The objective of this work is to develop a novel ontology-based approach to address the complexity of the railway system. The Digital Railway programme, is an application of the proposed solution for producing requirements. This section introduces the reader to the key concepts used throughout this paper.

1 Introduction

1.1 Background and motivation
In the context of a continuous evolution of technology and business needs, it is becoming challenging for the railway to keep pace.

New ways of thinking and system design are essential. It becomes important to understand and design a system as a whole in its environment, including all of its dimensions: business, processes, people, and technology, and their interconnections. A whole system approach is necessary to develop whole system emerging properties that go beyond each individual system; whole system emerging properties are features that are meaningful to the whole system rather than its parts [1].

In addition, a formal and precise language for describing a whole system and its context is crucial to avoid the ambiguity intrinsic to natural language.

Finally, in order to be able to develop ever more complex systems in acceptable timeframes, it is fundamental to be able to design systems iteratively at different levels of detail simultaneously.

1.2 Objective
The objective of this work is to develop a novel ontology-based enterprise architecture framework that implements a whole system approach to address the complexity of the railway system.

The framework enables understanding and design of the key aspects of a complex system as a whole, alignment of solution/technology to business needs, and ability to assess the impact of changes and modify designs or requirements with minimal effort and cost.

1.3 Paper structure
The rest of this paper is organised as follows: Section 2 provides a definition of the main concepts used in this paper; Section 3 discusses the most well-known frameworks and formal languages available in the railway industry using the identified needs as assessment criteria; Section 4 provides an overview of the framework proposed, followed by a case study proving its feasibility in Section 5; finally, conclusions are drawn in Section 6.

2 Key definitions
This section introduces the reader to the key concepts used throughout this paper.

2.1 Whole system approach
A whole system approach considers a system in its entirety in the environment in which it operates (context). The single elements of the system are analysed in relation to the other elements with which they interact and as part of a whole [2] (coherence); the interconnections between all the elements of the whole system are actively considered [3] (connectivity). Finally, the same formal language is used across the whole system (consistency).

Whole system modelling is to be interpreted in this study as the modelling of a whole system approach. A definition of ‘model’ is provided in Section 2.4.

2.2 Systems engineering
In [4], systems engineering is defined as an iterative, top-down and interdisciplinary process that describes the static and dynamic behaviour of a system as a whole in its environment, so that the requirements for the system under consideration are met.

2.3 Enterprise architecture
As found in [5, 6], enterprise architecture is defined as a whole system approach to describe the dimensions of an enterprise, such as strategic, business and technology, with the objective of improving performance. In [7], a more theoretical definition is provided that is in line with the one given in [5, 6]. In more detail, enterprise architecture is defined as the ‘fundamental concepts or properties of an enterprise in its environment embodied in its elements, relationships, and in the principles of its design and evolution’.

2.4 Model
As defined in [8], a model is an abstraction of a system that focuses on the key aspects of the whole system. The details that, although important, are not relevant to the model are excluded. A model is, therefore, able to provide a whole system view, supporting the requirements formalisation process since the understanding of the system to be developed is improved. A model also enables the analysis of a system, ensuring that the final system has the desired properties. This leads to the specification of a complete and coherent set of requirements.

Different types of models are available. The choice of the most suitable model depends on the problem being addressed. In the work described, conceptual and logical models are used and
described by means of unified modelling language (UML) diagrams. These will be introduced later on in this paper.

2.5 Formal method

As described in [9], a formal method is characterised by rigorous syntax and semantics, where the syntax describes how a given system is built while the semantics gives meaning to each element of the system using a mathematical model.

Formal methods are useful for formalising requirements, supporting system design or verifying system implementations.

Formal methods enable a better understanding of formalised systems and a thorough verification of properties. They are easy to extend and also ensure that formalised requirements are fulfilled in the final output.

The remainder of this paper will focus on system requirements’ formalisation.

3 State-of-the-art

Railway signalling is considered to be one of the most fruitful areas for intervention by formal methods [10]. Therefore, the following section will focus on the literature from this field.

