RSSB

Research Programme

Engineering

Ontology-based data management for the GB rail industry

Feasibility study

Copyright

Written by:
Jon Tutcher, John Easton, and Clive Roberts, University of Birmingham
Rob Myall, Mark Hargreaves, and Colin Tiller, Siemens Rail Automation

Published:
August 2013

Table of Contents

1 Introduction

The 2011 value for money study on the UK railways (DfT, 2011) found that:

> *"…the effectiveness of the industry's Information Systems is inhibited by a suite of legacy systems that are expensive to run, unable to communicate with new technology and encourage users to develop a wide range of bespoke local systems to overcome limitations. Many legacy systems were created and managed in company silos, with only a few systems crossing industry boundaries."*

In response, the industry has since been looking hard at both its current systems catalogue and future information needs. In particular, the 2012 Rail Technical Strategy (RTS) (TSLG, 2012) presented a vision of the future railway where:

> *"The businesses in the industry are information-rich and use that information effectively to enhance and drive decision-making processes. Advanced and appropriate decision support tools are in place and in use on a daily basis."*

The Semantic Web (Berners-Lee et al., 2001) is expected to be the next-generation of the World Wide Web in which data resources are enriched with machine processable metadata that describes their meaning (Denker, 2005). Through the provision of this additional contextual information, computers on the Semantic Web are able to better understand and evaluate the appropriateness of data for responding to a user queries, leading to increased automation of data processing tasks, more effective search and more accurate responses. As of October 2013 >15% of web sites implement some Semantic Web technologies (Guha, 2013).

Two of the cornerstones of the Semantic Web vision are conceptual data models and linked open data; for the Semantic Web the W3C have recommended that these be implemented in the form of ontologies and the Resource Description Framework (RDF) respectively, with RDF enabling the description of simple facts (e.g. "locomotive" hasPart "wheel") and ontology allowing those facts to be placed in an unambiguous context – essentially relating each element to a concept in a published model of the world. A more detailed description of the relationship between RDF and ontology will follow in section 2 of this document.

This feasibility study, which is part of the FuTRO Universal Data Challenge, aimed to show how shared, open access ontologies and linked data could help the UK rail industry realise the vision presented in the RTS, facilitating access to information resources and enabling their easy integration / combined usage in forming responses to operational queries. The study shows how ontology will enable the industry to adopt a "build once" approach to application development, protecting software from changes to physical systems in the real world.

The study begins by introducing the key technologies being discussed, namely ontology and RDF, and explains their relationship to each other and the rest of the Semantic Web. It then touches on some common concerns around the technologies, in particular security considerations, before presenting a series of brief case studies discussing how Semantic Web technologies have been used in other domains (oil and gas, retail and the public sector).

The study then introduces the core ontology model for the rail domain, developed at the University of Birmingham, before finally going on to illustrate the some of the benefits of ontology models through two technology demonstrators: the first of which uses the example of the Siemens Asset Monitoring as a Service (AMaaS) platform to show how the combination of ontology and linked data can enrich asset information management and the second, which uses vehicle positioning as an example of the preservation of application functionality through a systems upgrade.

2 Ontology and linked data

2.1 What is ontology?

Ontologies are often described in textbooks and journal papers as being “an explicit specification of a conceptualisation” (Gruber, 1993). That is somewhat inaccessible, so perhaps a more helpful description is that “ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge. They provide potential terms for describing our knowledge about the domain” (Chandrasekaran, Josephson, & Benjamins, 1999). Put simply, ontologies describe data by referencing published models of a domain; because of this, as long as the published models are both reasonably complete and share common root concepts, ontologies allow computers to combine pieces of information drawn from different data sources, understand the relationships between them, and infer new facts about the world.

In practice, two things are required to develop an ontology; a set of controlled vocabularies of terms used within the domain, and a set of related classes and rules that can be used to describe the domain from a particular viewpoint. When using a conceptual data model, developers state the relationship between an item of data and the model of the world, allowing that data to be seen in context by the computer. As an illustration, consider the US English and British English use of the word jelly. Jelly is a perfectly acceptable term in both US and British English, it even has the same syntax (can be used in the same places in sentences), but the meaning of the term, its semantic, is different in the two languages; in British English jelly is a gelatine-based dessert, whereas in US English jelly is a fruit preserve (a jam in British English). In a conventional data model such as a standard XML document, the tag jelly can be used ambiguously, because XML schema only enforce the positioning of the tag in the document and the values it can take, not its meaning in that context. Participants in a solely XML-based data exchange could legitimately use the tag jelly for either meaning, even if the designers of the XML schema had a particular usage in mind. By representing the data in a conceptual model this situation can be avoided, because the term jelly is defined as being a type of dessert that is composed of gelatine, and which may have a particular colour, flavour, shape and wobbliness.

Once facts have been entered into an ontology, it becomes possible to use the relationships within the model to infer new information about the world; a process known as reasoning. On the simplest level this could involve the user stating that an object “377 401” is an instance of “Class 377” and the reasoner then inferring that “377 401” must be a train because the ontology shows that “Class 377” is a type of train. Over and above this type of simple class-based reasoning, ontologies can be further enhanced by the additional of rules. The addition of rules make it possible for ontologies to capture and use the more complex logical statements that are needed for complex decision making, for example the rule “if axle bearing temperature sensor X has a reading of over 100 degrees, then the axle bearing sensor x is monitoring is faulty” can be used to conditionally add a new fact “axle bearing y is faulty” to the model if needed. By expressing these rules in the ontology, the operational logic associated with a domain can be stored in the data model, rather than in the code of individual applications, making it easier to manage changes (there is a single source to change, rather than many individual applications).

The World Wide Web Consortium's (W3C) recommended technology for the implementation of ontology models is the Web Ontology Language (OWL). Now in it's second major release, OWL 2 defines a number of subfamilies (called profiles) of the language that are intended to provide a number of trade offs between model expressivity and computing time. OWL 2 DL is the most expressive of the profiles allowing complex reasoning across data models at the cost of long processing times (time taken to complete reasoning becomes exponentially larger with every new fact in the data store). The less expressive OWL 2 EL profile by comparison, guarantees that reasoning will complete in polynomial time with respect to the size of the ontology but limits the range of ideas that can be expressed. A number of other OWL 2 profiles are also specified for other purposes; these include OWL QL, a profile intended to allow standard query languages to utilise ontology (representing relational databases), and OWL RL, which allows expressivity for an ontology to be represented using the logic employed in rule languages.

Although to date the majority of implementation efforts around ontology have been in the medical and biological sciences, where the large number of existing controlled vocabularies and structured information sets have provided low-hanging fruit for development terms, the new models have attracted a certain amount from many industrial domains, including transportation. Early adopters of the technology in this space include the EU-funded Framework 6 projects REWERSE and InteGRail, with the latter specifically focusing on the development of simple ontology models for rail and their application to a number of industry problems such as network statement checking for vehicles and the integration of data from different condition monitoring systems.

2.2 Linked data

Although ontologies go a long way towards enabling businesses to derive new knowledge from their existing data resources by allowing the meaning of data to be expressed unambiguously, the data itself must of course be accessible to all the potential end-users. Industrial information systems have traditionally existed as silos, self-contained entities that gather, process, archive and return their own data directly to the end-user for use in a specific context. In order to use the data from these silos in a wider context, interfaces must be provided that allow access to the data in known formats. The W3C recommends two principle standards for this purpose; the Resource Description Framework (RDF), which defines a mechanism for storing facts in semantic data models and forms the basis for most modern ontologies, and SPARQL, a powerful query language designed to allow easy interrogation of datasets.

RDF is a modelling language that allows data and associated semantics to be stored using a graph-based approach similar to Entity-Relationship diagram models (Chen, 1976). RDF uses the concept of 'triples' to record relationships between defined entities, statements consisting of a subject (the entity the relationship is referring to), a predicate (the type of relationship, for instance 'isA'), and an object (the piece of data the subject is linking to). Examples of triples include "Snoopy" isA "Beagle" and "Pendolino" hasPart "Wheel". By building up a number of RDF triples linking entities together, a graph can be composed showing relationships between concepts in a domain, resulting in a machine-understandable picture of the semantics of each one. RDF places few restrictions on data structure or vocabulary and as such cannot be used alone in machine reasoning applications.

RDF Schema (RDFS) extends RDF by defining a set of relations with fixed, defined semantics on which reasoning can be performed. Whilst RDF itself only defines basic structure and allows declarations of subject/object/predicate types, RDFS formally defines inheritance relations, annotation conventions, and several other useful constructs, allowing basic but powerful inference to take place across datasets. Classifications of class, membership, property range/domain and annotation are part of the RDFS specification; several most standard 'linked data' ontologies are defined only at this level. OWL (described above) sits atop RDFS as a modelling language, and enables even more powerful reasoning; hence its primary use in industrial and biomedical information systems.

2.3 Intelligent data and the semantic web

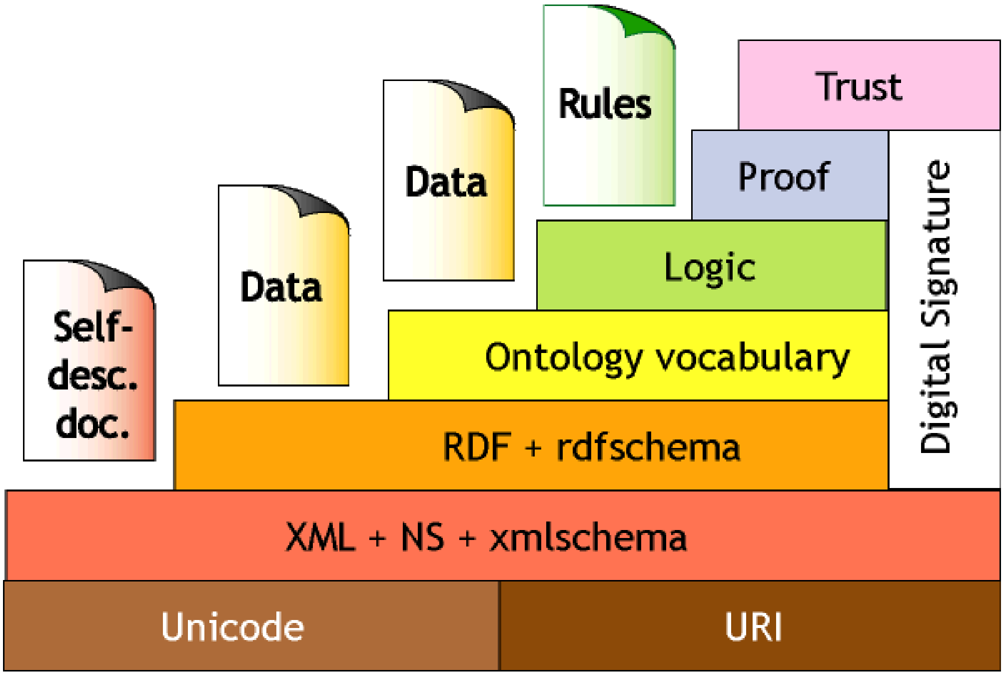


Figure 1: One view of the Semantic Web software stack (Berners-Lee, 2000).

In a wider context RDF and ontology form part of the Semantic Web stack (Figure 1), a multi-layered architecture in which existing data resources (for example databases of vehicles or known locations) are assigned Uniform Resource Identifiers (URIs) and described in RDF. Relationships and business rules described in the ontology are used to allow the inference of new knowledge in response to user requests (the logic layer). The upper layers of the stack then deal with tracking the provenance of the information (where the underlying data was generated, by whom and for what purpose) and with the evaluation of the data. The Semantic Web stack is built on existing Web standards and protocols for data transport, enabling existing tools to be used for implementation.

2.4 Security in open data systems

One of the great benefits of using conceptual data modelling approaches, such as ontology, is that they facilitate the easy integration and combined usage of data sets drawn from a diverse range of systems, resulting in richer information than could be derived from a single data resource in isolation. Underpinning this benefit is the assumption that the organisations involved have provided suitable access to a wide range of their information systems, resulting in a more *open* and *dynamic* ICT architecture than is traditionally used by industry.

Realising an appropriate level of information security in this type of architecture is technically challenging within a single organisation, where all the involved individuals are subject to the same ICT policies and have credentials issued by a single authority; however, in order to maximise on the benefits of open data, information resources ideally need to be made accessible across organisations / to all the stakeholders involved in the delivery of a business function (e.g. the designers of a product, the component supply chain and the manufacturer). In an open data environment the identity of individuals requiring access to an information resource may not be predetermined at the time of the request being issued (at least within the ICT systems of the data provider) and the list of individuals requiring access may change regularly (Kagal, 2006); individuals may be able to provide valid credentials, such as name / passphrase pairs, public / private keys, or certificates and yet not have pre-defined roles or explicit permissions within the information system. Therefore, in common with many other service-based approaches to ICT architecture, new ways of thinking about trust and the identity of individuals are required in open data systems.

Although a complete discussion of the security implications of semantic and open data systems is well beyond the scope of this feasibility study, the authors recognise that to businesses understanding who is accessing information, why they are accessing it, and how it is ultimately being used is of vital importance; as such, in order to demonstrate that conceptual data models have value to offer the UK rail industry, it is necessary to prove that acceptable methods for both securing data resources and recording the provenance of the data they contain exist for Semantic Web technologies. To date, for ontology and open data systems the relevant security models involve controlling access to RDF data, protecting and signing components of the ontology model, ensuring secure links to web services and guaranteeing the provenance of data being used for decision making. The relationship between the various security models is summarised in Figure 2.

