service Unavailable.
Discovery of Features	How easy is it for a new user to find out how to use the service?	The basics of the service can be determined by looking at the service definition data; manual documentation is required for full understanding.	Well-designed services a) describe what features are available; b) conform to http standards. These support user-driven discovery of what the service does.

Table 10 - Web Service API Approaches

Consideration	Note	Message API	Resource API
Degree of Coupling	How much does the data user need to know about the service?	Quite a lot. Messages must have the correct parameters or will fail with an error.	Relatively little. The URI encoding scheme needs to be known to find a resource; after that, repeated querying will enable the user to find out what the service does.
Work to Maintain	How difficult is it to keep up to date?	Fairly difficult. Any change to the service specification (such as a new parameter) will need all users to update their code in sync to match.	Fairly easy. Well-designed URI schemes are resilient to change, so data service can be updated without all users needing to update unless they need the new features.
Typical Implementation		SOAP-based XML	REST-based http

The Resource API approach generally suits the requirements of the data architecture better than the Message API approach, for these main reasons:

- Users of the architecture will be from many different organisations with different IT capabilities and development schedules. It is important not to force users to upgrade when changes occur to data services unless they need to.
- The same data may be used in different ways by different users using different code. It is therefore very useful for users to request the format they wish the data to be delivered in using a standard method such as http content negotiation.
- The ability for service users to discover the capabilities of the service by querying it supports the industry goal of opening the market up to new parties.

3.8.4 Files

For the purpose of the RCM data architecture, a file is a named data set containing zero or more datagrams, structured in a published format based around the Datagram Structure described in Section 3.7.1. Data can be exchanged in any recognised file formats, whether as part of data requests or data responses. Recognised formats are:

- Existing industry standard file formats. The datagram structure has been defined in a way which should make these compliant with the data architecture.

- New file formats which are compatible with the datagram structure and which can be given a recognised MIME type to identify their format. Typical such formats (in roughly descending order of desirability) are:
  - Application/xml. This format gives maximum flexibility and clarity, though it is verbose and comparatively complicated to generate. The big advantage of XML is that it can be validated by the producer and the consumer to verify that it conforms to the published schema, thus ensuring that data are of the correct format.
  - Application/json. This is a popular format for data to be consumed by end-user code such as web applications. It is more compact than XML but doesn't enable the same level of validation.
  - application/txt. This is a tab-separated text file format suitable for direct import into MS-Excel or other tabular processing packages.
  - Application/csv. This is a comma-separated values file also suitable for direct import into MS-Excel. However, it is less reliable owing to variability in the ways that data items can be shown in this format.
- Standard MIME types for multimedia files such as audio or video. Recommended standards would be the open standards such as Ogg Vorbis; common but proprietary standards such as .mp4 would also be acceptable.

A file may be attached to the response or may be provided in the form of a URI link to a file stored separately: this is particularly likely to be the case for multimedia files or other large data files.

3.8.5 Industry reference data requests

A particularly important set of requests and responses is those that will need to be provided by rail industry IT systems presently in existence or being developed, to provide data lookup services. These will be vital in ensuring that accurate and correctly-formatted data about assets, network topology and train services are available to all RCM data exchanges.

It is likely that over time an extensive catalogue of such services will be developed, particularly if the concept of providing reference data via this request and response method takes root, rather than the current methods which are typically based around periodic file dumps. However, our initial consideration of the use cases for the data architecture has resulted in a basic set of enquiries, summarised in Table 11.

Table 11 - Industry Reference Data Requests

Data Enquiry	Note	Request	Response
Network Location	What is the track location corresponding to a given GPS position? Provided by a Network Model Service	GPS position. (Optional) direction of travel (Optional) timetabled running line (Optional) requested frame of reference	Railway location in requested frame of reference (say ELR / Miles / Chains) Track ID if known or calculable. Precision.
Train Routing	What was the actual routing of a given train? Provided by a Train Operations Service	Train ID; Run Date Optional: List of known stopping locations Optional: Corresponding list of odometer readings	Full train timetable with planned and actual routings, planned and actual stop and pass times. Odometer offset and calibration correction.
Asset Composition – Drill-up	Which larger asset units are a given asset part of? Provided by an Asset Register Service or possibly a Network Topology Service	Asset ID and / or Asset UUID. Optional: date / time. For example the EVN of a rail vehicle.	Hierarchical list of the parents of this asset. For example, for a rail vehicle this would be the multiple unit or rake identifier for the vehicle's unit or rake, followed by the train consist of which this unit is a part at the given time.
Asset Composition – Drill-down	What is the component breakdown of a given asset? Provided by an Asset Register Service Asset ID and / or Asset UUID. For example, the identifier of a set of points.	Hierarchical list of the child assets of this asset. For example, for a set of points, a list of the components of the set of points, with text identifiers and UUIDs for each one.	

Table 11 - Industry Reference Data Requests

Data Enquiry	Note	Request	Response
Vehicle Identification by EVN	What is the train consist of a train passing a trackside location at a given time, where some EVNs are known? Provided by a Train Identification Service	Track Location Date + Time List of one or more EVNs – need not be complete.	Full consist of train, including all vehicles, with the EVN and the orientation of each.
Vehicle Identification by Network Location	What is the train consist of a train passing a given trackside location at a given time? No EVNs known Provided by a Train Identification Service	Track Location Date + Time Optional: Track ID Optional: Direction of Travel	Full consist of train, including all vehicles, with the EVN and the orientation of each.
Train Service Identification by Network Location	What is the operational train passing a given track location at a given time? Provided by a Train Identification Service	Track Location Date + Time Optional: Track ID Optional: Direction of Travel	Full train timetable with planned and actual routings, planned and actual stop and pass times.

