School of Electronic, Electrical and System Engineering

University of Birmingham

Whole System Railway Modelling

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Author: Garry Patrick Greenland

Supervisor(s): Professor Clive Roberts and Dr. Stuart Hillmansen

Date: 16th April 2019

Abstract

There has been a general view articulated within the railway industry that there needs to be greater systems thinking and systems engineering applied to major projects within the industry (Network Rail, 2013 and Rail Safety and Standards Board, 2012). However, there are many differing ideas held by practicing engineers of exactly what systems engineering is and how it is applied within the industry. There are also barriers within industry in general, management and practicing engineers to using systems engineering techniques. They can be seen as an overhead in terms of, training, tooling, effort and costs. Also the benefits to be gained from applying these techniques are not easily seen when they work well.

A key pillar of systems engineering and systems thinking is the ability to look at a system as a whole. Part of this is getting to grips with what a system really is, it's interaction with its operational environment and the world around it and to understand the various subsystems that the system is comprised of and their interaction, including people. This is particularly difficult when it comes to complex systems like railways.

This project attempts to develop an approach to modelling a whole railway system (or Guided Transport System (GTS) as it is defined in this project) by implementing a Model Based Systems Engineering (MBSE) approach and techniques. It also proposes definitions of a system and system engineering that are applicable to the Railway industry. Through a common view of a GTS as a whole and a common approach to modelling it, it should be possible to address some of the barriers to systems engineering techniques that currently exist.

MBSE has three pillars, a method, a modelling language and a modelling tool (Delligatti, 2014, pp. 4-7). The author has developed a method that can be applied to a whole complex system,

such as a GTS, supported by the SysML modelling language implemented through the Enterprise Architect modelling tool (other languages and modelling tools could also be used).

The method developed was then tested on a body of students studying for an MSc in Railway Systems Engineering and Integration at the University of Birmingham. This body was chosen because the course is part time and the majority of the students work full time in the industry. Thus the author was able to gain an insight into how diverse the opinions on systems engineering and its application actually are within the industry and get valuable feedback on the systems modelling methodology developed during this research.

It has been demonstrated through the development of a partial model of various representative parts of a GTS, that it is possible, within a single model, to capture and represent a large and diverse amount of information about a GTS as it is defined within this thesis. This includes:

- its context within the wider world and its operational environment;
- its physical structure;
- the relationships between its various subsystems and the outside world;
- the views of a diverse stakeholder group and their Requirements; and
- critical system properties and how these are derived from the various layers of abstraction within the system.

The methodology drives the user to develop a model that:

1. is re-usable, e.g. applicable to different railways at different times;
2. is extendable in length (be able to model more railway) and depth (greater levels of detail);

3. allows the inclusion of existing quantitative and qualitative models from other sources;
4. encourages the use of data from existing sources;
5. is open and transparent to allow others to use and add to them; and
6. enables the production of outputs that are readily understandable across disciplinary divides e.g. common representation.

This Thesis is dedicated to:

my wife Jenny, without whose support, through many years of part time study and learning, this thesis could not have been written.

Acknowledgements

There have been a large number of people who have encouraged and supported me through many years of part time and distance study to get to this point. Mr. Phil Bentley and Mr. Graham Preston both depot engineers at Stewarts Lane Depot, Battersea, without whose support going back to college to complete my ONC/HNC and first degree would not have been possible, and Mr Lynton Washer my shift manager whose encouragement at the start was invaluable.

Also Mr. Paul Thomas, Engineering Director at Parsons, now at London Underground, who's support secured financial support from Parsons.

Andrew Pearson and Richard Birkhead who allowed me to carry on this work when I moved to Lloyds Register Rail, now Ricardo Rail, without whose support this thesis could not have been finished.

My academic supervisors Professor Clive Roberts and Dr Stuart Hillmansen for their sound advice and support through the more difficult times. Also Dr Joe Silmon for his help with SysML and TRAK.

Finally, I owe a debt of gratitude to my colleague Dr Daniel Woodland, whose help and support throughout the last few years have been invaluable.