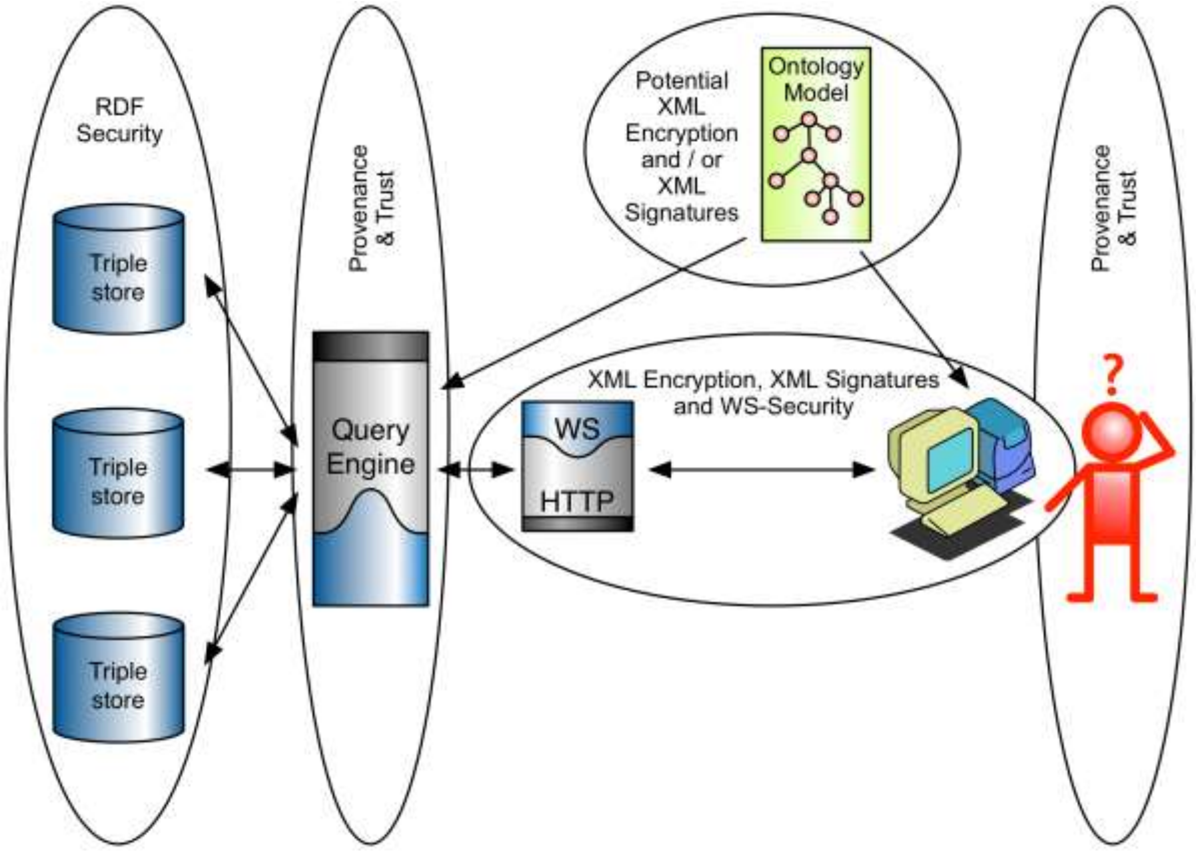


Figure 2: Relationship between the various security models.

2.4.1 XML security

As OWL and RDF can be expressed in the form of XML documents, the first step when considering security standards for the Semantic Web is to review XML security techniques. While ontologies may benefit from being open to all, thus encouraging reuse, there are obvious business use cases where the ability to keep parts of a model private have value; examples include extensions to an ontology model that describe the details of a particular organisation's product line, or simply guaranteeing that a model being used has not been maliciously altered by third parties since it was created. The W3C currently provide three sets of recommendations on XML security: XML encryption, XML signatures, and XKMS (W3C, 2014).

XML encryption is used when one or more elements (or the content thereof) of an XML document are confidential, for example in the case of financial records. The elements of the document to be encrypted are replaced with an <EncryptedData> element, optionally alongside supporting information such as details of the encryption algorithm used, the cypher key etc. As parts of a document can be encrypted independently, the XML encryption system is ideally suited to the protection of specific extensions to an ontology prepared by organisations for use by a limited audience (as described above); by encrypting the parts of the model they wish to remain private organisations can publish the whole OWL model online, enabling use of the non-confidential components by the public while ensuring the encrypted elements are only accessible to authorised parties.

XML signatures provide a mechanism for ensuring that the content of all (or part) of an XML document is as the publisher intended and has not since been altered by a third party (usually for malicious reasons). Put very simply, digital signatures work based on the information publisher having a public and private key pair; the private key is used to encrypt the data being signed before it is published, the published content can be decrypted by applying the public key, which the publisher makes available via a public key authority (a "trusted" organisation that all users agree is an authoritative source of ID information) thus proving the authenticity of the document's contents. As ontologies are designed to enable specific semantic information to be conveyed between stakeholders, it is obviously vital that all parties can trust the models being used have not been tampered with. Signed content in an XML document is contained within a <Signature> element, which itself contains a <SignatureValue> (the value of the digital signature) and a <SignedInfo> element (containing the signed content and algorithm details).

The XML Key Management Specification (XKMS) provides guidance on how the public keys used for XML encryption and XML signatures (amongst other things) should be distributed and managed. Although a vital component of the security infrastructure, the XKMS recommendation is only of interest at a very specific technical level and therefore no further elaboration will be made here.

2.4.2 RDF security

Unlike traditional data stores, such as relational databases, where data is grouped and thus a certain amount of access control is comparatively easy to enforce (in SQL this is done at the table level), most RDF triple stores keep all the facts in a single common space. This approach makes data integration much easier but comes at the cost of reduced security (any user with access to the data store has access to the set of triples it contains).

Providing appropriate access control to triple stores is still very much an open area of research, with common approaches ranging from the attachment of tags containing access control information to individual triples (Papakonstantinou, 2012) (very fine-grained control but computationally intensive, the method currently used by Oracle in their RDF database tools) to access control at the level of named graphs, which are defined collections of interrelated facts on a particular topic (for example all the information relating to an individual or vehicle). The latter approach is used in the Virtuoso triple store and offers less fine-grained control but does have the advantage of being computationally efficient (a single check is needed for each graph queried).

2.4.3 Web service security

Although other access regimes are conceivable, the inherently online nature of Semantic Web technologies means that linked open data, such as that described in this report, is almost always made available via web services. Web service security is a relatively well-understood problem by the industry and the WS-Security model, published by the Organization for the Advancement of Structured Information Standards (OASIS), is used to ensure that confidential information is protected and that the identity of the sender can be confirmed. WS-Security builds on the XML encryption and signature models described in Section 2.4.1. Additionally, these technologies build upon the OSI web technology stack; as such security methods at the physical or network layer level can also be utilised (such as SSL encryption and physical separation of communication media).

2.4.4 Provenance and trust

The growing reliance of the business community on accurate, up-to-the minute data has led to a number of initiatives designed to describe what information a data resource contains, how it was gathered, and what processing occurred since. Foremost amongst these are W3C's PROV family of recommendations, aimed at ontology amongst other standards, and the Vocabulary of Interlinked Datasets (VoID) for description of RDF metadata.

It is worth noting that numerous other provenance models also exist in varying stages of completion (such as Open Provenance Model, Provenir, and SWAN), although these tend to pre-date the W3C recommendations.

2.4.4.1 PROV

PROV (W3C, 2013) is a family of recommendations produced by the W3C. PROV aims to allow the wide publication and exchange of provenance on the Web and other information systems. PROV is based around a conceptual model that can be serialised in many different formats, including XML and OWL (PROV-O) and supports the representation of:

1. Object identity, attribution of an object to person or entity, and representation of processing steps.
2. Access to provenance-related information expressed in other standards.
3. Accessing provenance information.
4. Traceability of the provenance of provenance information.
5. Reproducibility (of processing).
6. Versioning of objects.
7. Representation of procedures.
8. Representation of object derivation.

2.4.4.2 VoID

VoID (W3C, 2011) is a recommendation for describing the metadata associated with RDF datasets, acting as a bridge between the end-users and publishers of RDF data. It is intended to help with the discovery and usage of appropriate RDF data for forming the response to a query. VoID enables the conveyance of:

1. General metadata for the dataset (for example: title, description, date of creation, publishers, contributors).
2. Metadata relating to access procedures using available protocols.
3. Structural data describing the schemas followed.
4. Descriptions of any links to external datasets.

2.4.4.3 Propagation of trust

The propagation of trust in inferred information (whether you can trust derived information based on trust of the source information) is still an open research question and requires further investigation by the community (Thuraisingham, 2005). Although the additional contextual and provenance information that ontology makes it possible to convey will, ultimately, mean that these problems are solved, at present it is largely down to the individual user to decide whether, based on their degree of trust in the quality of the original data sources and their understanding of the processing that the data has undergone, they feel able to trust newly inferred facts.

2.5 Why adopt semantic web technologies?

A question that is frequently asked by members of the industry is why, given that we're already investing in XML interfaces between our key business systems, do we need to develop ontologies within GB rail?

The simple answer to this question revolves around flexibility, in particular the flexibility to respond to a changing technological environment both within the rail industry and in the wider transport sector, and the flexibility to easily access information that could help the return to normal running in periods of disruption. As noted earlier in this document, the 2011 rail value for money study (DfT, 2011) found that:

> *"…the effectiveness of the industry's IS is inhibited by a suite of legacy systems that are expensive to run, unable to communicate with new technology and encourage users to develop a wide range of bespoke local systems to overcome limitations. Many legacy systems were created and managed in company silos, with only a few systems crossing industry boundaries."*

Few of those working in the industry would have been surprised by this statement; the problems with the industry's information systems are well known and projects are already in place to update or replace some of the most business critical systems such as RAVERS and RSL. XML interfaces are being added to many systems to help breakdown the traditionally siloed approach to information management and to an extent at least, the problem is in-hand. However, while at present the need for renewal of information systems is particularly obvious, as the railway moves forward there will naturally be an on-going slow churn of information system renewal over time. Each time a system is replaced the industry is faced with the prospect of updating and maintaining links to that system in other software, a process that is hugely expensive in terms of developer time. One solution to this problem is to create a common data model that is supported by all information systems within the industry; this has been possible for a number of years and could easily be achieved, at least to some extent, using common models within Europe, such as railML for infrastructure data.

Unfortunately however, problems still exist with this approach. Data models such as railML are based on particular file structures known as schemas. When the industry chooses to update the common model it is using as the integration layer between systems, the schema of the model will change and the individual information systems will need to be updated accordingly; while this is almost certainly a less developer intensive activity than maintaining direct links between individual systems, it still comes with an associated cost in terms of time and the inevitable slow creep towards poorly understood application code.

The concept-based nature of ontology models offers a solution to this problem; firstly, while the model will undoubtedly evolve over time as new ideas are added, the concepts already in place should not change substantially. Secondly, the way data is expressed for use with the ontology (as a set of facts in RDF) means that there is no specific file format that needs to be followed, new information can be added without changing the way existing applications use or access it. Ontology will allow the industry to develop applications once and then continue to use them despite changes to the external environment over time.

Open and linked data also have value to offer the industry in periods of disturbed running. In normal service the data exchange requirements of the railway are comparatively well understood, with staff needing access to known information based on their roles. In disrupted scenarios however, staff attempting to resolve the problem may require access to all sorts of information to determine whether it is safe for running to resume on a given section of line. Take for example the case of flooding; using linked data it would be possible, based on the known location of the flood, to access information on the type of substrate supporting the track, whether there are any services such as gas lines nearby that might be effected by the loading of the sodden ground as trains pass, and whether the vehicles running on that line can safely be run through flood water (and if so, based on the vehicle type, at what speed can they be run).

In an attempt to illustrate the flexibility of Semantic Web technologies, Section 3 will present some examples of their use by other industries (oil and gas, retail and the public sector).

3 Conceptual models in industrial contexts – case studies and best practice

Although still not commonplace, concept-based data models and linked open data resources have been growing in popularity in recent years; many larger organisations in fields as diverse as heavy industry (oil & gas, automotive etc.), the media and healthcare have started working with the models and exploiting some of their unique abilities, particularly in knowledge management systems.

The following section presents a selection of case studies draw from various industrial sectors illustrating how the technologies have been used and what lessons have been learned from the process.

3.1 Conceptual models in the oil and gas industry – ISO 15926

The ISO 15926 standard defines a generic data model for industrial processes, along with a reference data library of plant hardware. It was initially developed to serve as a data integration framework for use in the oil and gas sector, however the generic approach taken by the development team means that, given a suitable reference data library, it could be used in any large infrastructure system.

ISO 15926 differs from many other conceptual data models in that it not only defines objects with respect to their composition (an oil pipeline is composed of some lengths of pipe, some pumps, some valves, and other items) but also how that composition has changed over time. The treatment of time as a core component of the data model by the ISO 15926 developers, mean that the resultant four dimensional (4D) model is very well suited to tasks such as lifecycle management of component parts, an issue that is also prevalent in the rail industry and that can be best illustrated by the questions “which switches has motor xyz been used to drive during its lifetime?” and “are the reliability issues we have seen on switches a, b and c due to the reuse of a common component (potentially following refurbishment)?”

The first projects that would later contribute to the development of the 15926 standard (itself released in around 2003) began in the early 1990s, before the World Wide Web was widespread and over a decade before the W3C recommended OWL and its associated technologies (such as RDF) be used for the implementation of this type of model. The long timeline associated with the development of the model has therefore meant that the reference implementation of 15926 has migrated between languages over time, with examples including EXPRESS, XML, and most recently OWL.

3.1.1 Lessons learnt from the development of ISO 15926

At a recent workshop facilitated by the University of Birmingham (UoB) team on behalf of RSSB, Mathew West, one of the developers of the ISO 15926 standard, related his experiences of implementing a conceptual in an industrial context over a 20-year period (Easton, 2013). A summary of his thoughts and in particular the lessons learnt is included below.

3.1.1.1 Information

The correct use of information allows enterprises to identify opportunities, reduce risks and respond to change. Despite this, large organisations often find it difficult to justify investment in information systems and processes. Over time, the benefits of information-based decision making are often found to be embarrassingly large. However, if pressed to find justifications for investments in information management, remember that first and foremost information is used by businesses to reduce the risks of decision making; as such, the basic strategy to follow is when considering a new project are:

- Identify areas of the business where decision making has gone wrong.
- Investigate the failures of information provision (such as incomplete or late data, and inconsistent information from multiple sources) that led to the problem.
- Determine the cost to the business of the failure to provide suitable information to the decision maker (wasted materials, repetition of work, lost opportunities).

Good delivery of information is all about quality: meeting agreed customer requirements for key properties of data such as accessibility, consistency, provenance, timeliness, accuracy, relevance, and cost. If a customer has failed to state a requirement, then by definition the information supplier cannot fail to meet it. As such, careful specification of information requirements is vital when attempting to ensure information quality.

3.1.1.2 Data standards

Standards describe established best-practice methods of performing tasks. In data sharing the use of standards provides a common language for communications based around a shared data model and reference data set; standards are a key enabler for data exchange between systems and parties. Implementing data standards can take many years and a substantial amount of money before the first benefits are realised. However, adaption and reuse of existing standards can help to both reduce development time and improve compatibility with other models. Developed standards are in many ways repositories of the available knowledge in a field; therefore, data standards are in a continuous state of development as new domain knowledge is taken into account.

The development of new data standards requires that:

- A consortium of interested parties be established to guide the development and use the new standard.
- An independent body is identified to facilitate the development/provide administrative support.
- Sources of funding for development and maintenance are decided upon (voluntary donations, subscription)
- The standards environment is identified.
- Existing standards are adapted/developed where possible.
- The new standard is deployed.

3.1.1.3 Business models

Developing business models for the adoption of new data standards can be difficult. Contractors, for example, make money by selling man-hours and as such will resist efforts to make data exchange easier/reduce their workload. Software vendors too have something to loose from making it easier to move data out of their applications, as customers could then choose to switch to a competitor's system. In the oil industry, this problem was solved by brute force, with the industry announcing to vendors that it would only purchase software that supported the new standards. However this type of commercial issue is solved, it is vital when establishing standards within an industry that a business model is found which makes commercial sense for both the providers and receivers of data.