In [10], the available formal methods are compared in terms of conciseness and readability. Depending on complexity and constraints on a given system, different formal methods may be more suitable. For example, for train control, coloured petri nets (CPNs) and state charts have been used, while for interlocking, a modular specification method – European railway interlocking specification (ERIS) – has been defined.

Formal system requirements can be iteratively generated either from text-based requirement documents or from a conceptual system view. A combination of the two approaches is also possible. In the following two sections, the state-of-the-art techniques and languages for formalising system requirements are analysed.

3.1 Frameworks for formalising system requirements in railway engineering

The PhD thesis in [11] describes the developed parameterised model of the French computer-controlled relay-based interlocking system. The model is hierarchical, modular and configurable for different interlocking layouts. The model also includes an event-based concept to describe internal interactions. The current framework is capable of describing the static and dynamic behaviour of one specific system, a railway interlocking system. The ultimate goal is to develop a formal model for each sub-system that constitutes the new European train control system (ETCS) and integrate all the models into a global model. However, this goal has not yet been achieved. In addition, the approach is bottom-up within the context of a whole system approach; therefore, it is not capable of designing a whole system at different levels of detail simultaneously.

A similar approach is followed in [12] to develop system requirements for communication-based train control (CBTC) systems. It is stated that a bottom-up development strategy from sub-systems to the whole system simplifies the certification process of the overall system.

In more detail, the proposed process comprises three main phases. First, standards and architectural choices of the vendor are used to derive a consistent global model of the CBTC system, also called ‘the product’. The model is characterised by formal semantics and is represented as a modular diagram that hierarchically captures the architectural options. Then, an instance of the model, corresponding to the choice of an architectural option, is selected and a more detailed architecture for that instance is developed in the form of a block diagram describing system components and functionalities. The detailed architecture is used to develop scenarios that help analyse the product’s behaviour and manually derive natural language product requirements.

In the last phase of the process, the requirements for individual sub-systems of the CBTC product are defined from the detailed architecture and additional scenarios. In this phase, scenarios are developed to reach a better understanding of possible system usages. From each scenario, a set of system requirements is derived and verified. These are written in constrained natural language. An executable prototype is also built. This enables the identification of new scenarios and requirements.

The proposed approach can be adapted and applied to the development of the ETCS system requirements. However, the approach is characterised by the system (CBTC) rather than the whole system thinking. The latter is fundamental for developing a system as a whole, in all its dimensions.

Finally, the process described in [13] systematically integrates methods and techniques of requirements and systems engineering with project management ones for the concept phase of complex systems. It is stated that organisational and managerial aspects have a significant impact on the definition of requirements.

The proposed approach enables stakeholder engagement throughout the different phases of the process, leading to the definition of operational and technical system requirements that represent the stakeholders’ needs. Traceability to stakeholders’ needs and rationale behind each requirement is included in the final specification. The approach has been successfully applied in the concept phase of a defence system. The shortcoming of the approach is that it is mainly sequential; the need to design a whole system at different levels of detail at the same time is only partially addressed.

3.2 Formalisation languages in railway engineering

The formalisation of system requirements for the ETCS is mainly performed either in B language or using Petri nets [14]. In [14], a UML-based approach is proposed for modelling and partially verifying the ETCS specification; the approach is combined with B language to check the fulfilment of the safety aspects of the specifications.

A UML model represents the system and its environment as a set of objects that interact with each other. A UML model is able to provide static, dynamic and functional views of a system. Available UML toolkits are able to verify model consistency and remove redundancy within a model but they cannot check the satisfaction of safety requirements. A formal method of doing that is necessary. B language has been adopted in this work. B language represents safety properties as invariants. A system state is described by variables, which are modified by means of operations. The process of proof consists of showing that any operation on a system state respects a given invariant. This means that the system state under consideration fulfils the safety requirements for that state. The approach has been successfully applied to verify part of the ETCS specification.