Table of Contents

List of Figures

List of Tables

Glossary of Terms / List of Abbreviations

Acronyms

Acronym	Explanation / Meaning / Definition
AC	Alternating Current
ATC	Automatic Train Control
ATO	Automatic Train Operation
ATOC	Association of Train Operating Companies
ASAP	As Soon As Possible
BDD	Block Definition Diagram
BPMN	Business Process Model and Notation
BSI	British Standards Institute
CAPEX	Capital Expenditure
CBTC	Communications Based Train Control
CENELC	European Committee for Electrotechnical Standardization
COTS	Commercial Of The Self
DC	Direct Current
DfT	Department for Transport
DL	Description Logic
DMU	Diesel Multiple Unit
DoDAF	Department of Defence Architecture Framework
EMI	Electromagnetic Interference
EMU	Electrical Multiple Unit
ERTMS	European Rail Traffic Managements System
ETCS	European Train Control System
EU	European Union
EVC	European Vital Computer
FRS	Functional Requirements Specification
GSM-R	Global System for Mobile Communications – Railway
GTS	Guided Transport System
HABD	Hot Axle Box Detector
IBD	Internal Block diagram

Acronym	Explanation / Meaning / Definition
INCOSE	International Conference On Systems Engineering
IPR	Intellectual Property Rights
ISO	International Standards Organisation
KADS	Knowledge Acquisition and Document Structuring
km	Kilometers
KPI	Key Performance Indicator
LUL	London Underground Limited
MBE	Model Based Engineering
MBSE	Model Based Systems Engineering
MoDAF	Ministry of Defence Architecture Framework
NASA	North American Space Agency
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NR	Network Rail
NSW	New South Wales
ODSS	Operations Decision Support System
OMG	Object Management Group
OOSEM	Object Oriented Systems Engineering Modelling
OPEX	Operational Expenditure
OPM	Object Process Methodology
ORR	Office for Road and Road
OSI	Open System Interconnection
OWL	Web Ontology Language
RailML	Rail Modelling Language
RCM	Reliability Centred Maintenance
RDF	Resource Description Framework
RDO	Rail Domain Ontology
RFA	Rail Functional Architecture
RFG	Rail Freight Group
RIA	Rail Industry Association
RSSB	Rail Safety and Standards Board
RUPSE	Rational Unified Process for Systems Engineering
SE	Systems Engineering
SRS	System Requirements Specification

Acronym	Explanation / Meaning / Definition
SSUP	Sub Surface Upgrade Project
SOAP	Simple Object Access Protocol
SWRL	Semantic Web Rule Language
SysML	System Modelling Language
SYSMOD	System Model
TCMS	Train Control and Monitoring System
TMS	Train Management System
TOVE	TOronto Virtual Enterprise
TSI	Technical Specification for Interoperability
TSAG	Technical Strategy Advisory Group
TSLG	Technical Strategy Leadership Group
UPDM	Unified Profile for DoDAF/MoDAF
WILM	Wheel Impact Load Monitor
XML	eXtensible Markup Language
XSLT	EXtensible Stylesheet Language
UK	United Kingdom
UML	Unified Modelling Language
UPDM	Unified Profile for DoDAF/MoDAF
US	United States

Terms

Term	Explanation / Meaning / Definition
Actor	A human or external system that interacts with a system of interest
Action	The unit
Activity	basic building blocks of Activities in SysML
Activity Diagram	Behavioral diagram in SysML
Activity Parameter	Input to a Call Behavior Action is SysML
Activity Partition	Method of allocating behavior to structure in SysML
Block	The basic unit of structure in SysML
Control Token	Imaginary unit of control in an Activity Diagram

Term	Explanation / Meaning / Definition
Decision Node	A decision point in an Activity Diagram
Enterprise	An entity comprising the systems, people and operations that come together for a particular purpose
Initial Node	The starting point in an Activity Diagram
Input Pin	The entry point to an Action in an Activity diagram
Join Node	A point on an Activity Diagram where one or more tokens arrive from different places
Object Token	A unit of data, matter etc that flows on an Activity Diagram
Ontology	A formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse.
Output Pin	An exit point on an Action in an Activity Diagram
Requirement	A singular documented physical and functional need that a particular design, product or process must be able to perform
Rolling Stock	locomotives, carriages, wagons, or other vehicles used on a railway
Train	Collection of Rolling Stock connected up to make a single entity
Use Case	A list of actions or event steps, typically defining the interactions between a role (known in the Unified Modelling Language as an actor) and a system, to achieve a goal.