3.1.1.4 Governance and custodianship

Good governance of standards is vital if all involved parties are to follow the rules and changes are to be made to the standard over time. Establishing an effective framework for governance can however pose significant challenges, particularly in domains where authority is devolved (such as in the rail industry). Governance may be achieved through voluntary or regulatory approaches; voluntary approaches have the advantage of development effort being contributed by the consortium on a voluntary basis. Unfortunately however, this may ultimately lead to longer implementation timescales, patchy implementation of the domain based on partner interests, and a lack of clear leadership in the event of disputes. Regulatory approaches provide clear leadership that can be used to force consortium members to contribute to development and adopt the resultant standard, however they risk alienation of partners and avoidance (attempts to work around rules disliked by particular stakeholder groupings).

All standards require a body to administer and maintain them after the initial development effort is completed. This role includes housekeeping of any master and reference data. The custodian of a standard must be independent, respected by the user base, follow defined rules and processes, and operate a quality management process that ensures the standard remains relevant and consistent.

3.2 Conceptual models in the retail sector – good relations

'Good Relations' (Hepp, 2008) is a project that offers semantic web integration for online retailers by leveraging product information contained in existing online e-commerce systems. It provides the means for retailers to customise their semantic web visibility, exposing their products and services to semantic web software agents search as comparison sites, search engines, or accessible web browsers. Endorsed by Google and Yahoo amongst others (GoodRelations, 2011), the system allows data presented in a serialisation of OWL DL to be requested by clients, detailing 'offers' for products and services (the sale, rent, or maintenance of a particular thing). These offers are linked to a semantic web description and specification of each product if possible, usually hosted on a manufacturer's website, allowing computer systems to make comparisons between offers available at different shops. It is possible to query all good relations enabled shops to find (for example) prices for dealers selling TVs sized between 28" and 32", below £300, within a 20 mile radius of the user's house, and have results returned with no ambiguity. The power of this aspect of the semantic web is discussed in theory by Tim Berners-Lee's 2001 paper, which provides a detailed example of software agents being used to determine medical treatments for a set of particular circumstances (Berners-lee, James Hendler, & Lassila, 2001).

The benefits of this are self-evident to the end user, but provide a business incentive to retailers too – easier, more accurate search may encourage customers to use the semantic web enabled applications. The project is built on a number of international standards (unit and language specifications amongst others), and has also been built to tie in with other semantic web efforts. The "Friend of a Friend" (FOAF) language (Brickley & Miller, 2010), for example, is another large project set out to define semantic links in social networks between people. FOAF entities are utilised in Good Relations where necessary, to provide context-aware information about individuals and companies.

3.3 Conceptual models in the public sector – legislation.gov.uk

The National Archives is the official archive and publisher for the UK government. Its collection includes over 11 million historical government and public records dating back hundreds of years. One of the key services offered by the archive is legislation.gov.uk, a portal that provides members of the public with access to all UK legislation passed since 1988 and much of what was passed from 1948 onwards, all under the Open Government License. One of the major problems associated with the publication of legislative information is that one new act can have impacts on many others; however, these impacts are not necessarily global in scope and the changes many only effect specific paragraphs or clauses as opposed to replacing the previous act wholesale.

When planning the legislation.gov.uk website, the project team took the bold step of choosing to build a fully open access, dynamic portal based on linked data technologies; the linked data approach enables pages to be constructed dynamically using all the latest information on any given act, including the relationships to other pieces of legislation on the statute book. The RDF data for each act is completely open and free for all to access too, with users simply having to append /data.rdf to the end of the URL for any act. To help people use the linked data themselves (via the legislation.gov.uk API) the source code used for the portal has been made available on github, a public source code repository service (https://github.com/legislation/legislation).

In compiling the service, the National Archives have also adopted ideas from the open source movement and sourced expert assistance from outside of the core National Archives team as part of the Expert Participation Programme; this initiative has allowed the participants to submit corrections and changes to the linked data driving the portal, improving overall quality as part of a carefully curated process.

The linked data service is also driving exciting future plans, including the automatic tracing of provenance for legislation (a vitally important tool for law makers) and automatic determination of the impacts of new legislation via natural language processing of new acts (although this is at least in part made easier thanks to the very structured form of English used in legal documents).

4 The rail core ontology

The feasibility study demonstrators are based on an ontology model produced as part of a CASE funded PhD studentship between the University of Birmingham and Invensys Rail (now Siemens). Although initially developed with the representation of signalling and rail infrastructure in mind, the model rapidly developed into a general model for the railways, including a “core” of generic railway concepts with extensions capturing particular subdomains (infrastructure, timetabling, rolling stock etc.) and an upper level model to define concepts used more broadly than rail (all transport) The layered design philosophy behind the model is shown in Figure 3.

In order to construct the initial model, a significant amount of domain-specific knowledge was required; this was gathered in part from the analysis of existing rail data models (railML, Network Rail’s signalling data exchange format (SDEF) model, Invensys’ layout description language (LDL) model, InteGRail ontology) and the RSSB standards documents, alongside interviews with key individuals.

Drawing on queues from the ISO 15926 standard, two versions of the core ontology have been created: one that handles data in 3D, intended for use with real-time or high-volume datasets, and one that handles 4D (time-dependent) data, which is slower but more expressive, making it more suitable for the exchange of data between models.

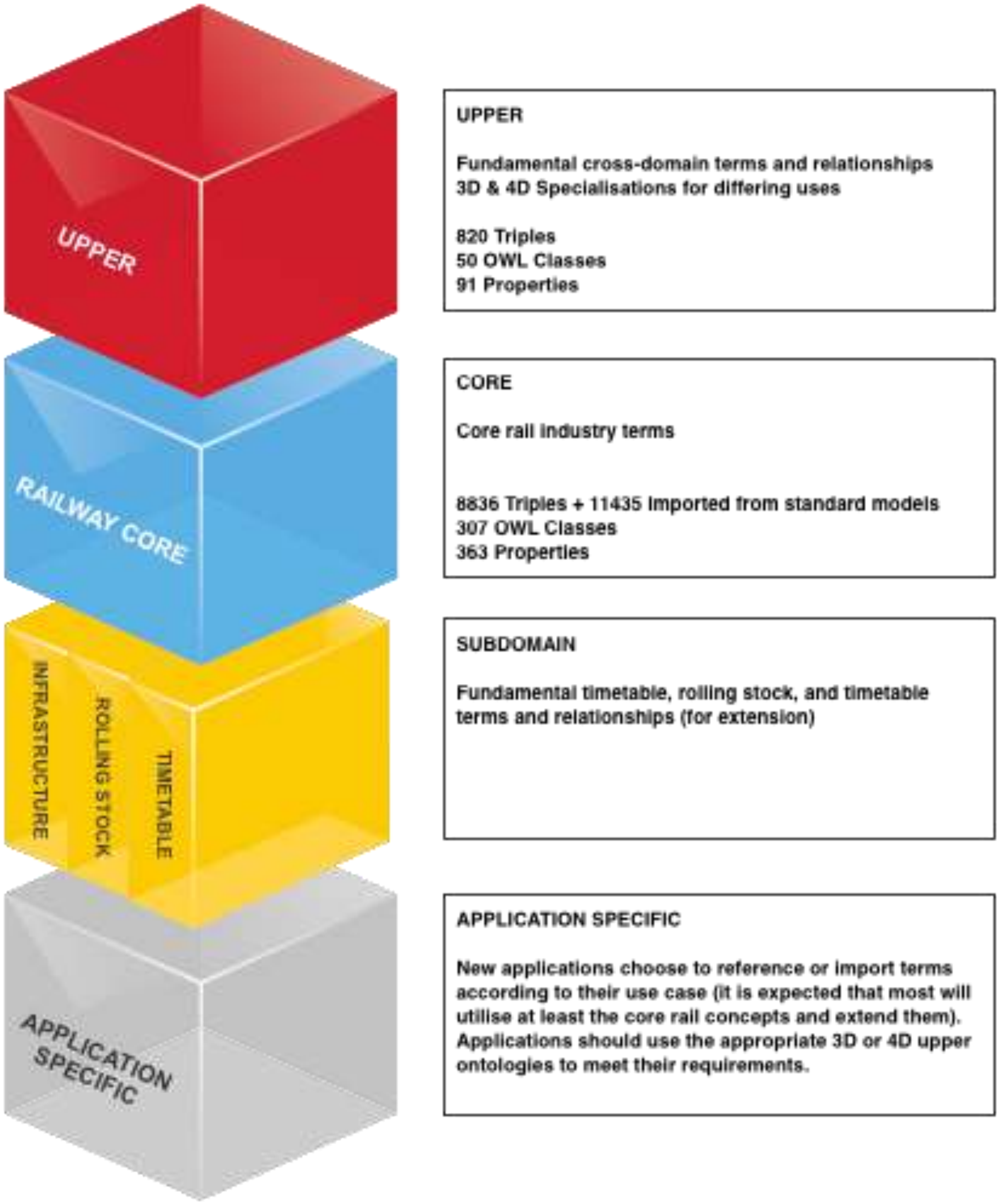


Figure 3: Layered design philosophy underpinning the core ontology.

5 FuTRO demonstration scenarios

In order to demonstrate the potential value of Semantic Web technologies to the industry, the feasibility study team have prepared two technology demonstrators looking at different aspects of the data integration / data sharing problem. The first, which is based around asset monitoring, uses the core ontology to manage generic asset information in a new monitoring platform being developed by Siemens. The second uses the model to illustrate how application functionality can be preserved despite system upgrades in the real world by allowing the appropriate information to be derived and delivered based on the available data and the application's requirements.

The demonstrators will be available from midday on the 18th July 2014 and can be accessed via:

- Demonstrator 1: http://purl.org/rail/futro/
- Demonstrator 2: http://purl.org/rail/trainlocator

5.1 Demonstrator 1 – asset monitoring

For the first demonstration, the project team were anxious to establish whether ontology, which has previously been shown to work in a railway research context, can be successfully used as a component of an industrial software system. A key element of this work would be proving that use of the ontology could add sufficient value to the system to offset the additional workload associated with implementing and maintaining the models.

The Siemens Asset Monitoring as a Service (AMaaS) prototype provided a promising test bed for the technology, combining a software package that is still sufficiently developmental for the data models to be integrated in the very short timeframe of the study, with a genuine business need on the part of the supplier (the need to build flexible, extensible data structures supporting a customizable, cloud-based monitoring platform for generic industrial assets). It is important to note that, while the project team have used the AMaaS platform for the demonstrator, the data models themselves are open and could be used by any industry supplier given the appropriate support (see section 6 - Exploitation and adoption plans).

5.1.1 Demonstrator 1: overview and key novel concepts

The AMaaS prototype was designed to address several key problems with existing asset monitoring and management systems. Three key technologies that are being demonstrated as part of this scenario are:

- **Linked Data** alleviates the need to create a single fixed data structure at system design-time, and provides the flexibility to specialise and extend system components easily, both for specific applications and customers, and for modifications into the future;
- **Ontology** provides a way of capturing and using domain knowledge to enrich railway data contained inside the system;
- A **Service-based Distributed System Architecture** allows the system to scale with volume and demand easily, creating better value-for-money and higher reliability.

5.1.2 Demonstrator 1: storyboard

Railway assets are currently monitored using a myriad of different approaches and systems, leading to many users only being able to access a subset of the asset information available to them at any one time. A selection of these information systems is shown in Figure 4. The diversity of asset types in most railway networks means that the one-off creation of a single asset monitoring platform for the whole industry is unfeasible, but what if it were possible to create a platform that could be adapted and extended over time? If such a platform could harness data from other systems and provide to other systems flexibly, significant benefits could be gained through data reuse and integration. The idea of a scalable, generic platform for asset monitoring is explored heavily in this demonstration, the scenario for which is shown in Figure 5.

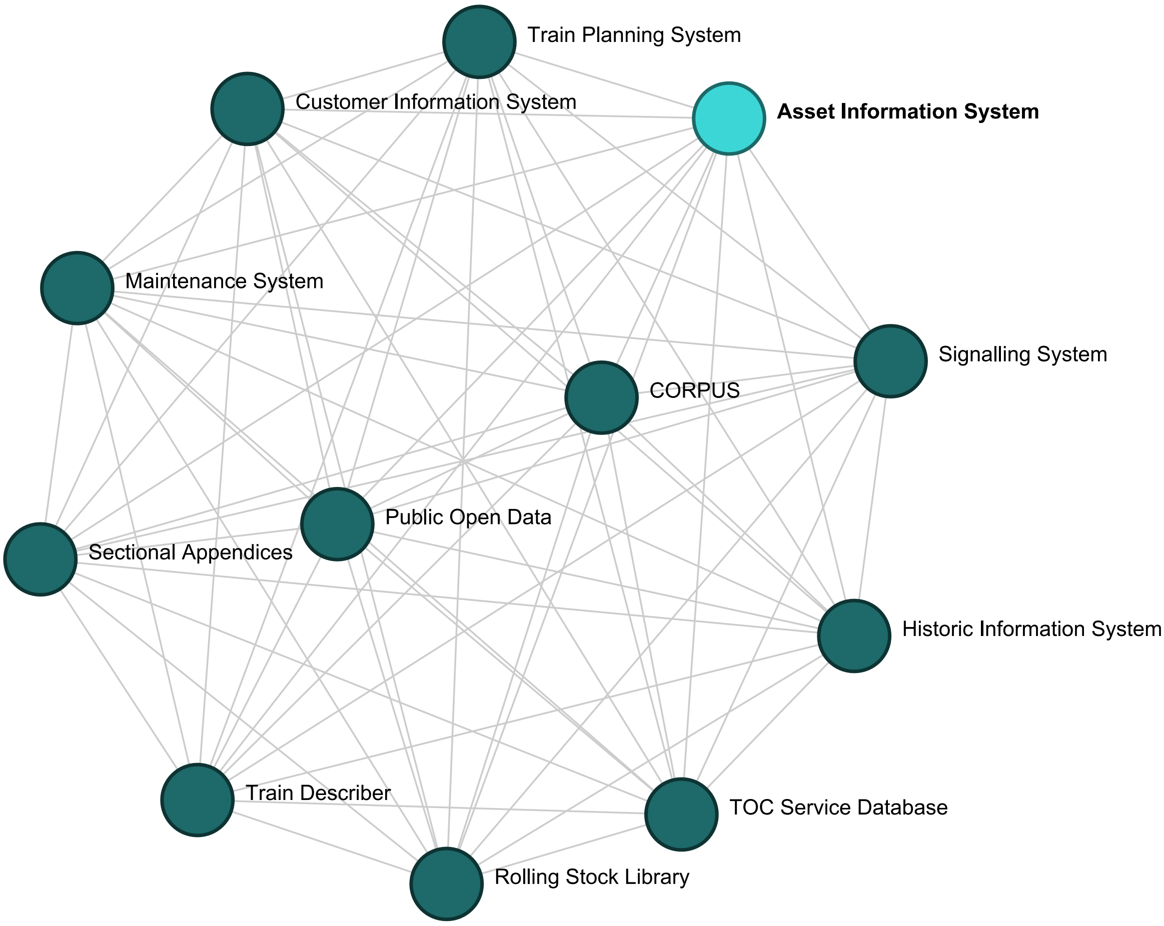


Figure 4: Integrated rail information systems.