Regarding the model itself, a UML model facilitates the understanding and design of a whole system, but it is not able to provide insight into whole system emerging properties because it lacks the support of iterative design.

As explained in [15], ‘CPN-tools’ is a piece of software that implements CPNs for system modelling and analysis. The study is focused on analysing the tool's ability to model the complete system requirements specification for the ETCS contained in Subset-026. A CPN model comprises two main parts: declarations and net. The declarations part contains variables and the definition of colours associated with each state, while the net is the system model. The system model consists of states, each one associated with a colour value; transitions occur between two states if the condition for the transition is met. Temporal and stochastic properties can be included in the model.

Analysis of CPN-tools has revealed the expressive capability of the software but also the difficulty in reading the resulting model as soon as the system complexity grows.

Therefore, a more formal model is proposed in [15] that summarises the transitions between different states in a more concise and humanly-readable format. The proposed model is automatically derived from a CPN one. Two additional models are proposed for representing conditions and temporal constraints. However, the modelling work of [15] is declared as still on-going; it is not possible to know how the proposed models would handle system complexity.
The work in [16] describes the translation of a subset of the ETCS specification into a formal model – a transition-system model – which can be automatically evaluated in terms of safety and interoperability properties. Cadence symbolic model verifier (SMV) is used for verifying properties formally expressed. Computational results show that SMV is fast, operating at less than one second on a dual-core processor with 3.06 GHz clock when it is applied to the verification of the properties that authorise the ETCS On-board sub-system to leave the TRIP (TR) mode state. For clarity, the ETCS On-board sub-system is in TR when it commands a sub-system to apply the brakes to bring the train to a standstill due to a number of possible reasons.

In [16], it is stated that an SMV model can be easily translated into other notations, such as automata or state machines, thus allowing different checking tools to be employed. However, limited information is provided regarding the formal model itself.

ERTMSFormalSpecs is an open source software tool [17], which formally models the ETCS requirements included in Subset-026 Version 3.6.0. The tool includes a railway signalling domain-specific software language, a tool chain, and a test environment. The tool has been used by the ERTMS User Group to manage change requests for the ETCS specification since it enables understanding of the impact of changes on the specification. However, little information is available on the domain-specific language due to commercial confidentiality.

RailTopoModel [18] is a logical object model aimed at standardising the representation of railway infrastructure in order to support rail business development. RailTopoModel can provide multiple consistent views of a given railway network at different levels of detail, depending on the needs and on data availability. Consistency is achieved by means of aggregation rules.

Although RailTopoModel is able to provide a whole system view of the railway system, it does not define the context in which the system operates.

Finally, the authors of [19, 20] propose ontologies as a formal language for describing a whole system. In [19], railway core ontologies (RaCoOn) was proposed for modelling the railway domain in-depth so that different views of the railway system at different levels of abstraction can be offered. The framework is modular, allowing only the required parts of the model to be used. The main drawback of the work lies in its low readiness level: only part of the domain is modelled. In order to overcome this, tools are provided that enable non-ontological experts to add data to the ontology-based model.

Instead, in [20], the major outcome of the European IT2Rail project is described: an ontology-based interoperability framework. The latter provides a formal, shareable, and machine-readable description of the domain knowledge for easier data exchange between the multiple, distributed and heterogeneous data sources. The scope of the ontology encompasses the planning, booking, and the realisation of a multi-modal journey. The framework would need extending in order to model the railway system as a whole.

The authors propose a new ontology-based framework since ontologies constitute a formal and precise language for describing a whole system, enabling them to understand and design a system as a whole in its environment, including all of its dimensions: business, processes, people, and technology, and their interconnections and constraints. Furthermore, the proposed framework supports the iterative design of a whole system at different levels of detail simultaneously, contributing to the development of whole system emerging properties.

To the best of authors’ knowledge, they believe that no similar approach has previously been taken.

In the following sections, the framework proposed in this study is described, together with a case study.