1 Introduction

1.1 Motivation for and Background to the Research

My interest in systems engineering on the railway began in 1998 when I joined a railway engineering consultancy Enotrac, as a new mature graduate engineer. It was with Enotrac that I first came into contact with systems issues on the railway in terms of rolling stock compatibility with signalling and power supply systems, while working on various rolling stock safety cases. It quickly became apparent that single domain knowledge, e.g. of rolling stock traction systems, or power supply systems etc. was not enough to solve complex compatibility and interoperability issues involving many disciplines. A deeper understanding of the railway system as a whole and the environment that it is implemented in was required. Initial reading around the subject demonstrated this was not a new problem or even a problem that is confined to the railway industry alone. Many large industries have been contending with compatibility and complexity issues for years, including aviation (De Neufville, May 1995), electrical power supply (Gorski, Park, & Southworth, 2004), defence (National Defence Industry Association - Systems Engineering Division, 2010) and aerospace (Beasley, 2012), (Davidz, Rhodes, & Nightingale, 2005) and (Dunford, Yearworth, York, & Godfrey, 2012).

A route to solving these issues is to apply some sort of a systems engineering/systems thinking approach (Network Rail, 2013), (Rail Safety and Standards Board, 2012) and (McNulty, 2011).

Systems thinking is a way of looking at whole systems in a broad sense. It has been defined as looking at relationships (rather than unrelated objects), connectedness, process (rather than structure), the whole (rather than separate parts and entities) and the patterns rather than the content of systems (Ackoff R. L., 2010, p. 6). However, this is a rather theoretical view, for a system like a railway it is clearly desirable to consider structure, content people and processes as well. The Arkoff (2010) approach is often quite hard for engineers as it requires a change in the way we think. Ackoff & Rovin, (2003, p. 3) articulate this well with their three step process of how we (engineers) normally view things: 1 take the thing or event to be understood apart; 2 explain the behaviour or properties of the parts taken separately; and 3 aggregate the explanations of the parts into an understanding of the whole. This bottom up kind of approach has traditionally dragged us away from thinking about whole systems. This in turn has led to problems with product quality, over spend and timescale issues on projects as identified by (Elliott, 2014), Beasley (2012) and Dunford et al (2012) amongst others.

One of the key issues in understanding systems engineering and systems thinking is actually defining what we mean by a system. There are different views on the definition, it has been said that if 10 systems engineers were asked to define a system the result would be 20 definitions(Elphic, 2010), this is demonstrated in section 6.3.3, where a group of practicing engineers are asked to supply such definitions. It would be helpful if there was one clear definition that applied to the Railway industry. Chapter 2.2 of this thesis discusses this issue and proposes the following definition for purposes of this thesis, in a railway/GTS context, "*a system is a complex whole, a set of interrelated elements designed to carry out some purpose or function in a given environment*" (British Standards Institute, 1999). This is a particularly useful definition as it highlights the fact that most complex systems deliver some purpose, they don't exist for their own sake and they exist in a given environment. The environment the system will interact with is particularly important as it will most likely effect the system's ability to deliver its stated purposes against required performance.

Once an acceptable definition of a system has been identified and agreed, there needs to be some means of understanding the system of interest, in this case a GTS, in its entirety. Given the complex and multi-disciplinary nature of systems like a GTS it is necessary to look to some abstraction of the whole, with extraneous detail removed in order to make it manageable, this in essence is a system model, as described in SEBok, (2015) and Weilkens, (2007, p. 9). The same thinking also applies in the case of the sub-systems that make up a GTS and the sub systems that make up the sub systems and so on. Clearly this can only mean more than one system model, a systems model of systems models in fact. The idea of system of systems approaches is also discussed in Systems Engineering Guide for Systems of Systems (Department of Defence, 2008), (Weilkens, 2007, pp. 134-135) and (Mori, et al., 2018). Then if consideration is given to the various different ways a system like a GTS could be viewed: finance, engineering, performance and many more, the number of systems models required multiplies.