Table 11 - Industry Reference Data Requests

Data Enquiry	Note	Request	Response
Train Service Identification by EVN and Time	What is the operational train which includes a given EVN or list of EVNs at a given time? Provided by a Train Identification Service	Date + Time List of EVNs – need not be complete	Full train timetable with planned and actual routings, planned and actual stop and pass times.
Stock Working	What are the future tasks to be undertaken by the identified rolling stock at a given time? Provided by a Stock Dispatching Service	Unit, Rake or Locomotive number; EVN Date + Time	List of trains to be operated and activities to be undertaken by the Unit / Rake / Locomotive (implied by the EVN) later the same day, with start and end locations and times; planned stabling location at end of day.
Crew Working	What are the future tasks to be undertaken by the train crew operating the identified rolling stock / operational train at a given time? Provided by a Crew Dispatching Service	Unit, Rake or Locomotive number; or EVN; or Train ID Date + Time	List of crew diagrams being worked on the train, and for each, the future workload to the end of the day.

Table 11 - Industry Reference Data Requests

Data Enquiry	Note	Request	Response
Affected Trains	Which trains are affected by an issue at a given track location and time period? Provided by a Train Identification Service	Track Location Start Date / Time End Date / Time (Optional) level of detail of response	List of trains passing or stopping at the specified location in the specified time window.

3.8.6 Messages

Section 3.2.6 shows a typical usage of a message-based approach to acquiring and distributing data. It assumes the presence of an Enterprise Service Bus data management infrastructure to provide the routing facilities.

Once such infrastructure is in place, a number of useful messaging patterns become available to support distributed business processes. Although these are not strictly functions of the data architecture they are more to do with an application architecture, it is worth considering them for their value in making business use of the service-oriented approach. These patterns are all described in Enterprise Integration Patterns [22] which is the reference document for this approach to integrating business processes. The most useful are listed in Table 12.

Table 12 - Useful messaging patterns

Pattern	Application or Purpose	Value for cross-industry RCM
Request-Reply	Requesting information from a provider and getting a response. The request or response can be synchronous: the requestor waits for the response; or asynchronous: the requestor carries on with other work while the response is being prepared and deals with it when ready.	The basic mechanism for requesting and returning data

Table 12 - Useful messaging patterns

Pattern	Application or Purpose	Value for cross-industry RCM
Guaranteed Delivery	Ensuring that messages (whether requests or responses) are received even if intermediate systems fail. This is done by storing the message and automatically re-trying it till successful	Transmission of alerts or alarms that have significant impact on safety or availability, so must get through.
Event Message	Informing receivers of the occurrence of an event	Live monitoring of assets.
Claim Check	Providing a link to data stored externally rather than including it in the message itself. This improves system performance.	Communicating links to such as video of a recorded incident.
Publish-Subscribe	Distributing messages to a list of recipients who have requested to be informed.	Broadcasting alerts and alarms to parties who have expressed the need to be told.
Recipient List	Sending a message to a list of recipients which is determined by the content of the message	Communicating alerts or alarms to different parties depending on their severity or criticality of the asset involved.
Process Manager	Managing multi-stage business processes by means of messaging.	Gathering data from multiple data sources in response to an alert or alarm from one source; deciding what to do based on the overall results of this data gathering.

3.9 Asset relationships: use of ontologies

An ontology defines the meaning of concepts and entities using rules and standard attributes. It therefore operates at a level of abstraction somewhat higher than a shared data model, enabling translation between different conceptual views instead of data formats. Section 2.3.2 gives details.

One of the ways in which an ontological approach can help the cross-industry RCM Data initiative is in navigating the sometimes-complex relationships between assets of different types, different levels of detail and different purpose-based views. Examples include the relationships between components and the larger assets of which they are part; and the interactions between the physical elements of the rail infrastructure and the train operational and commercial views of the network to which they contribute.

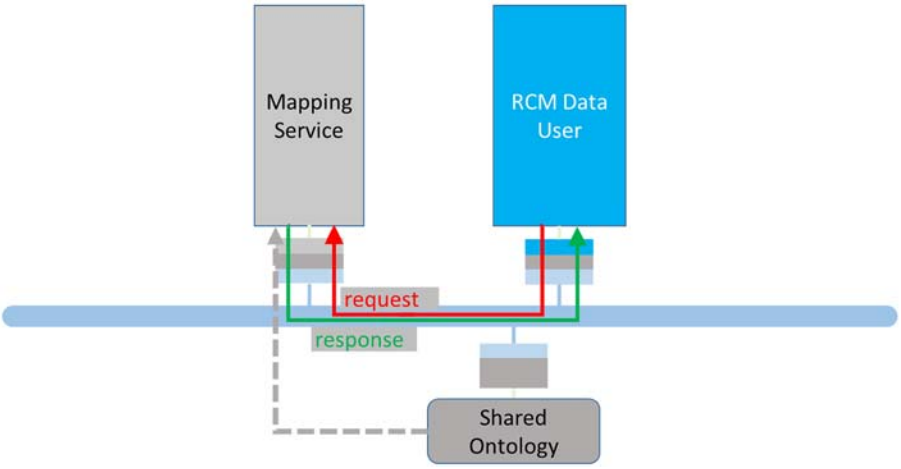
The data architecture provides these features which will support the development of an ontological approach to resolving this type of question:

- The service-oriented approach of the data bus.
- Support for the use of SPARQL end points in the design of data adapters.