## 4 Proposed framework

The framework proposed in this study uses the concepts set out in [21 and specifically applies enterprise architecture in systems engineering [22] to provide a systematic, consistent and coherent approach to transform business needs into system of systems requirements. At the centre of the approach is an ontology-based whole system model (system of systems). A model-driven framework was chosen to deal with system complexity and to support system understanding and communication [23]. A model-centric approach also has the advantage of being able to manage the tacit or dynamic nature of needs by allowing different perspectives to be analysed and the model to be updated accordingly.

The ontology-based whole system model is used to generate requirements for complex systems, where the complexity lies in managing changes to business processes and levels of automation in order to meet business needs.

An ontology-based whole system model offers a number of advantages: firstly, a graphical or visual representation of a large complex system is easier to comprehend when compared to reading text-based documents. Secondly, an ontology-based whole system model allows the relationships and dependencies between different pieces of information to be identified and described, supporting the design of a connected and coherent system in its entirety. A more detailed description of the model is provided in the following section.

### 4.1 Ontology-based model

From the left in Fig. 1, stakeholder needs (1) originate from the business strategy. Business needs are derived from the stakeholder requirements with clear outcomes (2), the latter representing the expected results. A capability (3) defines the means to achieve the outcomes. Specifically, capabilities define the processes, functions, agents, and information (4), i.e. the system of systems solution.

The system of systems solution is the core of the model and describes how the overall system behaves and performs. In more
detail, functions are the steps that form a process, while agents execute functions in a defined sequence. An agent can either be a role (human) or an application (system). The model apportions processes to roles or applications based on the level of capability prescribed by the outcomes. The model is also able to describe the scale of automation, from completely manual to a fully automated system, showing how humans and applications execute functions.

Information is included in the model at a capability level; it is derived from functions and it is documented as a logical information model (LIM). Information requirements are obtained using the same method described for generating system of systems requirements in an information context. In more detail, capability processes are decomposed into functions. The information input or output for each function is modelled as a LIM. The LIM is used to describe the information requirements of the system of systems. The definition of information requirements can be carried out in parallel to the definition of the system of systems requirements; therefore, the whole system is iteratively designed at different levels of detail (i.e. system of systems and information) simultaneously, allowing the development of whole system emerging properties.

System of systems requirements describe relevant system behaviour and/or function, the performance of a function, and constraints on the design, whilst being technology agnostic. A requirement is not a pure description and can be ambiguous unless given meaning or context. System of systems requirements (5) is, therefore, documented as use cases, from which text-based requirements can be generated. A use case describes the interaction between agents and processes to achieve a given goal.

System of systems requirements become system needs (6), at each individual system level, and drive the system solution design (7) that generates system requirements (8). As shown in Fig. 1, at a system level, the LIM is populated with attributes and properties and, once it has been validated, it becomes a proven logical data model (LDM). The LDM in tabular form represents a data specification for the system solution. The data specification describes the data requirements for the processes involved.

The framework allows the implementation of only the necessary parts of the model, highlighted with numbers in the figure, accelerating the system readiness. Another advantage of using this framework is its ability to include existing system requirements as system needs. For example, requirements stored in text-based documents can be manually translated into the ontology format to become additional constraints within the model. When requirements are modified, the model is able to reflect the new requirements once it has been updated.

The framework also enables the traceability and provenance of a requirement to achieve a business outcome and strategy. Traceability facilitates the analysis of the impact that missing features in the whole system have on the satisfaction of the corresponding business needs.

The ontology-based model describes system behaviour statically and is encapsulated in a UML framework with specific extensions, a UML profile [24]. However, a more dynamic version of the model usable for simulation can be produced.

In Sections 4.2 and 4.3, the capability concept and its role are explained in more detail together with the scenario-based approach used for identifying capabilities.

4.2 Capability concept

A capability is an organisation's existing or potential capacity to achieve a specific outcome. As shown in Fig. 2, capabilities encompass processes, people, technology, and information. Capabilities are elevated above any organisational, solutional or technical silos. In addition, context is added to capabilities to provide a perspective and purpose. A coherent and connected view of the whole system is, therefore, provided in the context.