The UK railway has changed significantly since privatisation, investment has increased 60% since 1996/97; ridership has increased by 60%; and reliability has grown to over 90% (Best & Hyland, 2012). With this undoubted success has come challenges, governments and the bodies that regulate and run the railway require greater capacity, higher reliability, and of course greater cost efficiencies. It has become apparent that these opportunities are being missed or not wholly exploited through a lack of whole systems thinking in the past (in some cases the very recent past) leading to calls from those running our networks for a greater emphasis on systems engineering going forward. The McNulty report into GB rail potential (McNulty, 2011), Network Rail Technical Strategy Leadership Group (Network Rail, 2013) and the Rail Safety and Standards Board (RSSB) (Rail Safety and Standards Board, 2012) have been leading these calls in the UK. This is also underlined by the work of Elliot (2014) and Barnes et al (2015). The need for whole systems modelling, systems engineering and systems thinking is also driven by various international and European agencies, through various standards, such as EN50126 (British Standards Institute, 2010), EN50128 (British Standards Institute, 2011), EN50129 (British Standards Institute, 2010) and Technical Specifications for Interoperability (TSIs) to mention just a few.

The development of systems models is an important enabler for improving systems thinking and the application of systems engineering. However, historically there has been a lack of general systems modelling tools for the railway industry and very little sharing of data across disciplines and work groups, which has hampered the application of systems engineering principles (Network Rail, 2013 and Rail Safety and Standards Board, 2012). This situation also hampers the re-use of existing models and data and leads to repetition and waste within the industry in terms of developing many models that do similar jobs, use similar data and produce similar outputs (Bayati, Iwnicki, & Stow, (circa), 2004). There are many "systems" models, tools, databases etc. that are used within the Railway industry. The Innotrack project estimates that there are in excess of 230 with respect to track and infrastructure alone (Bayati, et al, circa 2004) and this figure is constantly rising.

In general, the majority of the models used in the railway industry are designed to represent a certain aspect of the railway, such as train performance (Chymera & Goodman, 2016) and (Goodman, 2004), wheel/rail interface (Demboski, 2006), costs vs benefits (Mantzos, 2016), projected profits etc and these usually exist in isolation. Even though they can claim to be systems models (in that, in many cases they model more than one subsystem, the interactions between these subsystems and take inputs from a number of diverse sources), they tend to only model a limited aspect and very rarely allow differing viewpoints for a diverse user group and do not link to a larger picture. Other common systems models are even more confined to their purpose, they are those which are models of particular areas of the railway, such as those which model the interaction between a rolling stock traction package and a particular part of the railway signalling system, such as a track circuit type. These tend to be bespoke purpose built models for modelling specific areas on specific railways, for example interference frequencies in return current for signalling compatibility, this is demonstrated in Ogunsola (2009) and Greatbanks (2014).

There are exceptions to the above such as the Enterprise Architecture approach used by the Network Rail Digital Railway project (Umiliacchi, et al, 2018) and the TRAK systems model (Plum N. , 2016).

There is clear evidence in literature and across industries in general that systems thinking supported by systems engineering can provide tangible benefits in terms of quality, costs and time when applied in the right way at the right time (Beasley, 2012), (Dunford, Yearworth, York, & Godfrey, 2012) and (Elliott, 2014). This is magnified in large design and build projects. Improvements in design quality in terms of understanding the problem space and improving requirements were identified in Dunford et al's (2012) work at Rolls Royce as a major benefit of the application of systems thinking and systems engineering, (Dunford, Yearworth, York, & Godfrey, 2012). Their work along with that of Pickard, et al (Pickard, Nolan, & Beasley, 2010) showed that a significant reduction in the number of requirements errors can be achieved. This also has an impact on costs and time schedules in some cases up to 40% (Honour, Axelband, & Rhodes, 2004).

In his thesis, Elliott (2014) defines a number of benefits to the application of systems engineering practices to railway projects in particular, these include timelier identification of requirements which are more accurate and comprehensive, earlier identification of problems, a better understanding of the system to be delivered and when applied in conjunction with good project management, better and more timely decision making. All of which leads to a significant improvement in cost and time management (Elliott, 2014).