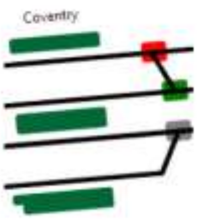
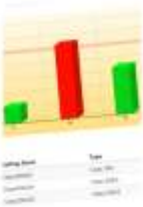
	Starting with a standard infrastructure monitoring application, several pieces of equipment monitor a piece of infrastructure, and each records the condition of one asset. These report back readings and generate either 'faulty' or 'healthy' verdicts based on internal data analysis algorithms. So far the system has no new functionality to that of traditional systems, but data is stored as a graph and aligned to an ontology. By using logic built-in to the ontology, it is possible to view conditions and faults from different points of view.
	Users view information in the Entity Browser, a view that captures all known information about a thing. Each real world thing is uniquely identifiable, and this view becomes richer as new data sources and associations are added. Asset register data is now brought into the model, and links created between monitoring equipment and railway assets. Asset lists are now enriched by historic fault information, and asset information now appears when browsing the system, with no change in application logic / code.
	Infrastructure information can also be brought into the model, in line with the rail ontology. Asset information is then associated with locations in the infrastructure, allowing users to view asset status on a trackmap, signalling diagram, or other view. Here a mock-up signalling view is displayed, with interactive assets highlighted in green. By using facts present in the ontology, diagnoses made in various monitoring systems can now influence the stored 'condition' of pieces of infrastructure – a points machine with a swing fault is inferred as being failed, and the infrastructure view updated. By extension, if enough is known about the infrastructure, the status of dependent components can also be known (for instance if a piece of track is out of action due to a signal failure, or a points machine relying on a faulty electricity supply).
	To illustrate the flexibility of the system, a Wheel Impact Load Detector device is introduced. Having made no initial provision for this piece of equipment, it is quickly integrated into the AMaaS platform. In order to correlate faults observed on the infrastructure (WILD) with faulty rolling stock, further data sources are brought into the system.
	Train schedule data is mapped from Network Rail CIF format into linked data, and rolling stock information added. This data aligns naturally with the infrastructure already present, and with the addition of a few statements to the ontology, it is possible to gather information on which trains pass across each piece of track, even when this is not explicitly defined (such as in the case of individual track sections).
	The 'train finder' view in AMaaS cross-references times stored for WILD fdetections with possible candidate trains. Ontology reasoning calculates matching characteristics by train, from rolling stock schedule information. Clicking one button, a user can associate a WILD event with a particular train, or even carriage. In a rolling stock asset managing application, this information can then be used to schedule maintenance. Given an appropriate matching algorithm, and further integration of train schedules, it is possible that the majority of WILD faults could be attributed automatically.

Figure 5: Storyboard presenting the scenario for the first technology demonstrator.

5.1.3 Demonstrator 1: technical solution

In order to reflect a real industrial use case as accurately as possible, the demonstration uses the prototype Siemens AMaaS system; a scalable monitoring platform designed to operate in the cloud, with assets self-reporting from their locations in the field. The demonstrator provides the following key functionality:

- Monitoring railway S&C assets and providing custom views for maintenance;
- Demonstrating deep integration of asset monitoring data with infrastructure data;
- Reporting of asset / network status based on monitoring equipment, domain knowledge, and custom logic;
- Demonstrating integration and enrichment of other, heterogeneous, datasets, through the example of including railway schedule data and rolling stock information;
- Presenting data to other information systems in the form of rail ontology-aligned linked data, allowing tight integration with other systems;
- Showing proof-of-concept of a distributed, cloud-based service architecture for railway information systems.

5.1.3.1 Introduction to the Siemens AMaaS system

The AMaaS platform (see Figure 6 for a schematic view) comprises of several key components:

- **Asset monitoring and data acquisition equipment** to acquire information about asset state;
- **Aggregators**, self-describing devices that process asset data and bring data from several assets into the AMaaS system;
- **The AMaaS Cloud**, which provides data storage and analysis functions, linked data endpoints for interfacing to other systems, and stores application state;
- **The AMaaS Web API & Front End**. Utilising information processed in the cloud, the AMaaS web application provides views on data present in the system. For this demonstrator, the front end provides views on infrastructure state, asset monitoring equipment status, and rolling stock.

The AMaaS system is designed to be scalable. Traditional barriers to system scalability can be split into two key categories; firstly, increasing data volumes put a strain on resources and require expensive infrastructure upgrades to keep up. Secondly, the variety of data types within the system increases – the scope of a system changes over time, and data structures developed to meet an initial brief become inadequate.

Through using cloud computing technologies and linked data, both of these problems can be mitigated, leading to a system that is scalable in terms of data volume, data scope, and system demand. As a customer’s asset monitoring requirements change, resources and functionality can be added or removed incrementally, avoiding the need for costly step upgrades, and increasing resilience. Aligning the data with a standardised railway ontology and making it available using linked data best practice allows other applications to make use and understand asset information contained in the model, without needing to know the specifics of every individual asset in the system.

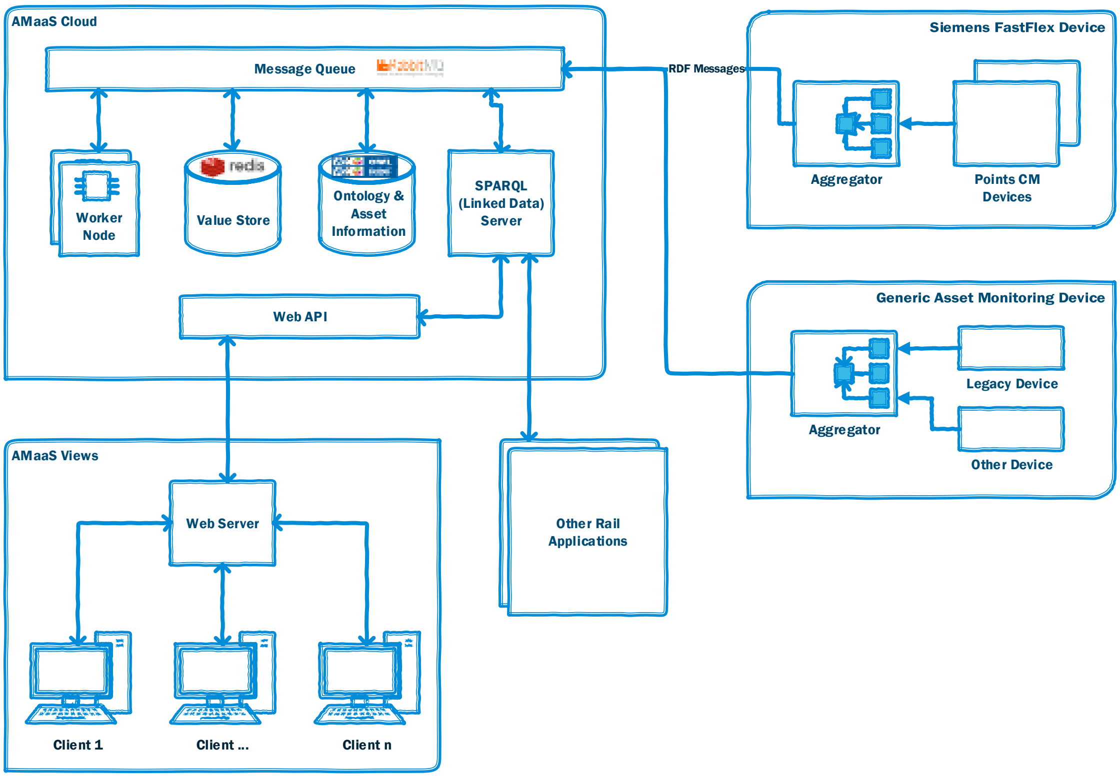


Figure 6: Block diagram of the high-level AMaaS system architecture.

5.1.3.2 Asset monitoring data flow and storage

The following briefly summarises how data is acquired at trackside AMaaS aggregator devices, added and enriched in the triple store, and then presented to users and to other systems.

1. On power-up, an aggregator device self-reports to the AMaaS cloud, using terms from the railway ontology, and creates a unique ID and description of. Through this identity, the data store can now infer additional information about it, for example its electrical characteristics or monitoring capabilities;

2. Other agents in the system may link additional data to the point monitor entity. In the story shown above, the identities and locations of asset equipment are linked together to enrich infrastructure information. Additional provenance data keeps track of which agent in the system added which pieces of data, and can be used for future decision-making;

3. When a measurement event occurs, the aggregator device sends another message to the AMaaS cloud, with its unique identifier, and an ‘observation’ object. Fine-grained measurement details (point swing sample data) is held in a separate data store, and referenced from the linked data;

4. Software agents on the AMaaS cloud listen for new measurement changes, and perform analysis on observations. For point machines, these include fault diagnosis, average current draw, and other characteristics;

5. Applications query the RDF triplestore for information when prompted. Unlike in a traditional database, the application need not know the exact syntax for retrieving information on all

different types of asset – a query for assets with unacknowledged faults will return anything matching that pattern, be it a points machine, a train, or a lamppost.

5.1.3.3 Using reasoning to automate system integration tasks

Wheel impact load detector devices are installed on a railway track to detect and record the wheel load and impact (for wheel flat diagnosis) of passing railway vehicles. Data acquired must be associated with one of a number of possible railway vehicles, a task traditionally carried out by either maintenance staff with local knowledge, or by radio frequency identification tags linked to an asset database.

However, by aligning available train running and schedule data to the rail ontology, it is trivial to form links between running trains and infrastructure. The demonstrator takes advantage of this by capturing knowledge used to map corresponding trains to their WILD observations, allowing users to view information by asset rather than by data source location.

5.1.3.4 Interactions between AMaaS and the open data models

AMaaS uses the core rail ontology developed at the University of Birmingham to enable data contextualisation and inference in a number of ways. The following section outlines how the demonstrator's key benefits are realised through the use of ontological models.

Asset Monitoring Identification and Asset Measurement Recording Using Core Ontology

In order to provide a working asset monitoring system, equipment must be classified in some way according to function. Using a traditional relational database approach, a table may be used, listing all asset monitoring equipment and associated attributes, and using database keys to provide means of linking measurements to sensors. Any additional context that may be available on certain devices is lost unless considered at the outset (creating very complex database schemas). Here, a simple approach using vocabulary from the rail ontology is used to create a hierarchy of asset monitoring devices, their sensors, and other information.

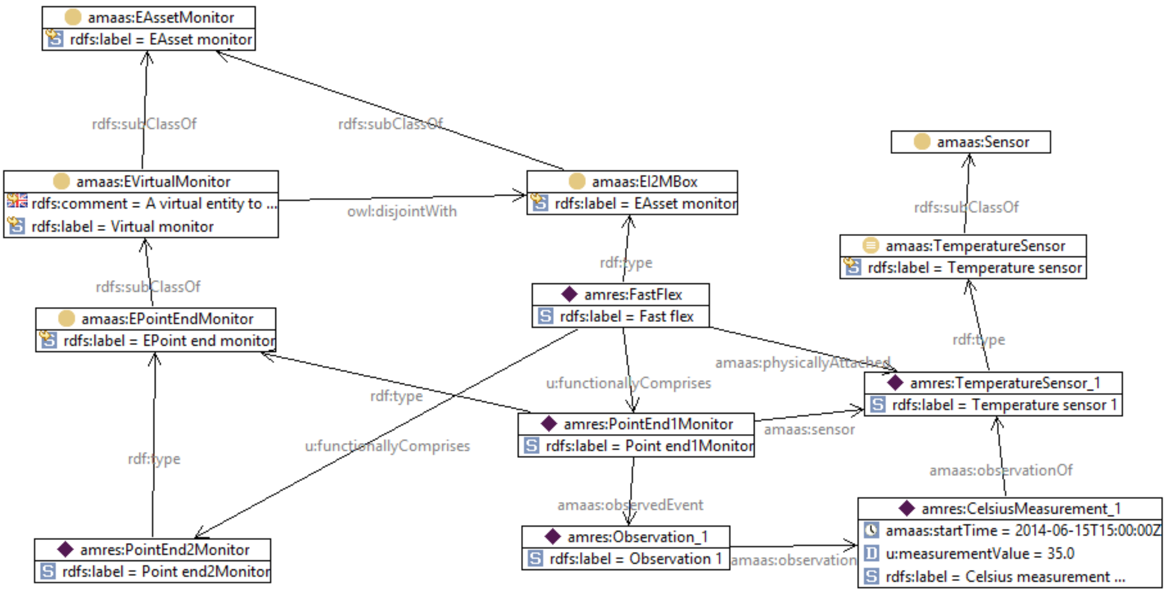


Figure 7: AMaaS point monitoring ontology design pattern (partial).

Using SPARQL, the RDF query language, it becomes possible to query for all EI2MBoxes (Fastflex points monitoring equipment), and by extension, all observations to do with them. If the ontology structure is known and defined, applications can be built without using reasoning at all. This is not always the case, however.

Asset Grouping And Integration Through Schema-Level Reasoning

By enabling ‘schema reasoning’, the most lightweight form of ontology reasoning, it is possible to gain a great deal of flexibility. The AMaaS ontology extension defines an ‘Asset Monitor’ class, which can be queried against to get the results of all railway asset monitors. In this way, associations can be built up that allow applications to continue functioning as new equipment is commissioned – an application relying on all ‘Asset Monitor’ devices located in the Midlands can continue functioning regardless of what types of devices are installed in the future.

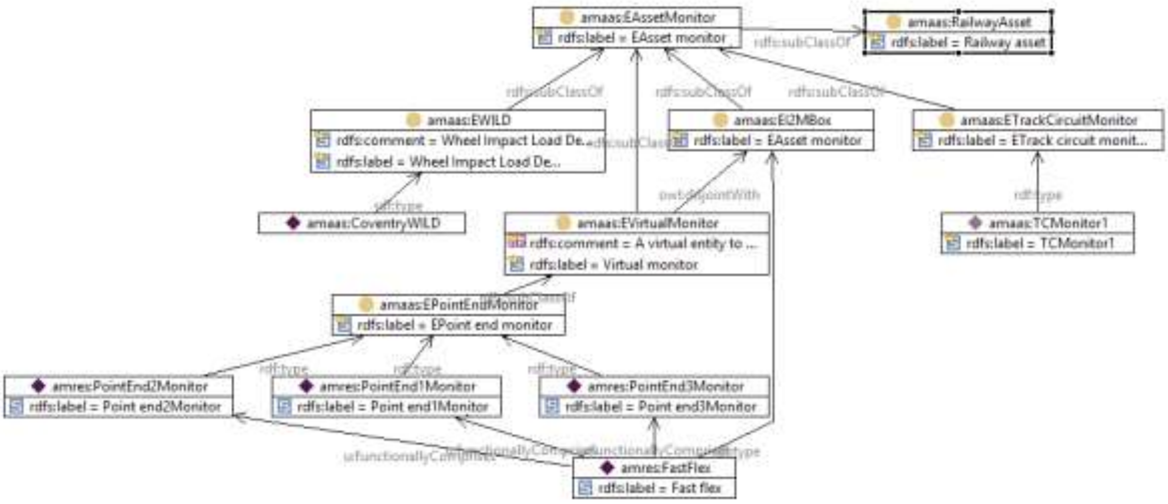


Figure 8: Ontology graph diagram showing schema-based type inheritance.