3.9.1 Service-oriented approach

This enables services to be set up which would take a query in terms of one type of asset and get a response in terms of a different, mapped, type. The ontology would be used by these services to carry out the necessary mapping. (Some of the industry reference data requests listed in Section 3.8.5 are of this type and could be implemented using an ontological approach).

Figure 18 - Ontology-supported mapping service



3.9.2 SPARQL endpoints

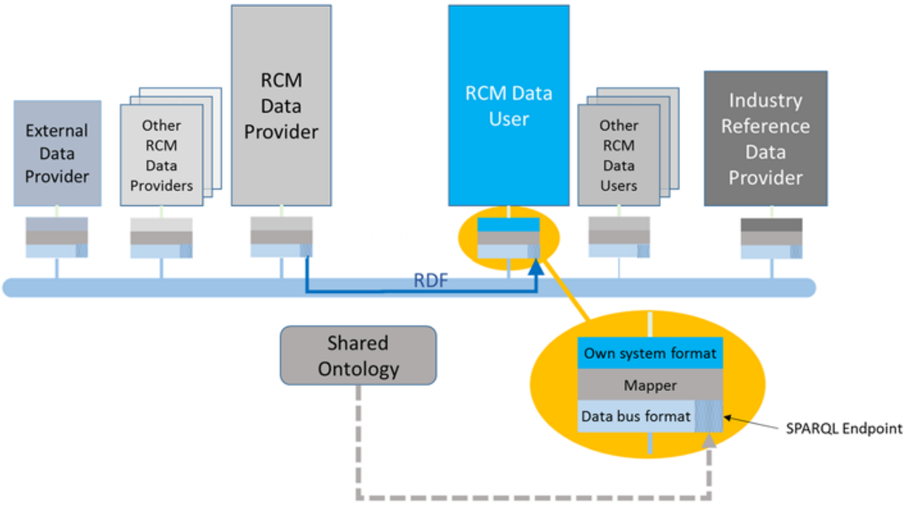
These enable RCM data sources or industry reference data sources to be queried using the SPARQL language and the results of these queries to be supplied as RDF, as described in Section 3.7.6. In particular this type of end point means that a centrally-maintained rail domain ontology could be created and made available to all data architecture users.

The RDF vocabulary used would be controlled by a shared Rail Domain Ontology which would ensure that the RDF data entries embodied concepts that could be understood by data consumers also using the same Ontology.

Figure 19 shows adapters to the data bus extended to use SPARQL and RDF. Essentially, the endpoint is an RDF interpretation of the existing data bus format adapter, informed by the concepts of the shared ontology.

The benefit of the use of SPARQL / RDF is that it supports the use of software reasoners, which can derive conclusions from relationships implied by the RDF data and the ontology for example, concluding that since a particular wheel bearing is on a particular vehicle, that the condition of that wheel bearing has an impact on the future likely reliability of the operational train that the vehicle is part of.

Figure 19 - SPARQL Endpoints and RDF



3.9.3 Semantic layer data adapters

As shown in Figure 14, a shared ontology would support the mapping between different source or target systems' internal conceptual models of the aspects of the rail network and that employed by the data bus.

4 Deployment and implementation

In this section, we consider the technical aspects of how to bring the data architecture into being, in terms of the IT infrastructure required, the demands to be placed on existing industry IT systems, and how to embed the use of the architecture in new projects with an RCM component.

This section also suggests a sequence of implementation of the elements of the architecture, driven by the following factors:

- Where immediate benefits can be gained by minimal investment.
- Funding for pilot projects and other necessary background research.
- The state of development of key surrounding industry initiatives such as ORBIS, LINX-TM, AVI and R2.
- Lead times for the development of supporting data structures and their data content.
- Lead times for the formal standardisation of the data architecture principles for the rail industry.
- Actual RCM-related projects that might benefit from application of elements of the architecture.

The data architecture will best be implemented by a combination of bottom-up and top-down initiatives, with the emphasis on bottom-up evolution through trial and use, the top-down aspects being limited to those strictly necessary to maintain long-term stability of the architecture.

4.1 IT Infrastructure requirements

Figure 6 shows the main component elements of the data architecture, with gradually increasing levels of data and process integration. These have been outlined in Section 3.2 and described in more detail in the following sections.

Each of these integration levels implies a certain amount of shared infrastructure and data: IT artefacts that need to be managed by a third party aside from the producer and user of any given RCM data stream. In this section we outline what the IT infrastructure would need to be to support each of the integration levels.

4.1.1 Overview

Table 13 lists the data architecture levels of integration, showing for each one the requirement for IT infrastructure and possible ways in which that infrastructure might be provided – whether by enhancement or application of existing systems or by the provision of new shared IT systems.

Table 13 - Infrastructure Elements for data architecture Layers

Data architecture Layer	Infrastructure Requirement	Possible Implementation
1 Standard Data Item Formats	Reference data store of asset types, data types, sensor types	Spreadsheet repository on a shared website (for example sparkrail.org)
2 RCM Data Types	Unique ID generation and referencing system for rail assets and components. Reference data store of XML schemas and lookup data types (such as engineering units and sensor types)	New shared UUID generator; possible use of existing UUID generator, for example from Intelligent Infrastructure. Spreadsheet or database repository; web service.
3 Standard Reference Asset Nomenclatur e	Industry reference data for moving and fixed assets for validation. Industry standard formats for representing assets, locations, and trains	Lookups on industry reference systems (RSL, ORBIS, ITPS) or shared proxies that provide service. Rail Industry Standard (RIS).
4 Datagram Structures	Formal shared Conceptual Data Model and Realisations. Repository of valid XML schemas; online data validator; documentation on schemas.	Shared database model. Shared file repository. Open documentation (wiki) for community of practice.