Even though the evidence for the benefits of applying these techniques is plentiful, Beasley (2012), Dunford et al (2012) and Davidz et al (2005), all point out that there are barriers within companies and indeed industries to its application. They state perceived cost and effort and cultural effects in terms of resistance to change and logistical problems as being the key barriers. In their work they describe efforts to combat these barriers and realise the benefits, through:

- Training and guidance;
- Having the right processes in place for a particular industry or business unit;
- Ensuring that the processes are flexible and scalable to ensure the appropriate systems engineering overhead;
- Applying the processes at the optimum time; and
- Convincing companies to change how they work.

1.2 Scope of this Thesis

The softer issues associated with the barriers to applying systems thinking and engineering are applicable across industries and are being treated extensively through the work of Parsley et al (2013), Beasley (2012), Dunford (2012) and Davidz (2005) amongst others. However, the literature also points to barriers that are caused through the systems engineering processes themselves (Beasley, 2012), (Dunford, Yearworth, York, & Godfrey, 2012) and (Davidz, Rhodes, & Nightingale, 2005) and these tend to be industry and company specific. A GTS is a collection of interacting subsystems that also interact with the wider world. Therefore, its stakeholders need to have a common understanding of the system as a whole, as understanding the whole helps them understand the effect of new and changed subsystems on that whole. Therefore, the focus of this research project and thesis is to investigate the possibility of developing a general modelling methodology for modelling the whole system which will provide that common understanding, a guide to modellers in modelling specific parts of the system and provide much better and more complete sets of system requirements.

To this end the project will develop a GTS modelling methodology and through the application of that methodology show how that general representation/model could be applied to assist the industry in gaining a better understanding of the problem space in general and be scaled to specific problems and issues. The methodology will demonstrate how it can be applied to building general multi-view, multi-purpose, multi-aspect models of GTS' as a whole. This will enable a consistent approach that uses available data from existing industry recognised sources, models and simulators as well as bespoke models and simulators and is also able to represent the railway from a number of stakeholder viewpoints such as:

- Shareholders (financial predictions, spending, revenues etc.);
- Operators (performance, capacity, revenues, options and energy consumption);
- Maintainers (effects of maintenance strategies, component usage, forward costs predictions); and
- Passengers/Customers.

This approach should also guide the modeller on how to construct this model, e.g. have a general view on what a railway system is and what subsystems it should contain at its simplest level, based on some initial information, which can then be specialised and populated as shown below in Figure 1.

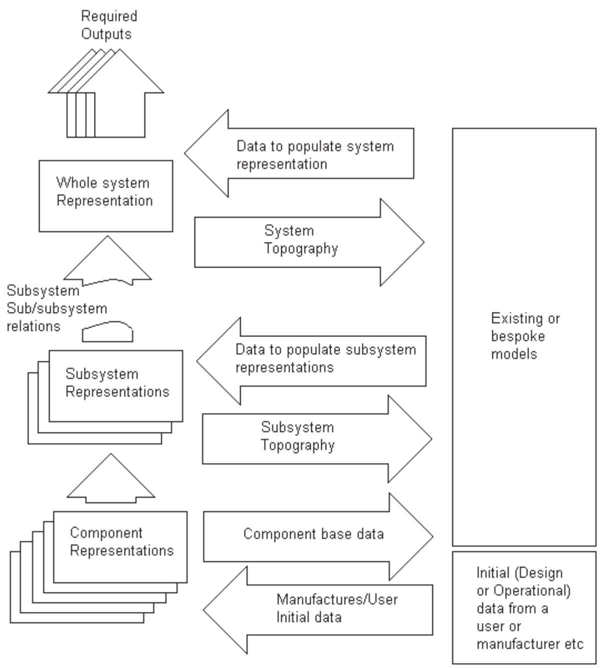

Figure 1 Modelling approach (author)

The guidance approach will be demonstrated through a Model Based Systems Engineering approach, allowing a model to be built on partial information and assumptions and expanded as it matures.

1.3 Approach

This research and thesis is structured as follows:

Chapter 1: (this chapter) provides the background and motivation for this work. It also sets out the scope and approach

Chapter 2: is the literature survey

Chapter 3: provides the research question based on the results of the literature survey.

Chapter 4: is the design section where the system modelling methodology is developed

Chapter 5: provides a partial application of the methodology to a number of integrated GTS subsystems as an initial proof of concept

Chapter 6: is the results section which details how the methodology was evaluated by a representative body of railway systems engineers and the results of that evaluation. It also explores how diverse opinions are around the railway industry with respect to systems engineering and thinking.