Capabilities are essential for aligning systems to business needs. Capabilities can be prioritised based on business strategic objectives, investment, and/or business planning decisions. Conventional mapping of systems and processes to business objectives is not effective as an excessive number of systems and processes are often required to form a meaningful and actionable system [26]. The proposed ontology-based model addresses this issue by describing the existing capabilities, allowing a comparison with future capabilities to achieve the expected outcomes. Therefore, changes to processes, people, technology, and information, necessary to develop capabilities that achieve the outcomes, can be identified.

4.3 Scenario-based capability identification

A scenario is a set of circumstances that are expected to cause the railway system to act in a particular way. A 'scenario sequence' shows the railway's response to a scenario, modelled as a UML activity diagram [27]. A UML activity diagram shows the order in which functions are executed.

Scenarios help to identify the capabilities needed to solve particular business problems, describing the current and future states using events, processes, and functions. They inform the system of systems requirements during the design phase. Scenarios can also be used to validate a Solution provided by a vendor during the testing phase.

The proposed framework has been applied to the formalisation of the system of systems requirements for the GB Digital Railway programme. The full case study is described in the following section.

5 Case study: formalisation of the system of systems requirements in a railway context

The GB Digital Railway programme is a transformational business change programme designed to modernise the rail network for the benefit of passengers, the freight industry, and the wider economy. A number of migration states have been defined to enable currently planned upgrades, as well as an incremental roll-out of new technologies to be performed gradually.

A whole system approach has been adopted and implemented in an enterprise architecture framework. An overview of the framework has been described in Section 4.

The framework has been used to formalise system of systems requirements at a migration state level.

The following sections describe in more detail how the framework enables the decomposition of the strategy into objectives for the desired migration state and provides a methodology to generate architecture and requirements that meet these objectives. A complete example is also provided.

5.1 Objectives definition

Objectives and corresponding outcomes are defined in the Digital Railway strategy [28] and Digital Railway programme business cases [29]. These objectives and outcomes are refined into the system of systems' needs. Other needs can be found in business needs, system definition, and concept of operation documents describing what is needed from a business perspective, what components the system should comprise and their related interfaces, and how the system should operate.
5.2 Capabilities selection

Capabilities are identified to fulfill the needs. In more detail, capabilities are selected from a generic existing model and enriched with information specific to the migration state under consideration.

Using scenarios (introduced in Section 4.3) specific to the migration state, the correct generic capabilities are identified and, if required, enhanced to reflect the continuous evolution of the operational environment (context).

5.3 System of systems solution

As shown in Fig. 3, the capabilities’ constituents introduced in Section 4.2, which form the system of systems solution, are developed. These include processes, functions, agents, and information. The necessary steps to define each constituent are described in the following subsections.

5.3.1 Definition of processes and functions: Processes are systematically organized into groups, each of which is associated with a set of functions performed by a discrete entity within the railway, such as a ‘train operator’ or a ‘station’.

Process grouping is further divided according to a time horizon. A time horizon is the period of time over which a decision or other piece of information is expected to have a direct impact on what happens. A time horizon can be strategic, tactical, daily, or immediate.

5.3.2 Agents: Agents represent the discrete entities within the railway that will perform processes and functions. Processes and functions are apportioned to specific agents.

In the example in Fig. 5, the management of a train's dwell is performed by a dwell manager. Agents can be roles or applications which indicate whether a human (role) or a computer-based system (application) is carrying out a function. In the example in Fig. 5, a platform steward is the dwell manager and is responsible for ensuring that the process of alighting from/boarding a train occurs seamlessly.