Table 13 - Infrastructure Elements for data architecture Layers

Data architecture Layer	Infrastructure Requirement	Possible Implementation
5 Interchange Methods and Transaction Protocols	Enterprise Service Bus infrastructure. Security repository with industry-wide access control.	Re-use of existing ESB infrastructure such as NR ESB or LINX-TM; or new centrally-managed ESB.
6 Asset Relationships	Shared ontology repository and access services. Enhanced reference data services provided by master system sponsors	New shared ontology server and web services. New sponsor-provided facades for industry systems; centrally-managed proxies for these systems.

More detail about these infrastructure elements is given in the following paragraphs.

4.1.2 Reference data store

To support the MIMOSA framework and the derived realisations of it for UK rail asset types, sensor types and data types, a set of reference data tables needs to be populated. These are relatively static tables, changing only when new types of asset template, sensor or data type are required by a project. They are also small tables. The whole data store can therefore be managed as a set of text files.

4.1.3 Industry reference data

The two concepts here, asset reference data and standard representations, are handled differently.

4.1.3.1 Asset reference data

Given the volume and rate of change of UK rail assets and components and the fact that certain industry systems are designated as the master data source for each type of asset, it is neither desirable nor realistic for there to be a shared data store for them all as part of the data architecture. We have therefore assumed that the system sponsors for the master reference systems (such as Rolling Stock Library (RSL) or track (ORBIS, ultimately)) will provide lookup functions for assets using services similar to those discussed in Section 3.8.5.

Alternatively, there may be a case for shared proxies for some or all reference data, based on the concept discussed in Section 3.5.2. These would be centrally-managed servers providing easy-to-use standard reference lookup web services, based on periodic data extracts from the actual reference data sources.

4.1.3.2 Standard representations

Standard ways of representing UK rail notions such as track location, train ID, and vehicle structure will be defined outside of the data architecture, by the sponsors of the systems or by an industry-wide specification document such as a Rail Industry Specification.

4.1.4 Repository of unique identifiers

<note on UUID repo>

4.1.5 Shared conceptual data model and realisations

As the scope of the data architecture expands as projects use it, new realisations of the conceptual data model will be required to represent the new asset types, data types or interchange patterns required. To maintain syntactic compatibility of these realisations, they must all be based on the same conceptual data model and created using a standardised process of instantiating ('de-generalising') the model.

The conceptual data model itself therefore needs to be stored in a way that makes it suitable for this purpose. This should be held as a UML model in a commonly-used format such as Enterprise Architect. In time, though, a more standards-compliant method would be to store it as an OWL ontology; however, the tool support for this approach is still immature.

4.1.6 XML schema repository and namespace

The XML schemas are set up to realise aspects of the conceptual data model to support specific types of data exchange. They define what constitutes a legally-formatted datagram for that exchange. The great advantage of the XML schema language is that it allows any specific datagram to be validated automatically against the schema. It therefore acts as a 'gatekeeper' to support the maintenance of data quality for data transferred using the data architecture.

Each specific data interchange will likely be mediated by its own XML schema which will use and derive from the published architecture schemas. To enable this to be done in a robust way, the XML schemas must be available in a stable 'namespace': a URI which refers to the formal definition of the schema.

Associated with the formal XML Schema language will be textual comment which will be valuable for implementers of data interchanges.

For both these reasons it is important that these schemas are made easily available on robust and permanent web addresses for data integrators to use and to refer to in their code. This needs to be via a centralised website such as sparkrail.org or a website set up specifically for the purpose.

4.1.7 Enterprise service bus infrastructure

This infrastructure makes available the data bus and message-oriented approach to data interchange, with all the benefits described elsewhere in this document.

The UK rail industry already has several ESB implementations in place. Network Rail has its own ESB infrastructure, as well as a separate message-based infrastructure for the LINX-TM programme. It would be preferable to re-use and extend one of these to host the data architecture's data bus elements, though a centralised separate ESB could be set up if this is not feasible. There are many possible ways of implementing an ESB, including open-source ones that can be deployed initially on a small scale and extended as required for relatively little cost.

4.1.8 Shared security and access control

Access to the data bus needs rigorous fine-grained access and security control. It is particularly important that the connection of centralised UK railway systems to the data bus via adapters does not compromise their security in any way.

This means the implementation of a shared security system in which any user of the data architecture can be validated and authenticated and an agreed level of access to the various connected systems managed.

There may be suitable authentication services already available in the industry which can be extended for this use.

4.1.9 Shared ontology

As the data architecture is developed and the facilities it offers move higher up the ISO 13374 hierarchy, the types of function being added will tend to require the merger of more disparate data types from different connected IT systems. The mapping of concepts rather than just data items becomes more important.

The best way to manage these mappings is the use of a shared rail ontology. This needs to be made available on a standard and stable URI (such as sparkrail.org) for the use of data integrators who will then be able to reference it in their own ontologies and data schemas.