The railway core vocabulary ontology consists of many so-called ‘class trees’, and alignment with this ontology allows this structure-agnostic approach to programming applications.

Fault Attribution and Infrastructure Availability

Through use of slightly more advanced reasoning, it is possible to start exploiting further railway domain knowledge through use of the core ontology model. In AMaaS, entities are linked through particular relationships, and with reasoning enabled; these relationships allow “common sense” inferences across the data model and in all applications that use it. Here, this form of reasoning is demonstrated by the attribution of faults detected in monitoring equipment onto the infrastructure model:

1. An asset monitoring device is linked to a railway asset represented in the model through the ‘u:monitors’ relation. The semantics of this relation say that if A monitors B, any observation made by A is automatically linked to B. So if a track circuit monitor observes a fault, it can be said that the track circuit has an associated fault;
2. Assets are linked to their observations using the ‘u:associatedObservation’ relation. The semantics of this relation mean that when it is reasoned on, any fault diagnosis included in

the associated observation is automatically mapped onto the asset. As a result, when track circuit monitor A records a fault, this fault is directly associated with B;

3. The ‘u:dependsOn’ relation is used to determine which assets are dependent on others to operate. This property is transitive; if A depends on B, and B depends on C, then A depends on C. Consequently, if asset C is in a fault status, an application can immediately infer that asset A is faulty too. The track view in the AMaaS demonstrator uses this inference to provide a visual indication of which track sections and routes are unavailable due to points machine failure.

Whilst these relations are fairly naïve, and do not capture the real world semantics of how equipment depends on each other, they are designed to be extended for particular applications. If a new use case requires that detailed understanding of how assets interact functionally is required, the model and its relations can just be extended to allow this.

Rolling Stock and Infrastructure Integration

The AMaaS demonstrator’s ‘Train Finder’ view allows users to inspect wheel impact faults recorded, and allows them to associate these faults with particular trains. This functionality uses the core ontology for capture and mapping of train schedule data, and uses a graph query to match candidate train services to wheel impact load detector faults. A simplified version of the pattern used to capture train schedule data is shown in Figure 9.

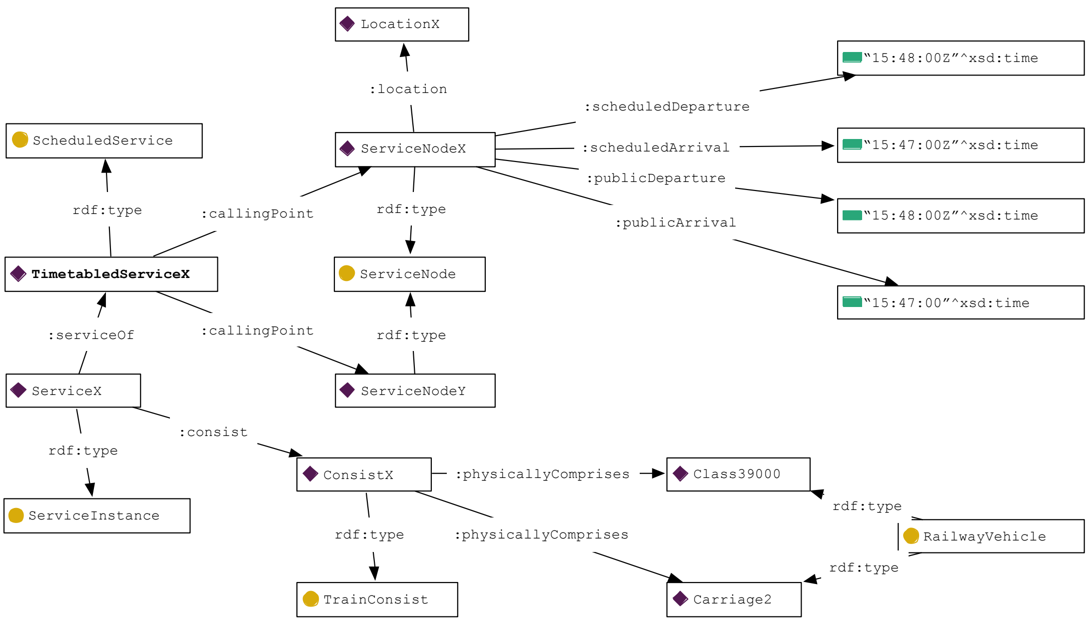


Figure 9: ATOC working timetable ontology design pattern (simplified).

Using the full version of this pattern, the most recent ATOC working timetable file was mapped into RDF, and an open source data reconciliation service used to find links from ATOC location codes to infrastructure data already available in the model. Whilst timetable data only provides locations at a relatively high level (stations and passing points), the infrastructure model present allows the association of these high level points with individual track sections and assets through reasoning. Thus, a query finding the services running past a location is reasonably simple (lines starting with a hash symbol are comments):

```
SELECT ?service ?servicelabel WHERE {
	# bind the observation we're looking at to a variable
	BIND(amres:WILDObservation1) as ?obs .
	# get the location of it's associated asset monitor
	?obs amaas:observationOf ?wild .
	?wild u:location ?loc .

	# find all the service nodes that go through the WILD's location
	?node a tt:ServiceNode .
	?node u:location ?loc .
	?node tt:ttOrder ?order .

	# resolve these nodes to potential train services, and return their labels
	?service a tt:Service .
	?service tt:serviceNode ?node .
	?service rdfs:label ?servicelabel}
```

By adding a filter (in the query language) to match a particular date range, we can find candidate matches. To simplify further, modern triplestores (such as Stardog used in this implementation) allow these queries to be stored as rules or magic properties, such that a user or application can instead send a much simpler query in their application:

```
SELECT * WHERE {amres:WILDObservation1 ?candidateMatches ?service .
	?service rdfs:label ?servicelavel}
```

This enables business logic to be stored with the data model rather than at applications. Rules concerning how the railway operates can be expressed in this way and utilised by anyone accessing the ontology, without any knowledge of what logic is being processed.

5.1.3.5 Outcomes of demonstrator

The demonstrator application, available at http://purl.org/rail/futro/, shows a number of web pages that demonstrate the tasks and functionality outlined in this section (section 5.1). The demonstrator web application is a customised version of the Siemens AMaaS front end and different scenarios are re-played by the server by allowing users to select one of a number of scenarios.

AMaaS Entity Browser

The entity browser (see Figure 10) allows a user to see information about any entity in the AMaaS system. It uses the AMaaS RDF federation service to combine ‘real time’ (replayed) point movement data with ontological/linked data about any asset or resource. Text in blue indicates a link or resource, and further information. Data is displayed in different ways depending on its data type, which is gathered from the triple store.

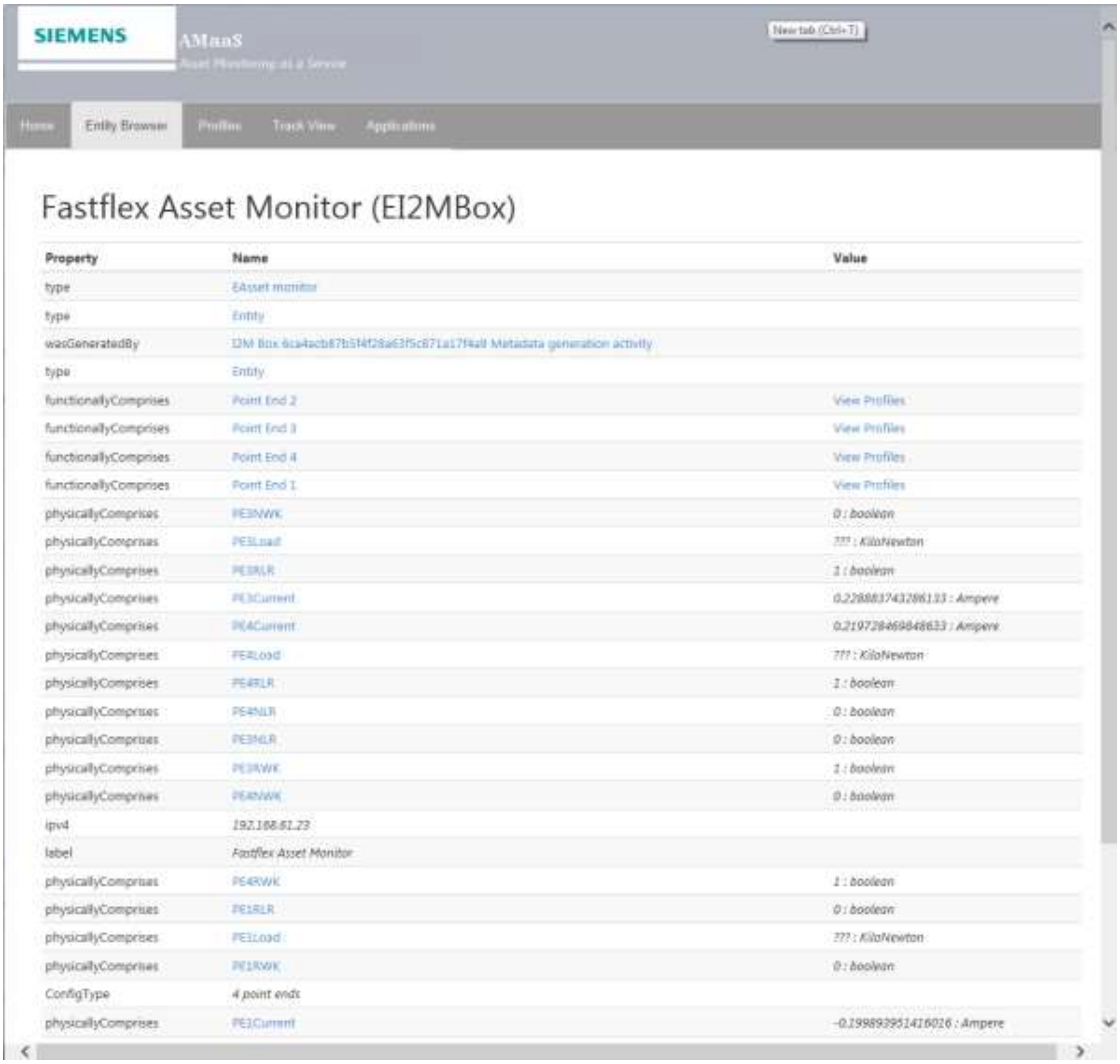


Figure 10: Screenshot of AMaaS entity browser showing data on a fastflex aggregator.

The data populated here heavily relies on ontology reasoning as described earlier. With no reasoning enabled, fewer links appear and the system behaves more like a traditional database application. As more information is entered into the system, more links between entities appear, and a richer experience is provided to users.

Track View

The Track View page (see Figure 11) shows a section of the West Coast Main Line around Coventry taken from Network Rail's National Sectional Appendix. Each piece of track and asset in the track view page is associated with an ontological entity, and the state of each entity can be observed. With basic reasoning enabled, the state of four point machines in the centre of the diagram can be observed, and further information gathered by hovering or clicking on the asset of interest.

As further rules and axioms are introduced into the system (by selecting the appropriate system state from the home page), unavailable or faulty assets are shown in red, and have an associated fault status inferred on them in the entity browser.

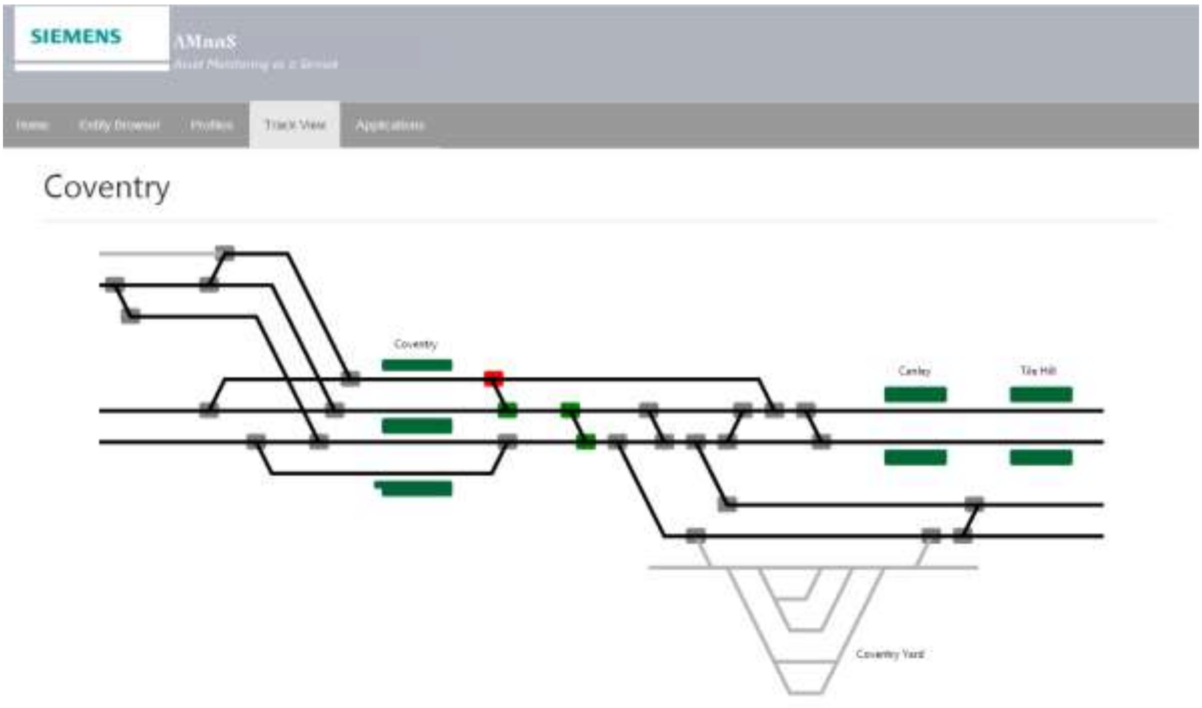


Figure 11: AMaaS track view.

Points Profile Viewer

The points profile view (see Figure 12) demonstrates the ability of an application to tailor views to suit different entities and scenarios. Any entity associated with points monitoring observations is shown with an option to view them, and the points profile view shown. Graphs are populated by calls through the RDF data store into a separate AMaaS database for finely grained data, and fault statuses for each point throw shown in red.

Figure 12: AMaaS points profile viewer.