Chapter 7: provides a discussion on the results achieved in the previous chapter.

Chapter 8: provides the conclusions to the work as whole.

Chapter 9: addresses the need for further work in this area.

Chapters 10, 11 and 12 are appendices that contain supporting information.

2 Literature Review

2.1 Literature survey objectives

The objectives of this literature review are to understand the current state of systems engineering and thinking within the railway industry and to understand the recent and current academic and industry practice in this area. The scope will include:

- How systems are defined;
- Current thinking on what systems engineering is and how it is defined and applied;
- Defining a suitable definition of a system that is applicable to this research;
- Identify a set of necessary and optional GTS/Railway subsystems that can define when a system is a GTS and only a GTS etc;
- Recent and current systems engineering approaches within the railway industry;
- Definition of a systems engineering approaches to modelling systems going forward.

2.2 Definition of systems

Before it is possible to model a system it is important to establish, in the context of this project, what is meant by a system and what the system of interest to this project is comprised of and does. This section intends to do this. As noted by Elphick (2010) in his presentation, why is water wet? There are many definitions of a system, some of which are:

- The Oxford English Dictionary (Thompson, 1995) defines a system as:
    - A complex whole: a set of connected things or parts;
    - An organised body of material or immaterial things;
    - A set of devices functioning together;
    - A method of considering principles of procedure or classification; or

  - A group of related hardware units or programs or both when dedicated to a single application;

- The ANSI/EIA 632 Standard (American National Standards Institute, 2003) defines a system as an aggregation of end products to achieve a given purpose;
- INCOSE (circa 2010) defines a system as "a construct or collection of different entities that together produce results not obtainable by the entities alone";
- The ISO 9000 definition of a system is a set of interrelated or interacting elements (American National Standards Institute, 2015);
- The US Department of Defence Architecture Framework (DoDAF) (Department of Defence, 2001) describes a system as "*any organised assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions*";
- In her book, Thinking in Systems: A Primer, pp 11 and 17, Meadows (2009) provides two definitions "A system is an interconnected set of elements that is coherently organised in a way that achieves something" and "A system is more than the sum of its parts. It may exhibit adaptive, dynamic, goal seeking, self preserving and sometimes evolutionary behaviours behaviour".

There are also many other more specific definitions for complex systems or for when systems are defined in a particular context such as requirements engineering (Thanh, 2004).

All of the above definitions have commonalities among themselves, as expected they all deal with entities, elements or parts and all working together either in concert or to achieve some given purpose as pointed out in the definition by Long and Scott (2011, pp. 1 - 2). It is also noted by this author that they say nothing about the given environment the system has to exist and deliver in. For this reason, the definition adopted for this project is that used in the Guide to the specification of a guided transport system (British Standards Institute, 1999) "*a system is a complex whole, a set of interrelated elements designed to carry out some purpose or function in a given environment*". This definition is chosen because, although general and not GTS specific, it fits easily with the technical concept of a GTS, in terms of hardware, software and human systems etc. It also takes account of the mission provided by functions that are not necessarily technical aspects of a GTS and it is delivered in a particular environment.

2.3 Railway system components

The Office of Rail Regulation (ORR 2014) defines a railway as: "*a system of transport using parallel rails which:*

- *Support and guide vehicles carried on flanged wheels; and*
- *Form a track which has a gauge of at least 350mm or crosses a carriage way (whether or not at the same level).*"

This definition excludes mono rails, trolley buses and some metros.

This is not a particularly helpful definition in the case of this thesis as it restricts thinking to the fact that the railway has vehicles and they are supported on flanged wheels that follow a track. It has nothing to say about its purpose or how it functions or even its environment.

In a broader sense the railway can be generalised as a Guided Transport System (GTS). The ORR defines a GTS as: "*a system of transportation, used wholly or mainly for the carriage of passengers, employing vehicles which for some or all of the time, when they are in operation are guided by means of:*

- *Rails, beams, slots, guides or other apparatus, structures or devices which are fixed and not part of the vehicle; or*
- *A guidance system which is automatic*."

(Office of Rail and Road, 2015).