5.3.3 Information: As illustrated in Fig. 3, functions use information; this becomes knowledge when used to provide input (output, respectively) to (from, respectively) functions. The process of bringing information into the model is explained as follows. The capability under consideration is decomposed and described as a series of process steps or functions. These are documented as a dynamic network model that relates the process steps; the latter can be influenced by external factors or constraints. The resulting process model is linked to information entities. The process steps and information entities are validated and the network model becomes a proven LIM for the capability. Validation and verification nodes (e.g. clauses) are included in the network model as test nodes. This enables an iterative refinement of the model and the development of automated model validation and checking methods.

The LIM, with data attributes and properties, enables full traceability from business outcomes to system of systems data requirements.
Finally, the constituents of the system of systems solution can be developed in parallel to constituents of the system solution (modularity). For example, processes referring to the capability ‘performance management’ can be built in parallel to the LDM for the capability ‘movement control’.

6 Conclusion
An enterprise architecture framework is proposed. Its core is an ontology-based model that formally describes a system as a whole in its environment, considering each element of the system in relation to the other elements with which it interacts. The framework is, therefore, able to provide a coherent, connected and consistent view of a system as a whole in its context.

The framework is modular, allowing the development of the necessary components to achieve a given outcome, speeding up system readiness.

The framework also provides a model-based system engineering approach to requirement engineering, generating architecture and requirements that achieve the objectives and outcomes derived from a business strategy. More specifically, capabilities are identified to achieve the outcomes. Capabilities represent the means by which the outcomes are achieved. The framework is able to describe existing capabilities; therefore, it supports the analysis of the required changes to achieve future outcomes, enabling transformation.

Capabilities are refined into processes, functions, information, and agents: the ontology-based model. A key feature of the framework is its ability to apportion functions to roles (people) or applications (systems). The framework is, therefore, able to represent the digital evolution of a system, in which ever more systems will be computer based.

Information is aggregated data. Information becomes knowledge when used to provide input (output, respectively) to (from, respectively) functions. Information is included in the model by means of a LIM. LIM and enterprise architecture are integrated within the framework. LIM employs the same ontology-based model to decompose functions into data and functional requirements. Data is handled by design, enabling an accurate estimate of whole life cost for data management.

From the ontology-based model, formalised system of systems requirements can be automatically generated. Owing to its structure, the framework supports the traceability and provenance of a requirement to meet a business outcome and strategy. Consequently, the impact of missing features in the system of systems on the satisfaction of the corresponding business needs can be analysed.

The framework has been applied to the formalisation of the system of systems requirements for the GB Digital Railway programme. Additionally, it could be used throughout the whole life of a system.

Future work, already in progress, includes the development of a more dynamic version of the ontology-based model to be used for simulations.

7 References
[1] Checkland, P.: ‘Systems thinking’, in Currie, W.L., Galliers, B. (Eds.): ‘Re-thinking management information systems’ (OUP, UK, 2003), pp. 45–56
[2] Hitchins, D.K.: ‘Systems philosophy’, in ‘Systems engineering: a 21st century systems methodology’ (John Wiley & Sons Ltd, US, 2007), pp. 3–30
[3] Stasinopoulos, P., Smith, H.M., Hargroves, C.K., et al.: ‘Whole system design, an integrated approach to sustainable engineering’ (EarthScan Press, UK, 2009)
[4] Haskins, C.: ‘Systems engineering handbook: a guide for system life cycle processes and activities’ (INCOSE, USA, 2007)
[5] Scott, B., Grasso, J.: ‘A need for formalisation and auditing in enterprise architecture programs and approaches’, J. Enterp. Archit., 2009, 5, pp. 18–30
[6] Kotusev, S.: ‘Enterprise architecture: what did we study?’, Int. J. Coop. Inf. Syst., 2017, 26, (4), p. 1730002
[7] Hinkelmann, K., Gerber, A., Karagiannis, D., et al.: ‘A new paradigm for the continuous alignment of business and IT: combining enterprise architecture modelling and enterprise ontology’. Comput. Ind., 2016, 79, pp. 77–86
[8] Hull, E., Jackson, K., Dick, J. (Eds.): ‘Introduction’, in ‘Requirements engineering’ (Springer, US, 2005), pp. 1–18
[9] Peti-Duclos, M., Mentré, D., Guidemann, M.: ‘OpenTCS methods – definition of the methods used to perform the formal description’. Available at http://opentcs.org/tool/, accessed 29 September 2017
[10] Fantechi, A., Fokkink, W., Morzenti, A.: ‘Some trends in formal methods applications to railway signalling’, in Gnesi, S., Margaria, T. (Eds.): ‘Formal methods for industrial critical systems: a survey of applications’ (John Wiley & Sons, Inc., US, 2013), pp. 1–20