Wheel Impact Load Detection and Train Finder

The WILD view (see Figure 13) is shown when data on wheel impact load detectors is requested, either through the entity browser, track view, an external application, or by clicking through menu options. With the relevant reasoning enabled from the home screen, the view shows WILD profiles, and highlights faults in red. The user is given the opportunity to select potential train matches for each observation, based on scheduled train services and rolling stock axle counts (gathered through rule reasoning).

Figure 13: Wheel impact load monitoring view.

As with the entire application, rolling stock have their own unique identifiers and are regarded as entities in the linked data store. The page for Pendolino 39005 is shown in Figure 14, with options to click through to retrieve information on each individual carriage.

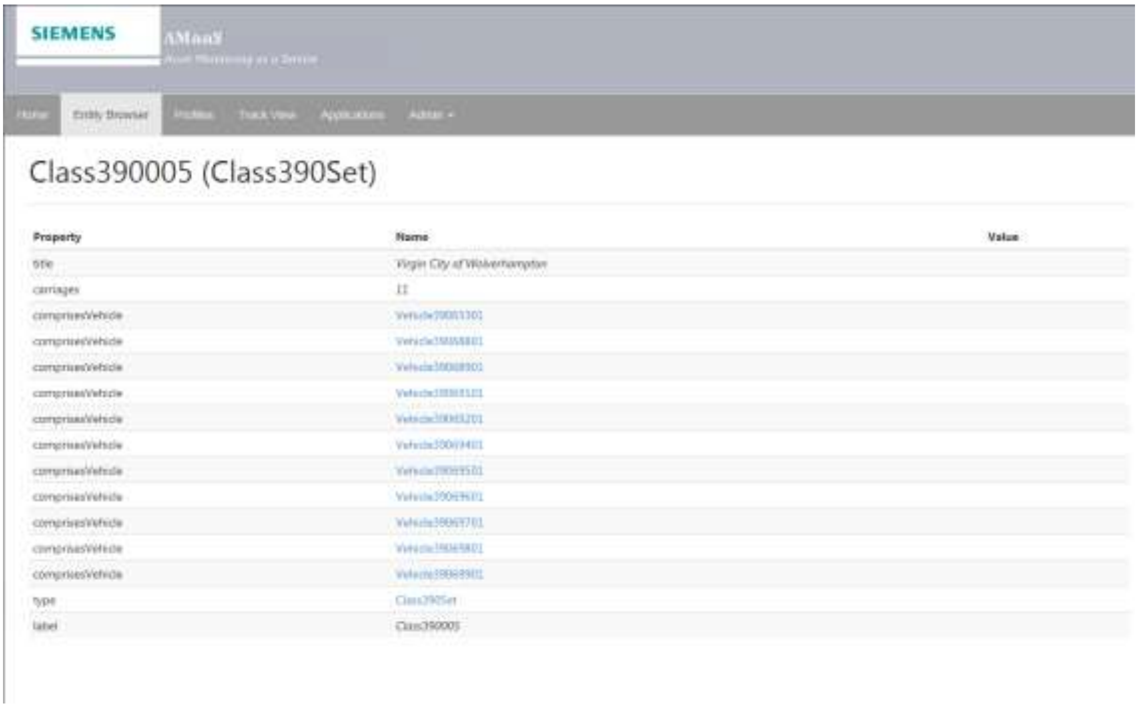


Figure 14: Entity browser showing imported rolling stock information

5.1.4 Demonstrator 1: discussion of benefits to the industry

The AMaaS demonstrator mostly seeks to show that application development using linked data is a viable solution for rail industry knowledge management. In creating an asset monitoring application based on the prototype rail core ontology, the benefits outlined above can be realised and seen working. However, the main benefits to any ontology-based system come not in its initial development, but in the longer term. These longer-term benefits are as follows:

Easy System Maintenance and Evolution

Upkeep of traditional information systems can be extremely costly, especially if modifications and upgrades are required. The AMaaS system demonstrated here effectively mitigates this issue by combining an extensible data model with scalable cloud-based implementation technologies. After commissioning, AMaaS can easily be modified, extended, or adjusted to include new features without any modification of existing ones. As more intelligent monitoring of assets takes place across the industry, the ability to expand easily is likely to make the system more effective and usable. The barrier to entry for new features is lowered significantly by reduced upgrade cost, allowing smaller systems to realise business benefits where investment would otherwise have been unfeasible.

Preservation of Data Semantics for Graceful Degradation & Future proofing

As systems progress through their lifecycle, technology and trends inevitably move on. Unlike traditional systems such as relational databases, the AMaaS system and the rail core ontology allow the semantics of all information used in the system to be absolutely preserved. RDF, the language linked data is based on, is fundamentally technology-agnostic, meaning that its information content will hold even as technologies progress and evolve.

Easy Data Integration with Other Information Systems

Use of a core ontology model here provides AMaaS with the opportunity to integrate deeply with other information systems. Whilst it is unlikely that the rail industry as a whole will base all their critical systems on RDF-based data stores, mapping technologies can pull data out of existing systems, align it with relevant ontologies, and make use of it inside AMaaS and other linked data applications. This was demonstrated in part by the mapping of ATOC timetable data in this demonstrator, but possibilities for integration exist across and outside of the rail domain.

Better Decision Making For Existing Purposes

As a monitoring system, AMaaS allows ontology rules to enrich information, as shown in the demonstrator above. Over time, the ontology and associated rules can be tweaked and modified to include more decision support mechanisms across a range of asset monitoring use cases.

Additionally, the graph-based structure of the system's data, and the core rail ontology, allow more advanced data analysis methods to be used in conjunction with the system itself. Emerging 'big data' methods such as map/reduce, as well as established graph analytics packages, can provide valuable insight into data contained within AMaaS as it progresses. These methods can be used to some advantage on traditional data stores, but the additional context present in an RDF store allow far more accurate analysis.

Greater Data Availability, Reliability, and Accuracy

The ability of AMaaS to integrate other systems easily provides incentives to bridge gaps between systems that have previously been used together by railway maintainers, signallers, and operators. The act of doing so immediately removes many data duplication issues, and greatly improves system accurately through the omission of any user input error by humans. As long as mappings between systems are maintained accurately, the context provided by AMaaS and its ontology provide users with more context around data they seek, enabling them to access all railway system data from one place, and coherently.

5.2 Demonstrator 2 – consistent passenger information provision despite technological change

For the second demonstration, the project team decided to illustrate how the use of ontology can help the industry maximise on existing investment in information systems despite technological changes elsewhere in the railway system.

The demonstrator sets out to show how the use of ontology can provide a bridge between legacy systems and newer replacement services without sacrificing functionality, and how interfaces between such legacy systems and more contemporary linked data-based systems can be set up. As the volumes and variety of data gathered in new information systems on the railway continue to increase, this demonstrator seeks to illustrate the practical uses of semantic data models in simplifying interfaces and applications, and enriching content.

5.2.1 'Train Locator' overview and key concepts

The web-based system presented here illustrates a few specific areas in which ontologies can allow better integration and management of data across subsystems. It focuses on the benefits that can be gained by using ontologies to unambiguously describe data to applications, and the ease with which new data in the system can be translated to accommodate existing applications. The following scenarios will be demonstrated:

- How new data (train location mileage information) can be quickly integrated into a data model given a new application or upgrade – in this case from track-circuit based location recording to radio-based mileage location recording;
- How ontology reasoning allows a legacy customer information system to continue functioning, even with loss of the initial track circuit location data;
- How ontology rules can be defined in the ontology to provide graceful degradation of functionality in passenger information systems.

In addition, the demonstrator application also illustrates the advantages of a linked data-based approach, showing fully contextualised content on every page, and allowing users to explore all relevant information available about a particular train, location, or schedule.

5.2.2 Demonstrator 2: storyboard

The storyboard for the second demonstrator is shown in Figure 15.

Imagine a rail network equipped with legacy, low resolution train positioning systems e.g. track circuit bays and axle counters.

The data produced by the train positioning systems is used to (amongst other things) power a number of passenger information systems, including platform boards and third-party applications for mobile devices.

As part of an upgrade programme, for example a migration to ERTMS, the existing low resolution train positioning equipment on a line is replaced by a more accurate system. Future passenger information systems can be designed to operate using the higher resolution positions from the new system but the existing passenger information systems, that require positional data to be at a track circuit level, will all need updating - a costly process that potentially involves many stakeholders if third-party applications are included.

In an information landscape utilising ontology, the data being delivered by the positioning systems and being used by the passenger information systems is described unambiguously; the computer "knows" exactly what data is available and what is needed by the applications.
Rules can be added to the data model describing how data in one form is converted to the other, allowing legacy systems to request track circuit level information as before and the system to deliver it (despite it not existing in that form) by inferring that it can be derived from the new ERTMS data.

By using the combination of ontology (to unambiguously describe the data available), rules (describing the conversion of data from one operational form to another) and reasoning, it becomes possible to maintain the functionality of existing applications, despite changes elsewhere in the rail system, without altering the applications' codebase.
Ontology will allow the industry to design and implement information systems once in a changing technological landscape. Old and new applications will be able to co-exist and can be driven by the same underlying data resources.

Figure 15: Storyboard presenting the scenario for the second technology demonstrator.

5.2.3 Demonstrator 2: technical solution

The second demonstrator is designed to showcase the benefits that can be gained through integrating data across a simple semantic data model using only a few very simple rules and ontological axioms. Two main applications are outlined, and ontology reasoning used to remove these applications' reliance on specific input data types.

The demonstrator itself, available at http://purl.org/rail/trainlocator, is a website that provides a number of views to simulate real world railway customer information systems. Each view illustrates a usage scenario, and the application is designed to allow users to understand the effects and advantages of differing ontology constructs on the system. Train movement data is provided by simulated values, which update the website in real time and drive outputs on each page.

5.2.4 Implementation system architecture

The key technological components used in the presentation of this demonstrator are:

- **Stardog**, an RDF triple store used to store all data (ontology and resources) in this demonstrator. Also used in the AMaaS demonstrator above, Stardog is a scalable web application that provides several levels of ontology reasoning – from the schema reasoning described above, to the ability to read custom-written rules. Stardog conforms to W3C standards on linked data storage and presentation, and can be used with all modern web technologies and programming languages;
- The **train movement simulator**, residing on the web server, which updates the locations of a set of trains as they pass through the demonstrator's railway network. Train locations are simulated through internal logic in this Java application, and pushed to the Stardog server through its linked data endpoint. Controls on the demonstrator website allow control over whether the simulator sends legacy (track circuit) or mileage train position data;
- A **web user interface**, written using modern web technologies (HTML, CSS, and Javascript). This web front end provides all of the application functionality, and queries the Stardog data store directly for each function. Logic in the web front end is limited purely to presentation details; all other information about interactions between trains, infrastructure, and location is stored and computed in the triplestore.

These three components communicate via the industry standard SPARQL linked data protocol, and data is exchanged in linked data at all points. Further input and output applications could quickly be realised by leveraging industry standard linked data practice and concepts shown in the core railway ontology.

5.2.4.1 Demonstrator Scenarios & Implementation

The web user interface shows information in any one of three scenarios:

- **Legacy Departure Board System (Using Track Circuit Data)**. In this scenario, a user can select a train station and view a very basic simulation of a platform-based passenger information board, including departure point, destination location, scheduled, and expected times. Expected times are calculated based on the position of trains on a track circuit (such as would be provided by a train describer system), which is queried directly from the triple store. The current track circuit of each train can also be displayed for exploratory purposes;
- **Train Position Map (Using Mileage Data)**. The train position map shows the 'live' locations of each train on the network. The system queries the ontology for mileage location, and displays it in line with the train's route through the network. Through rule reasoning, the ontology provides the train position map with the most relevant data should both be available;
- **Entity Information View (Using Linked Data and Inference)**. The final view is provided should a user want more information on a particular train, station, or location. The application requests information from the ontology about the location in question, and returns useful information. In the case of train services, inference provides information about the rolling stock itself as well as the train service; for locations, reasoning provides additional information such as touching/neighbouring entities and line reference information.

A summary of the behaviour of the ontology given differing applications and input data is shown in Table 1.

Table 1: Information sources for train locator application scenarios.

	Track Circuit Data	**Mileage (Moving Block) Data**
Legacy Departure Board System	Asserted ('real') track circuit data.	Inferred track circuit data based on train mileage.
Train Position Map	Inferred (approximate) train location based on known track circuit positions.	Asserted ('real') mileage data.
Train Position Map (When Both Sets of Location Data are Available)	Rule reasoning chooses optimum location object for the task.	

5.2.4.2 Key design patterns and reasoning devices

Infrastructure and Location Storage Design Pattern

Infrastructure and location data is stored in the train locator demonstration model as linked data, following the same core ontology terms and design patterns as in the AMaaS system. Data taken from ATOC working timetable files was used as a base for modelling train movements, and track circuits were added manually, using fictitious track circuit distances. Each 'Track Circuit' object has a start location and end location, each of which have an associated mileage and GPS co-ordinates.

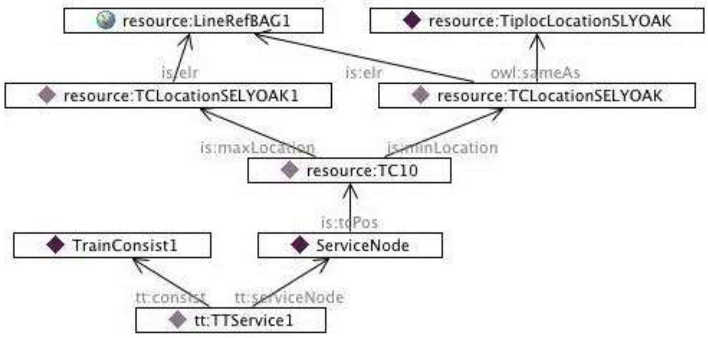


Figure 16: Ontology graph diagram showing track circuit positioning.

By linking track circuits to mileages and known pieces of infrastructure, inference can provide train services associated with them with further information. For example, in the case of a train stoppage or cancellation, passengers using linked-data based applications are immediately able to check the next station's facilities and connections based on the train they are currently on.

Reasoning to Allow Legacy System Functionality Given New System Input Data

In order to provide legacy system functionality when a system upgrade occurs, a rule is constructed and added to the triple store. Rules are custom-based reasoning patterns that a triplestore applies to matching data at query-time. The aim is to capture the following knowledge:

> *"If a train's current mileage is between the minimum and maximum mileages of a particular track section, and on the same line[1], the train is defined as being in that track section"*

A rule that encodes this knowledge, according to the ontology design set out above, is as follows:

```
IF {
  ?node a tt:ServiceNode .
  ?node u:location ?nodeloc .
  ?nodeloc is:elr ?elr .
  ?nodeloc is:mileage ?mileage .

  ?tcPos a is:TrackCircuitLocation .
  ?tcPos is:elr ?elr .

  ?tcPos is:minLocation ?minLoc .
  ?minLoc is:mileage ?min .
  ?tcPos is:maxLocation ?maxLoc .
  ?maxLoc is:mileage ?max .

  ?tc is:tcPos ?tcPos .

  FILTER(?mileage < ?max && ?mileage > ?min)
} THEN {
  ?node is:trackCircuit ?tc .
}
```

The code snippet works as follows:

- Checks for the current node's line reference and location;
- Narrows down a list of possible track circuits to just those on the current line;

[1] The logic required for this rule in the real world is slightly more complex; the demonstrator intentionally makes simplistic assumptions for reasons of clarity.