4.1.10 Shared asset relationship services

Applications using the data architecture will come to require more complex and sophisticated information about the relationships between different elements of the rail network. Whilst we have anticipated that simple asset relationship queries will be met by the reference systems themselves (via data adapters and web services on the data bus) or by centralised proxies for these, it is likely that more complex enquiries will need specialist code to meet them.

To make the development and deployment of these queries faster and more reliable, it may make sense for them to be provided centrally on dedicated web servers associated with the shared ontology and the reference data proxies rather than with the reference data systems themselves.

4.1.11 Other infrastructure elements

There are other elements of IT infrastructure which may justify central provision and management rather than being left to existing industry parties to provide. These may include:

- An open documentation, user community site. This would provide a repository for best practice, information sharing and support.
- Large-scale data stores, such as for bulk application logs, video and audio. These would support the storage of these 'big data' items and the linking to them from data architecture datagrams, so disconnecting the source systems from the demand for this type of data.
- Online data mapping tools and XML validation tools. These services would support developers in the creation of interfaces to the data bus and the testing of their adapters and message formats.

The decision on whether to organise central provision will be based on whether it addresses bottlenecks or inefficiencies in the take-up of the data architecture.

4.2 Implementation tasks and sequence

In this section we consider the tasks that will need to be carried out to implement the data architecture and make it ready for use. The tasks are grouped into 4 phases, which gradually implement the levels of integration described in Sections 3.2 and 4.1.

4.3 Integration of existing IT systems

To provide the industry reference data requests functionality described in Section 3.8.5, an individual data adapter interface will need to be built for each of the existing reference data systems to the data bus. This adapter will be in the form of a set of web services which receive data requests in data bus format and provide the response in data bus format.

Each such adapter will need to be designed around the requirements of the data architecture and the constraints of the existing system. Its design will also need to take into account considerations of security, access control, availability, performance and throughput,

Different existing systems will present different types of problem for their adapters. Older systems will need more 'façade' work doing to them to provide a web service interface, and may only be able to supply data in an existing standard file-based interface format that is not entirely compliant with the data architecture; newer systems may already be designed around a service-oriented architecture and so the effort will be more on creating the mapping between its internal message data structure and that of the data bus.

The adapters will need to be built by the maintainers of the source IT system, with the support of the data architecture team and will require the firm backing of the system's sponsors.

The work to provide each adapter will follow a development programme that should be iterative in nature. Each iteration will comprise the following stages:

- Inception. Outline the web services to be created, the form of request and response, the non-functional requirements such as availability, performance, accuracy, timeliness, and the security considerations. There may be feedback from this phase into the data architecture design, if there is found to be a fundamental misalignment between the data available and that required by the architecture.
- Elaboration: Define Integration Test Cases. Define how the web services interface is to be delivered (such as interface style, data serialisation, error states and messages, fallback behaviours). Prepare test cases, automated scripts and expected results.
- Construction: Build and carry out Unit Tests and Integration Tests. Build the web services so that they meet their test cases.
- Transition: Acceptance Test. Carry out end-to-end testing of the services' functionality. Gather configuration data for the live web service and update the Shared Services Repository. Prepare usage documentation and automated acceptance / regression test cases. On successful completion of the automated tests, declare the interface compliant with the data architecture and deploy it.

Each iteration should be geared to the delivery of a single query/response set of services or a small number of related ones.

The data adapter should be kept in a 'beta' or 'project-specific' state until it has been used in quasi-production use for some time and has stabilised i.e. there are no new feature or bug requests outstanding for it. It can then be promoted to be a full part of the data architecture, at which point its further development will need to be carefully version-controlled to avoid breaking changes (see Section 5.5).

4.4 Deployment - new projects

Whenever a new RCM data interchange project is implemented, the series of tasks described in this section will need to be carried out to ensure that the data interchange is compatible with the data architecture, to identify any opportunities for broader cross-industry data sharing and to make any changes necessary to the data architecture to accommodate the project's requirements.

We consider that there are three styles of project interaction, each with its own tasks:

- Bilateral data interchanges
- Data provision services
- Data processing services.

5 Governance

The analysis of the literature and development of the data architecture principles and requirements described in the preceding sections have generated some suggestions about the management activities that will need to be carried out centrally to develop and maintain the architecture. They relate particularly to the shared infrastructure elements and tasks described in Section 4.

5.1 Architecture repository management

A body will be necessary to manage the formal definition of the data architecture and to ensure that the artefacts which define it are readily available to users. This body will need to:

- Monitor the external standards from ISO, MIMOSA and others for changes which might impact on the data architecture
- Maintain the architecture Principles on which the data architecture is built to ensure that they keep in step with emerging IT best practice, technological opportunities and changes in the rail business environment
- Manage a Change Control process to gate-keep changes and extensions to the data architecture as it is developed, refined and extended through use in projects
- Manage any changes to UK rail industry standards and railway group standards based on the data architecture
- Manage the documentation and web assets which define the data architecture and make sure that they are available for implementers of the data architecture to use.

5.2 Architecture infrastructure management

Where shared IT infrastructure is used to support the implementation of the data architecture functions described in Section 4.1, a governance function is required to fund and provide for them and ensure that they are managed appropriately. This function will need to:

- Get industry support for the infrastructure investment required. This will mean soliciting and eliciting the needs of industry data sharers for the infrastructure, generating a business case, securing funding and setting up a development project.
- Work out the technical specification and non-functional requirements for the infrastructure (such as availability, performance, data quality indicators)
- Work out a contractual framework for its provision.
- Work out a contractual framework for its use by data sharers. Existing contracts between data suppliers and data users will need to include provision for the cost and responsibility for this use.
- Procure the supply of the infrastructure
- Monitor the performance of the infrastructure
- Manage a Change Control process for enhancements and extensions to the infrastructure to meet new demands on it.