[11] Pengfei, S.: ‘Model based system engineering for safety of railway critical systems’. PhD thesis, Ecole Centrale de Lille, France, 2015

[12] Ferrari, A., Spagnolo, O.G., Martelli, G., et al.: ‘From commercial documents to system requirements: an approach for the engineering of novel CBTC solutions’, Int. J. Softw. Tools Technol. Transf., 2014, 16, (6), pp. 647–667

[13] Amaut, B.M., Ferrari, D.B., De Oliveira, M.L.: ‘A requirements engineering and management process in concept phase of complex systems’. Int. Symp. on System Engineering, Edinburgh, UK, 2016

[14] Kais, Z., Saltak, M., Schon, W., et al.: ‘UML approach for modelling and verifying railway signalling systems specifications’. 10th Int. Conf. on Modeling, Optimization and Simulation, Nancy, France, November 2014

[15] Barger, P., Schön, W., Bouali, M.: ‘A study of railway ERTMS safety with coloured Petri nets’. European Safety and Reliability Conf., Prague, Czech Republic, September 2009

[16] Ghazel, M.: ‘Formalizing a subset of ERTMS/ETCS specifications for verification purposes’, Transp. Res. C: Emerg. Technol., 2014, 42, pp. 60–75

[17] ‘ERTMSFormalsSpecs Software Tool’. Available at https://www.ertmsolutions.com/products/testing-optimization/lab-testing/ertosformalspecs-open-source, accessed 26 September 2017

[18] IRS30100: ‘Railtopomodel – railway infrastructure topological model’, 2016

[19] Morris, C., Easton, J., Roberts, C.: ‘From data to information: provision of railway data to passengers in the information age’. WCRR, Milan, Italy, May/June 2016

[20] Santoro, R., Lavillonner, E., Roberts, C., et al.: ‘The open web of transportation data as a shared networked environment for the realization of the single European railway area’. WCRR, Milan, Italy, May/June 2016

[21] Holt, J., Perry, A.S., Brownsword, M.: ‘Model-based requirements engineering’ (The Institution of Engineering and Technology, UK, 2012)

[22] Joannou, P.: ‘Enterprise, systems, and software engineering – the need for integration’. Computer, 2007, 40, (5), pp. 103–105

[23] Nikolaidou, M., Tsadimas, A., Alexopoulos, N., et al.: ‘Employing zachman enterprise architecture framework to systematically perform model-based system engineering activities’. 42nd Hawaii Int. Conf. on System Sciences, Big Island, HI, USA, 2009

[24] Alhir, S.S. (Ed.): ‘Extension mechanisms’, in ‘Guide to applying the UML’ (Springer, US, 2002), pp. 343–363

[25] Bhatia, D., Burbeck, M.: ‘Digital railway programme requirements management plan’. Internal Report, 2017

[26] Cassell, B.: ‘Building and putting your business capability model to work’, Enterprise Architecture Executive Council (CEB), 2012

[27] ISO/IEC19510:2013: ‘Information technology – object management group business process model and notation’, 2013

[28] Digital Railway: ‘Digital railway strategy’. Available at http://digitalrailway.co.uk/, accessed 9 January 2018

[29] Digital Railway: ‘Digital railway programme business cases’. Available at http://digitalrailway.co.uk:2016/12/07/driving-towards-digital-railway-deployment-establishing-investment-route-route/, accessed 9 January 2018