- Gets the minimum and maximum mileages for each candidate match;
- Narrows down the match further to only track circuits that contain the current mileage;
- Asserts that the current node is associated with the matching track circuit.

Consequently, whenever a legacy application now requests a node's track circuit location, this rule is checked and the correct track circuit returned whether it was encoded explicitly by an input system, or calculated based on a train's current mileage position.

Reasoning to Allow Improved Resilience of Information Systems during Degraded Service

The strengths of an ontology-driven data store do not only allow the mapping of new data back into other forms for use in legacy systems, but also make it possible to increase data availability during periods of degraded system reliability. Using the capability of the system to interlink data, a hierarchy of 'preferred' properties can be specified for each system concept, and these hierarchies used with closed world rule reasoning to find the best available data for a particular application. Take the following scenario from the storyboard shown above:

- A railway line has recently been upgraded to ERTMS operation, and now provides very rich location information for each train on the track, rather than only track circuit occupation details;
- New applications for customer information and service monitoring are built using the new, more accurate ERTMS location information. It is desirable, however, for these systems to continue functioning in times of degraded operations – for instance if ERTMS systems are unavailable and the line reverts to fixed block operation.

In this case, the usual approach would be to include application logic to search for available systems and make a decision specified at system design time as to which data source to choose – an approach which is inflexible and unsustainable in a complex system.

To enable the data model to find which data to provide for a train location application, the following pattern encodes knowledge of 'preferred systems' (see Figure 17).

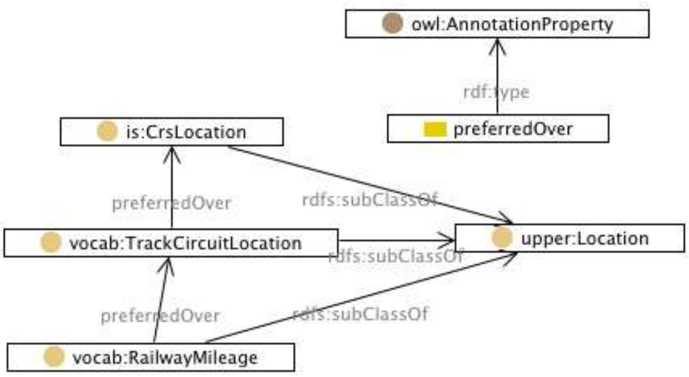


Figure 17: Ontology graph diagram showing 'preferredOver' relation between locations.

With this knowledge of which system of measurements is preferred given data availability, it is now possible to encode a rule that states:

"If entity X has multiple locations associated with it, and one is preferred (location Y) over the other (location Z), then insert a new fact: entity X -> preferredLocation -> location"

This is encoded in as follows[2]:

```
IF {
  ?s u:location ?p .
  ?s u:location ?q .
  ?p a ?class1 .
  ?q a ?class2 .
  ?class1 :preferredOver ?class2 .
} THEN {
      ?s :preferredLocation ?p
}""" ;
```

Systems now utilising this :preferredLocation property will automatically be presented with the most accurate data for their needs.

Note: applications have varying requirements for location data (some rely on GPS co-ordinates, others rely on CRS codes, and others on data with other constraints). The pattern above does not ignore these constraints; they are represented through other clauses in the query.

5.2.5 Demonstrator 2: outcomes

The web site for demonstrator 2 includes several views which show the effect of reasoning based on location, as discussed above. Some of these views are shown below:

Admin Page – Scenario Control

The Train Mapper home page (see Figure 18, page accessed when first visiting the site) briefly explains the aims of the demonstration and gives users control of the various scenario configuration options. These options influence the behaviour of both the legacy 'Departure Boards' view, and the 'Map View' application. Buttons on the left of the screen allow users to control the supply of data into the system: whether it receives track circuit data, position data, or both. The button on the right of the screen turns reasoning on and off for the entire web application – allowing users to see the effect with or without rules being triggered in the ontology.

[2] Code shown is implemented using Stardog Rules and is edited for clarity.

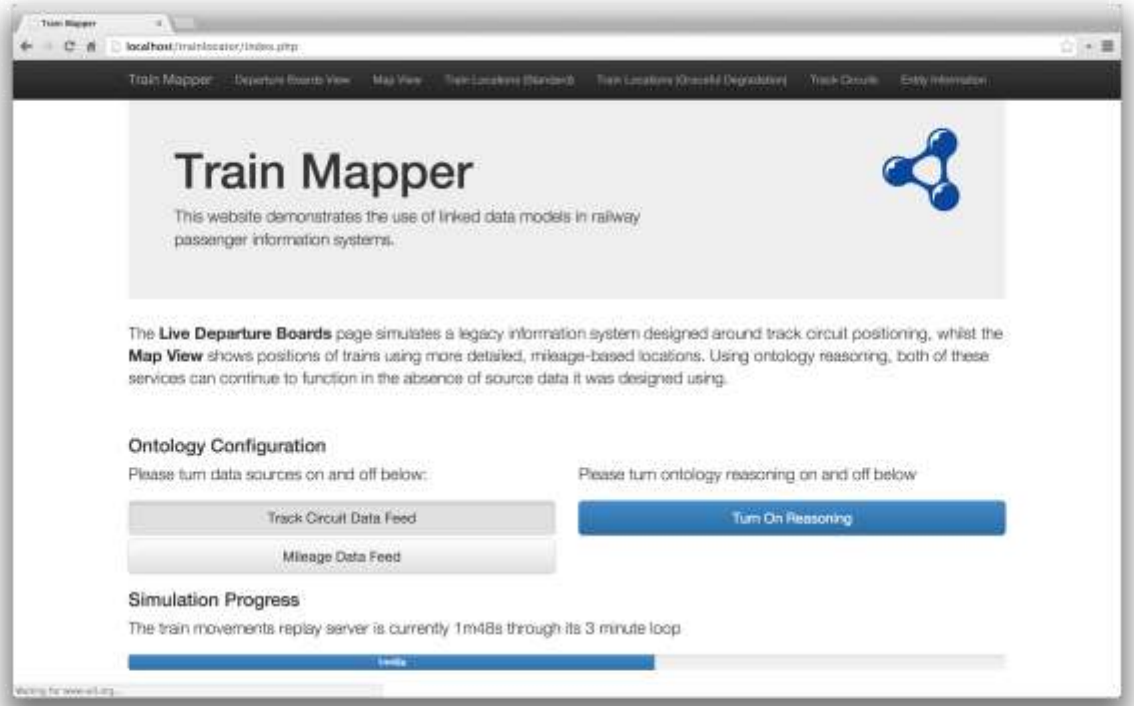


Figure 18: Train mapper home page.

Legacy Departure Boards View

The departure boards view (see Figure 19) shows trains soon to arrive and depart from a station. These are determined by querying the triplestore for relevant service nodes with an appropriate arrival time, and station information if present. Expected train times are naively obtained through adding a 'trainTime' property to every track circuit, and calculating the difference in this property's at the current train's location and the station being viewed.

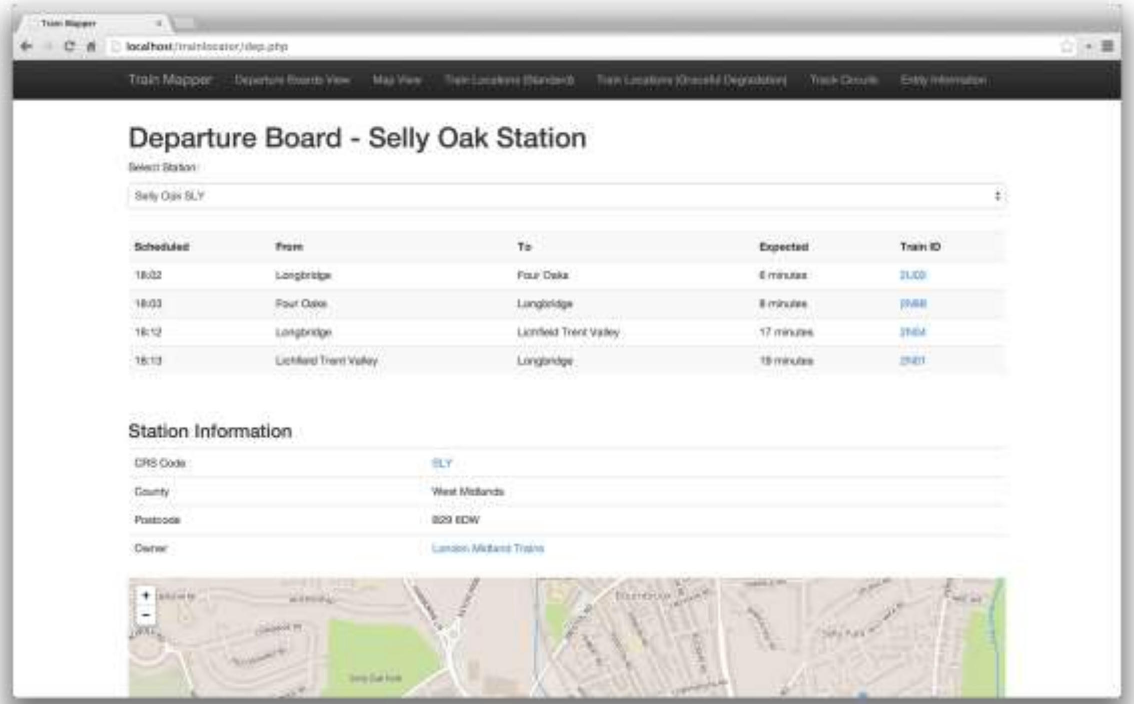


Figure 19: Train locator departure board view.

If the 'Track circuit data' data source is turned on, this view utilises no ontology reasoning whatsoever – it is presented as a legacy system using linked data as a data storage and interchange format. There are advantages even to this approach, as can be proven by the millions of websites utilising RDF & linked data on the World Wide Web. If the 'track circuit data' data source is missing, however, ontology reasoning steps in, and resolves live train locations to track circuits for the benefit of this application.

Map View – Dynamic Train Progress

The dynamic train progress page (see Figure 20) allows a user to track the progress of a train in real time, using mileage values resolved from a fictitious moving block signalling system. Users can select the train they want to track, and watch its position change across the map.

- With only the 'mileage data' source turned on, this display uses no inference and displays the current mileage of the train selected on a map;
- With both mileage data and track circuit data, this display calls the ontology to ascertain the priority of these location values (as described above), and displays the mileage location, with its track circuit displayed as a secondary information source.

With only track circuit data available, the ontology resolves a less accurate position for the train based on available information. Whilst it would have been possible to build this logic into the application itself, this approach quickly becomes complicated and hard to maintain when deployed as part of a more complex system.

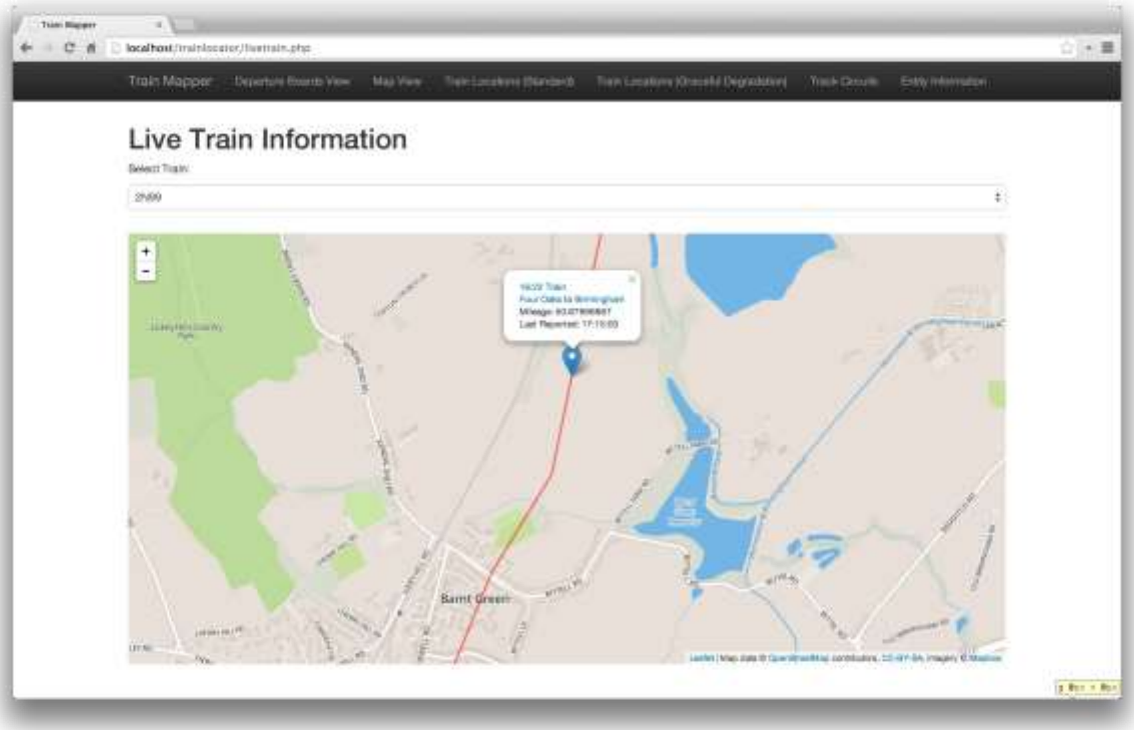


Figure 20: Live train information map view in train locator.

Map View – Track Circuit Information

Finally, the track circuit and entity views (see Figure 21) allow users to view more detailed information about each track circuit, or other entity. With reasoning disabled, queries used to populate this view bring back only explicit information held in the infrastructure database about track circuit information. However, with reasoning enabled, links between track circuit locations and other infrastructure items become apparent, and users are able to browse information about train stations, maintainers, and nearby trains. This view is included to further illustrate the use of ontology reasoning to enrich knowledge and convey useful inferred information.