5.3 Certification and compliance

The duty to provide data services in accordance with the data architecture will be included in commercial agreements between providers and data users, based on the framework established in project T1010-02 [23]. Data providers in particular will need to be able to confirm that the services they offer are compliant with the data architecture.

This suggests a need for a Compliance function to manage the certification of compliance to the data architecture. This will have the following tasks:

- Setting up and maintaining a set of tests and tools by which data providers and data users verify their own compliance with the data architecture. These may be standard datagrams which should be able to be processed correctly; XML schemas which can be used to validate datagrams; dummy service endpoints that can be used to test interactions and performance.

- Maintaining a register of compliant data providers
- Adjudicating on disputes between data providers and users on the reasons for problems with their data interchange, where these are germane to the data architecture
- Verifying performance of data providers' services according to their published or agreed Service Level Agreements, in terms of availability, currency, performance, data quality …

5.4 Support for implementers

Wide adoption of the data architecture will depend on making it easy to use. New users will need to be supported in getting connected to the data architecture. This suggests a support organisation whose responsibilities will include:

- Managing worked examples of using the architecture and other user-oriented documentation.
- Maintaining a user community website and wiki where best practice, users' experiences and support queries can be raised and answered; evangelising the use of this community documentation.
- Assisting in users' efforts to define the form and scope of their data interchanges making best use of the data architecture's facilities.
- Assisting in prototyping and proof-of-concept exercises by intending users.
- Facilitating changes and extensions to the data architecture to meet emerging user requirement.

5.5 Versioning and releasing

The data architecture will be subject to change throughout its lifetime. Particularly where shared infrastructure or common data services are involved, changes to the data architecture could affect other users than those any change was made for, leading to enforced software upgrades, failures of data interchange and additional cost.

Technical approaches and version numbering conventions for managing the versioning of the data architecture are discussed in the architecture Requirements document [30]. The goal of these approaches is the maximisation of both backwards compatibility (for example, where new data user code can use older data provider code) and forwards compatibility (for example, where a new version of a data service can still be used by data users using older client code), thus minimising the amount and rate of forced change for users.

There is a need for a body to regulate the versioning of the elements of the data architecture and to manage the release process in conjunction with the user community to minimise the disruption caused by change. The functions of this body will be:

- To assess all change requests for the data architecture to determine how they should be released and versioned based on the breadth of value, urgency of change and impact on the installed user base.
- To assist in defining the scope of changes so that they can be implemented with minimum disruption
- To manage a release schedule for new versions of the data architecture and to ensure that it is clearly communicated. This involves indicating what new functions are available to ensure that they are used; and indicating where breaking changes are likely to occur to existing connected systems.
- To manage a deprecation schedule for obsolete or superseded items which gives users ample time to migrate their code and data formats away from them.
- To liaise with the sponsors of reference data source systems to ensure that changes to these systems are managed with the minimum of disruption to the users of the data architecture.

6 Recommendations

This section contains recommendations on how the work done for this project should be taken forward. The theme behind them is to establish momentum for the reality of cross-industry RCM data exchange through practical steps, whilst simultaneously engaging the major stakeholders who will need to contribute to the specification, development and use of the data architecture.

6.1 Validation of the data architecture principles

The data architecture Principles described in this document and in more technical detail in the Architecture Principles document [4] need to be validated against good IT architecture practice, the direction of rail industry system development and the needs of data suppliers and consumers. The key tests are:

- Are the technical proposals coherent and consistent?
- Is there a sensible upgrade path into use of the data architecture from the current position?
- Are there any major technical hurdles to delivering the data architecture?
- Do the proposals represent a significant commercial risk for data contributors or users?
- Do the proposals cut across other developments in the rail industry or support them?
- Do the proposals require excessive central investment in governance effort and IT infrastructure?
- Will the proposed data architecture be stable over time?

Key stakeholders who should validate the architecture are:

- Network Rail’s ORBIS programme
- Network Rail’s LINX-TM programme
- The Rolling Stock Library replacement programme, R2

- Suppliers of bulk RCM data: train manufacturers and maintainers; traction power suppliers; trackside monitoring equipment suppliers; on-board monitoring equipment suppliers
- Consumers of RCM data: asset managers, asset data processing specialists; asset data visualisation specialists
- Academics in the field of RCM data integration and processing

6.2 Liaison with industry stakeholders

The development of the data architecture needs information and input from industry stakeholders – specifically, those that manage the key groups of rail assets being monitored: track, OLE, vehicles. The types of information required are:

- Data formats for standard rail data items such as track locations, asset IDs, vehicle numbers, and operational train numbers
- Data values for such as asset types, fault types, health levels, alert and alarm statuses
- Formats of standard data messages from overlapping message-based systems such as Network Rail's ESB and LINX-TM and the R2 programme

Some of the data services required to support RCM data interchange may have much more general application in the rail industry. Examples might be the Track Location Service, the Train Location Service and the Vehicle Identification Service, all of which would be of benefit to several completely different business processes.

There is thus considerable overlap of interest between the RCM initiative and other cross-industry initiatives such as the Information Portfolio Group, the DRACAS working group.

A process should therefore be set up to establish formal consultation links between the various programmes listed and to harmonise the production of data formats and data services across the IT initiatives involved.

6.3 Setup of governance arrangements

The governance requirements described in Section 5 will need to be established to ensure proper management of the data architecture is in place from the earliest stages. Decisions need to be taken at an industry level on where the responsibility for the various functions lies: whether each can be given to an existing regulatory or technical standards body or whether a new one needs to be set up.

6.4 Top-down architecture elements: standards

Although the data architecture will grow largely from the bottom up, the guiding principles and significant elements will need to be enforced or encouraged by the means of formal standards. These may take the form of a new Rail Industry Standard or amendments to existing Rail Industry Standards or Railway Group Standards.

The main elements which need central management will be:

- Compliance with legislation, existing Railway Group Standards and TSIs
- Compliance with guiding ISO standards, specifically ISO 13374, ISO 15926, ISO 8601
- Registers of reference data required to support the MIMOSA framework
- Licensing of shared frameworks such as MIMOSA
- Agreement on unique identifiers for all relevant items
- Reference documentation on the architecture
- Version management of architecture components
- Registry of data services supporting the architecture.

6.5 Bottom-up architecture elements: pilot projects

The main thrust of the development of the data architecture will come from its use in actual RCM projects. Pilot projects should be used to force the development of the main elements of the architecture and to bed in the processes for enhancing it through continuous extension and refinement.

The pilot projects should be chosen with the following goals in mind:

- Validating the key principles of the architecture in actual data sharing. These principles are:

  - Segmentation of RCM data according to the layers of the ISO 13374 stack.
  - A service-oriented approach to data interchange.
  - The use of ISO standards and MIMOSA standards for RCM data encoding.
  - The use of XML schemas for defining data packets and validating them.
- Setting up an embryonic data bus to allow simple message-based transfers of RCM data.
- Establishing the ground rules for creating adapters from existing RCM data systems to the data architecture.
- Setting up embryonic reference data services to establish the use of shared reference data from the start.
- Replicating the type of process that future real projects will need to follow, so that guidance documentation is generated and the interactions between parties trialled and documented.
- Setting up an embryonic ontology and related SPARQL endpoints to bench-test the use of ontological techniques for RCM data linking in the context of the data architecture.

We recommend two pilot projects, to take place simultaneously, with an overlapping element of shared work which may be considered part of the project control function or of a third pilot project. Each of the projects will have several phases, each subsequent one of which will demonstrate a further use of the data architecture. These projects are described in the following paragraphs.

6.5.1 Pilot project 1 – trackside detector data sharing

This project would involve two different-type trackside detectors from different suppliers making their data on passing vehicles available to a third party in consistent formats compliant with the data architecture. If the two detectors were chosen so that the same vehicles were known to pass over them both, even if on different dates, further data integration possibilities would be opened.

Phase 1 would involve both systems generating datagrams containing State Detection and Data Manipulation level data for passing vehicles, with the data content being compliant with the data architecture. This would demonstrate that mapping this type of data to the architecture is possible. At this stage the vehicles and axles within vehicles would be identified using the source systems' own vehicle identification mechanisms.

Phase 2 would require both suppliers to look up vehicle and axle information from a specially-built vehicle data store populated with known vehicle data, made available via a web service compliant with the data architecture and with the future Vehicle Identification Service. If the same vehicles pass both systems' detectors and are recognised by their AVI, information about the axles of each vehicle will be able to be merged by the third (data-reading) party from the two systems.

Phase 3 (optional, if the same vehicles pass both systems detectors) would add an embryonic Vehicle Location Web Service based on a known timetable and known vehicle allocation to services from, say, the operator's Gemini or Genius or similar stock dispatching system. This would enable a third trackside detector (say an HABD) with no in-built AVI to be connected and for the train consist to be 'estimated' based on known set consists, train allocations and the timetable. This service could be extended to receive tag IDs / EVNs from the source systems to enable detailed vehicle data to be supplied.

6.5.2 Pilot project 2 – train-gathered infrastructure data processing

This project would involve an existing train-mounted data gathering system such as a TMS measuring pantograph voltage, or a UOMS-like OLE monitoring system, and the automatic gathering of data from such a system and publishing via a web service compliant with the data architecture so that multiple users could carry out processing of the gathered data.

Phase 1 of the project would involve the formatting of the gathered data into Data Acquisition or Data Manipulation datagrams, with all the recorded data being shown in open data formats that conform with the data architecture. This would initially be available as a .csv file for each data event, then in an XML datagram with suitable metadata indicating the calibration setup of the recorder.

Phase 2 would involve the setup of a web service which would respond to requests for data for a given location or geographical area and date range, returning properly-formatted XML datagrams compliant with the data architecture.

Phase 3 (optional, if a suitable service is available from Network Rail) would enhance this to add the track position to each recording, based on lookup of an Network Rail data service which would accept a train ID or movement direction and a series of GPS positions and return the track position of each reading in ELR / miles / chains or other standard method.

7 References

References

[1] W3C, “XML Schema 1.1,” [Online]. Available: http://www.w3.org/XML/Schema. [Accessed 2014].

[2] W3C, “SOAP Version 1.2,” 2007. [Online]. Available: http://www.w3.org/TR/soap12-part1/.

[3] IETF, “RFC 3986 Uniform Resource Identifier (URI): Generic Syntax,” 2005. [Online]. Available: http://tools.ietf.org/html/rfc3986.