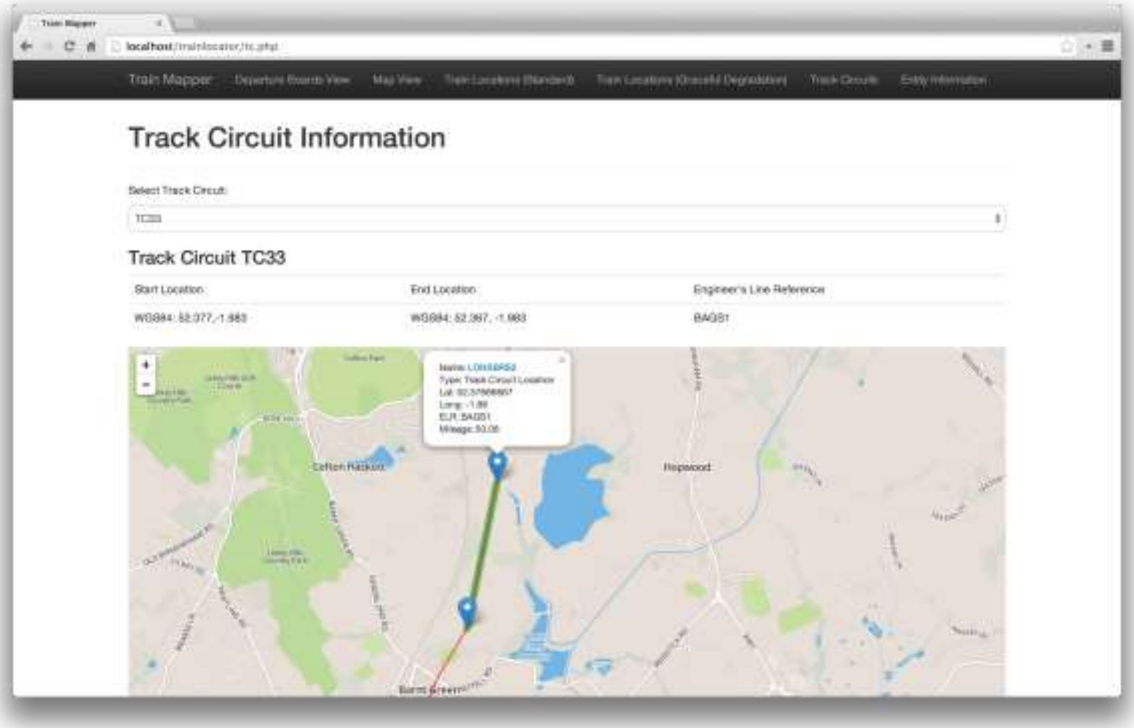


Figure 21: Track circuit detail and track circuit boundary overview screenshot.

5.2.6 Demonstrator 2: examples of alternative use cases for scenario and benefits gained

The demonstrator shown above was designed to illustrate how basic design patterns can show large potential benefit when implemented to integrate multiple information systems. As such the benefits seen here have a number of other use cases, a few examples of which are outlined below.

The design patterns and processes shown here have diverse applications across the railway; in fact the demonstrator highlights a fundamental technique that can be implemented wherever multiple systems provide the same type of information into a data store (whether as part of an upgrade or not). Across the industry, the idea of being able to pick and choose better information where or when it is available is an attractive prospect, and methods shown here allow an easy mechanism for not only providing this function, but also managing it in a way which is data-centric rather than application-centric.

Use Case 1: Railway operations management and train routing during degraded railway service

Whilst an ontology will not in itself evaluate decisions on train routing, the ability to provide data at whichever level of specialisation is available can inform human signallers and computer algorithms and help them to make operational decisions based on the most accurate information available at the time.

For Example: Two trains are waiting outside a major interchange station, and one platform is available. Which train is delayed depends mostly on a controller's intuition; using integrated data, puling in the most accurate data about each train could allow a more informed decision-making process. Ontology reasoning could infer typical train capacity in absence of it being known, or show *actual capacity if it is. Train connections from the following station could be displayed if known, or based on a rule if not.*

Use Case 2: Railway maintenance on tracked and untracked rolling stock assets

Using the same approach, plus knowledge of rolling stock composition (as provided by the infrastructure ontology), rolling stock maintainers can be informed of likely asset failures in absence of monitoring information. If a particular class of railway vehicle is known to develop a fault after a certain number of miles, it is possible for the ontology to display these likely faults on appropriate vehicles, and not to display them on vehicles with more detailed explicit information stored.

Use Case 3: Cross-Railway Train Position Reconciliation

The property translation pattern used to map mileage values into track circuit values is only one example of the ability of semantic data models to accommodate transition from legacy systems. Whilst ontologies cannot themselves provide very complex algebraic mappings from new systems to old (for example, geographic transforms), reasoning allows more common sense properties to be conveyed between systems with very little overhead. An example of this may be in resolving a problem encountered by open data enthusiasts when reconciling London Underground and Network Rail train movement data where the two systems overlap. Where multiple systems log the same information about trains in different ways, ontology rules and mappings can help to align data to be appear coherent. As these system interactions change, rules can be updated, and no change to application code is needed.

6 Exploitation and adoption plans

Late in 2011, as part of a Rail Research UK funded study into the potential for a reduction in overall CO_2 levels from the transport industry by encouraging a modal shift towards rail, the University of Birmingham team in conjunction with a team from the University of Nottingham proposed a developmental timeline for an ontology model for the rail industry (see Figure 22). This was later more widely published by Golightly et al. as part of a paper on the benefits of data sharing to the UK rail industry (Golightly, 2013).

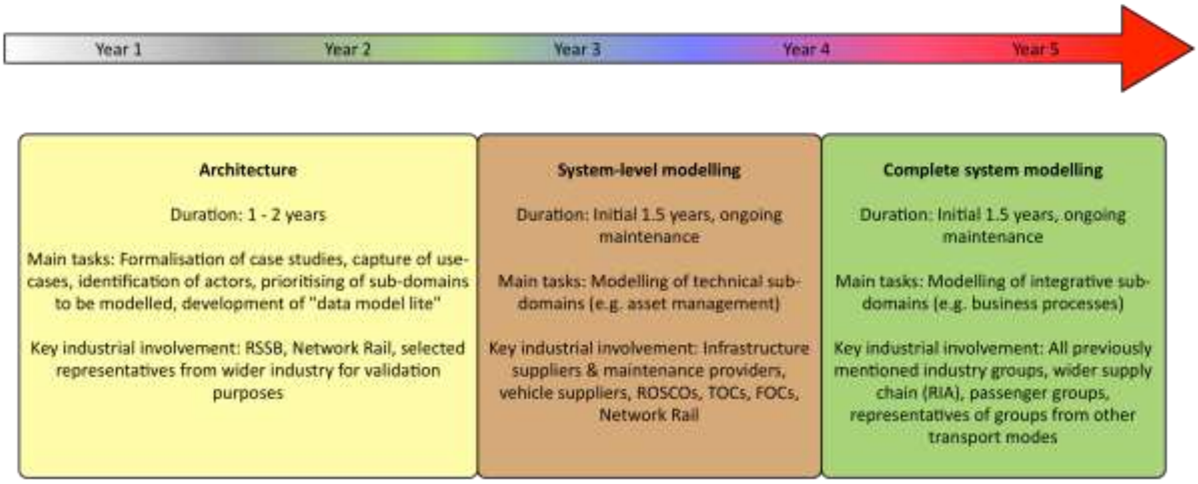


Figure 22: Roadmap describing the developmental steps needed to implement a conceptual data model for the UK rail industry (as taken from Golightly et al. 2013).

The timeline was split into three phases; architecture, system-level modelling and complete system modelling.

In phase one, which was expected to take around 2 years to complete, a core ontology model would be developed. The core model would take into account the key elements of, and actors in, the UK railway network and would be capable of describing the network without going into the specifics of any single system. Development of the core model has since been completed to schedule utilising an Invensys Rail / EPSRC CASE PhD studentship as the main funding source; it is this “data model lite” that has been used as a basis for the work presented in the feasibility study.

Moving forward, the team envisaged that as the work progressed through phases two and three of the timeline, the increasingly clear business benefits would mean that a greater proportion of the development costs could be met by industry. It was also expected that individual suppliers would be able to take the lead in modelling their own equipment, albeit with guidance from the development team where required.

One important element in this work will be the provision of more complex demonstration scenarios that illustrate how individual modelling efforts, such as those performed during this study, can be drawn together and used to inform the operation of the railway. However these will need to be supported by the production of training materials, guidance notes and code templates that can enable software developers working for industry suppliers to develop ontology extensions for their own products and systems.

Alongside new products being supplied to the industry, there is the question of existing assets and systems, and in particular which existing systems should be priorities for the potential development of APIs using the approaches outlined in the study. As part of the 2011 RRUK study, the UoB team asked an invited workshop of experts from across the industry about priority use cases for data sharing; the outcomes of that work have since been aligned with systems listed in the RSSB T962 National Information Systems Catalogue, and the results are shown in Table 2 (Note: this work was performed some months ago and the authors recognise that plans are in place to update / replace some of the systems listed).

Table 2: Non Network Rail owned industry information systems that may be involved in a range of operational scenarios.

Scenario	Key Information	Relevant Systems
Integrated, cross-interface condition monitoring	Headcodes, train formations, locations of assets, measured values and vehicle norms, vehicle maintenance schedule and history, timetable information	GENIUS, RAVERS, ATOC component tracking, B Plan
Operations planning	Information on routes, available train paths, measured and predicted capacity, availability of trains and crew, vehicle access, demand	GENIUS, RAVERS, RSL, B Plan, Journey Check, RMS

Increasing route utilisation	Infrastructure data (geometry, gauging, signalling), rolling stock parameters (speed, envelopes etc.), timetable information, demand forecasts, climate data	Journey Check, GENIUS, TOPS, RSL, B Plan
Maintenance planning	Vehicle component models, component availability, condition assessment and prognostic models	ATOC component tracking, RSL, Certifyx, RAVERS, PADS2000, GEMINI, GENIUS, IMACS, SLIMSTOCK/SCAAS, SLYX, TIGER
Supply chain data	Supplier database, installation dates, maintenance schedules, track information (design, geometry etc.), delay information, asset history, asset location, recorded incidents	Journey Check, GENIUS, RSL, Safety Management Information System, GEMINI, RAVERS, ATOC supplier assessment, ATOC component tracking, B Plan
Customer information	Timetable information, delay information, rolling stock characteristics (availability of seats, bicycle racks etc.)	Journey Check, GENIUS, RSL

7 Conclusions

Academic work on the potential applications of ontology in the UK rail industry has now been going on for around a decade, beginning with the EU FP6 InteGRail project. During this time the rail industry itself, in response to the findings of the 2011 rail value for money study (DfT, 2011), has identified both common standards for data (specifically ontology) and open architectures as important enablers of the future railway (TSLG, 2012).

The FuTRO Universal Data Challenge has given the feasibility study team an opportunity to establish whether or not the promising findings of research projects, such as InteGRail, into the topics of ontology and linked open data could be translated into tangible benefits for the industry; in particular by acting as an integration layer allowing the industry to “develop applications once” and as a means of accessing linked data from a range of data resources in response to disturbed operational scenarios.

In this report the feasibility study team have presented the arguments for the use of ontology in the UK railways, beginning with an outline of the key technologies and notations (OWL, RDF) before proceeding to look at mechanisms for securing models and data, and discussing the tracing of data provenance. Three case studies have been introduced illustrating how Semantic Web technologies have been used in other domains (oil and gas, retail and the public sector) and reflecting best practice and lessons learnt where available.

The team then present the two technology demonstrators, which focus on different aspects of the data integration and data sharing problem. The first, which is based around asset monitoring, uses the core ontology to manage generic asset information in a new monitoring platform being developed by Siemens. The second, uses the model to illustrate how application functionality can be preserved despite system upgrades in the real world by allowing the appropriate information to be derived and delivered based on the available data and the application's requirements.

Finally, the study looks at the next steps in the process; including the second phase of the model development timeline as published by Golightly et al. and the steps needed to make the modelling process itself more accessible to industry software developers.

8 References

Berners-Lee, T. (2000) – from slides presented to the XML 2000 meeting, 6 December 2000. Washington DC. Slides available at http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html last accessed 14th July 2014.

Berners-Lee, T., Hendler, James, & Lassila, O. (2001). The Semantic Web. Scientific American, 284, 34-43.

Brickley, D., & Miller, L. (2010). FOAF Vocabulary Specification. Retrieved October 5, 2011, from http://xmlns.com/foaf/spec/

Chandrasekaran, B., Josephson, J., & Benjamins, V. (1999). What are ontologies, and why do we need them? IEEE Intelligent Systems, 14(1), 20–26.

Chen, P. P.-shan. (1976). The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.123.1085

Denker, G., Kagal, L., & Finin, T. (2005) Security in the Semantic Web using OWL, Information Security Technical Report, Volume 10, Issue 1, 2005, Pages 51-58, ISSN 1363-4127, http://dx.doi.org/10.1016/j.istr.2004.11.002.

Department for Transport / Office of Rail Regulation. (2011). Realising the potential of GB rail: Final independent report of the rail value for money study (ISBN: 978 1 84864 124 2). Retrieved from http://assets.dft.gov.uk/publications/report-of-the-rail-vfm-study/realising-the-potential-of-gb-rail.pdf

Easton, J.M., and Roberts, C. (2013) "RSSB Cross-industry Railway Information Systems Workshop - Summary Report". Available from http://www.eee.bham.ac.uk/eastonjm/Dr_John_Easton/Publications_files/RSSBDataInitiativesWorkshopReport.pdf last seen 15 July 2014.

Golightly, D., Easton, J.M., Roberts, C., and Sharples, S. (2013) "Applications, value and barriers of common data frameworks in the rail industry of Great Britain." The Journal of Rail and Rapid Transit, vol. 227, no. 6, pp. 693–703, 2013.

GoodRelations. (2011). Prominent Users of GoodRelations. Retrieved October 5, 2011, from http://wiki.goodrelations-vocabulary.org/References

Gruber, T. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199–220.

Guha, R.V. (2013) – from slides presented to the 12th International Semantic Web Conference (ISWC), October 2013. Sydney, Australia. Slides and recording available at http://videolectures.net/iswc2013_guha_tunnel/ last accessed 16 July 2014.

Hepp, M. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In A. Gangemi & J. Euzenat (Eds.), Knowledge Engineering: Practice and Patterns (Vol. 5268, pp. 329-346). Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-87696-0_29

Kagal, L., Finin, T., Joshi, A., & Greenspan, S. (2006). Security and privacy challenges in open and dynamic environments. Computer, 39(6), 89-91.

Papakonstantinou, V., Michou, M., Fundulaki, I., Flouris, G., and Antoniou, G. (2012) Access control for RDF graphs using abstract models. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies (SACMAT '12). ACM, New York, NY, USA, 103-112.

Thuraisingham, B. (2005) Security standards for the semantic web, Computer Standards & Interfaces, 27(3), 257-268.

Technical Strategy Leadership Group. (2012). The Future Railway: The Industry's Rail Technical Strategy 2012. Available from http://www.futurerailway.org/RTS/About/Pages/default.aspx

W3C (2011) Describing Linked Datasets with the VoID Vocabulary. Available from http://www.w3.org/TR/void/ last seen 13 July 2014.

W3C (2013) PROV-Overview. Available from http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/ last seen 13 July 2014.

W3C (2014) Security. http://www.w3.org/standards/xml/security Available from last seen 13 July 2014.

RSSB Research Programme
Block 2 Angel Square
1 Torrens Street
London
EC1V 1NY

enquirydesk@rssb.co.uk

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