[4] W3C, “Web Services Description Language Version 2.0,” 2007. [Online]. Available: http://www.w3.org/TR/wsdl20/.

[5] “Rail Technical Strategy 2012 The Future Railway,” 2012. [Online]. Available: http://www.futurerailway.org/Documents/RTS%202012%20The%20Future%20Railway.pdf.

[6] L. McNulty, “Rail Value for Money,” 2011. [Online]. Available: http://orr.gov.uk/__data/assets/pdf_file/0017/1709/rail-vfm-detailed-report-may11.pdf.

[7] T1010-01 I, “T1010-01 I Review of Relevant RCM Developments,” 2014.

[8] RSSB, “T1010-01 II Architecture Principles,” 2014. [Online].

[9] RSSB, “T857 Detailed Overview of Selected RCM Areas,” 2012. [Online]. Available: http://www.sparkrail.org/_layouts/Rssb.Spark/Attachments.ashx?Id=75NEMTS3ZVHP-8-3018.

[10] RSSB, “T853 Mapping the Remote Condition Monitoring Architecture,” 2012. [Online]. Available: http://www.sparkrail.org/_layouts/Rssb.Spark/Attachments.ashx?Id=75NEMTS3ZVHP-8-3033.

[11] Network Rail, “Technical Strategy 2013,” 2013. [Online].

[12] RSSB, “T912 Railway Functional Architecture,” 2011. [Online]. Available: http://www.sparkrail.org/_layouts/Rssb.Spark/Attachments.ashx?Id=75NEMTS3ZVHP-8-2987.

[13] DCMI, “Dublin Core Metadata Element Set, Version 1.1,” 2013. [Online]. Available: http://dublincore.org/documents/dces/.

[14] ISO, “ISO 13374-1:2012 Condition monitoring and diagnostics of machines -- Data processing, communication and presentation,” 2012. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=21832.

[15] RSSB, “T844 Mapping current remote condition monitoring activities to the system reliability framework,” 2009. [Online]. Available: http://www.sparkrail.org/_layouts/Rssb.Spark/Attachments.ashx?Id=75NEMTS3ZVHP-8-3031.

[16] ISO, “ISO 15926,” [Online].

[17] UIC, “RailTopoModel RC2,” July 2014. [Online]. Available: http://documents.railml.org/science/280714_uic_railtopomodel_rc2.pdf. [Accessed 08 09 2014].

[18] C. Roberts and J. Easton, “The Specification of a System-wide Data Framework for the Railway Industry - Final Report,” 2011.

[19] MIMOSA, “MIMOSA OSA-CBM,” 2006. [Online]. Available: http://www.mimosa.org/?q=resources/specs/osa-cbm-330.

[20] MIMOSA, “MIMOSA OSA-EIA,” 2012. [Online].

[21] MIMOSA, “MIMOSA Structured Digital Asset Interoperability Register,” [Online]. Available: http://www.mimosa.org/?q=wiki/structured-digital-asset-interoperability-registry-sdair-functional-requirements.

[22] ISO, ISO 8601:2004, ISO, 2004.

[23] EPSG, “EPSG Geodetic Parameter Dataset,” [Online]. Available: http://www.epsg-registry.org/. [Accessed 11 09 2014].

[24] Wikipedia, “UUID,” [Online]. Available: http://en.wikipedia.org/wiki/UUID#Random%5FUUID%5Fprobability%5Fof%5Fduplicates. [Accessed 10 06 2014].

[25] G. Hohpe and B. Woolf, Enterprise Integration Patterns, Boston, MA USA: Addison Wesley, 2004.

[26] RSSB, “T1010-02 Commercial Framework for Cross-Industry Remote Condition Monitoring Data Sharing,” RSSB, 2014. [Online].

[27] RSSB, “T1010-01 III Architecture Requirements,” 2014. [Online].

[28] NIST, “NIST SI Units,” 2008. [Online]. Available: http://physics.nist.gov/Pubs/SP330/sp330.pdf.

[29] IANA, “MIME Internet Media Types,” [Online]. Available: http://www.iana.org/assignments/media-types/media-types.xhtml. [Accessed 21 05 2014].

[30] W3C, “XML Encryption Syntax and Processing Version 1.1,” 2013. [Online]. Available: http://www.w3.org/TR/xmlenc-core1/.

[31] W3C, “XML Signature Syntax and Processing Version 1.1,” 2013. [Online]. Available: http://www.w3.org/TR/xmldsig-core1/.

[32] W3C, “PROV Series of Documents,” 2013. [Online]. Available: http://www.w3.org/TR/prov-overview/.

[33] W3C, “Dublin Core to PROV Mapping,” 2013. [Online]. Available: http://www.w3.org/TR/prov-dc/.

[34] IETF, “RFC 4122,” 07 2005. [Online]. Available: http://tools.ietf.org/html/rfc4122. [Accessed 28 05 2014].

[35] R. Fielding, “PhD Dissertation "Architectural Styles and the Design of Network-based Software Architectures",” 2000. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.

[36] IETF, “RFC 2119 Key words for use in RFCs to Indicate Requirement Levels,” March 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2119.txt. [Accessed 10 06 2014].

[37] IETF, “RFC 6919,” 01 04 2013. [Online]. Available: https://tools.ietf.org/html/rfc6919. [Accessed 10 06 2014].

RSSB R&D Programme
Floor 4, The Helicon
1 South Street
London
EC2M 2RB

enquirydesk@rssb.co.uk

http://www.rssb.co.uk/research-development-innovation/research-and-development