This is a little better in that it at least alludes to some of a GTS function, although freight and most of the environment is absent.

A better definition is "*A land based system for transporting passengers and/or goods between locations and which has a fixed means of guidance on a dedicated path*" (British Standards Institute, 1999). This definition is much better as it allows the inclusion of trolley buses, people movers and other transportation systems that don't necessarily use rails, within its scope. It also mentions its most important function, transporting freight and passengers and also that it is between defined points.

From this definition it is possible to derive the set of general components that make up the GTS as shown in Figure 2 Composition of a GTS system (British Standards Institute 1999), the guide calls the system a railway system not a GTS, however in this instance the term railway can be replaced by the term GTS. For an entity to be a GTS it must have some or all of these components.

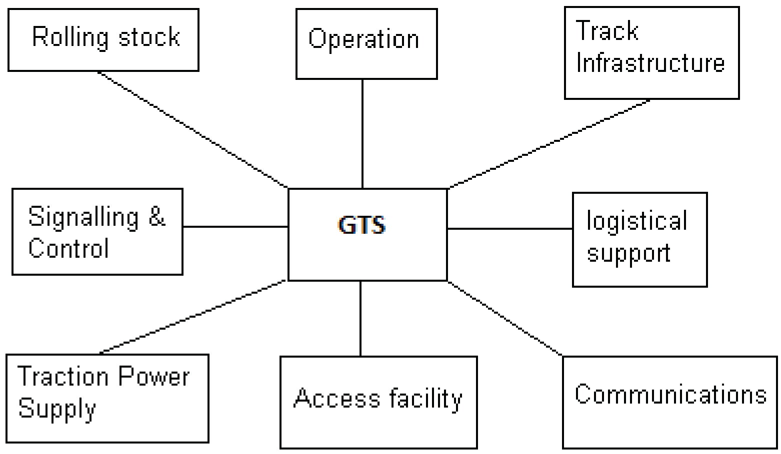

Figure 2 Composition of a GTS system (British Standards Institute, 1999)

Whilst the system breakdown shown in Figure 2 is a good starting point it describes a complicated GTS (such as a modern mainline railway) which is not necessarily representative of every GTS, some subsystems may or may not be necessary for a system to be a GTS. A much simpler view and the one adopted for this project, is to start off with a minimum set of subsystems, which when taken together, not only have the correct set of components to be a GTS but must be a GTS and only a GTS. Therefore, the author proposes that for a system to be a GTS and nothing else it requires as a minimum the subsystems shown in Figure 3.

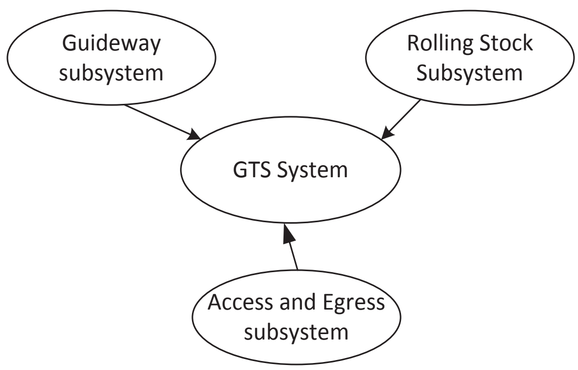

Figure 3 authors proposed minimum requirements for a system to be a GTS and only a GTS (author)

The other subsystems detailed in Figure 2 are of course necessary for more complex GTS systems, but for the essence of a GTS only the three subsystems above MUST always be present. Once a system becomes more complex signalling and communications can be added, but, for example, for a single element of rolling stock going between two points it is not necessary to have either signalling or communications systems.

2.4 System Boundaries

In order to understand the system interest, how it delivers its purpose and how it interacts with its environment, it is necessary to define its boundaries. This is not an easy task; this author has not found a definitive definition of the system boundaries for a generic railway or GTS in any of the literature identified during the research. In general systems boundaries tend to be defined by contractual arrangements. Therefore, the author proposes for the purposes of this work the following system boundaries based on what could be directly under the control of various railway or GTS entities. These assumptions can be altered for specific applications if necessary:

- Traction power supply - the power supply boundary will be the utility suppliers input to traction power supply sub-stations, it assumes that everything after this point, facing into the GTS, will be under the control of the GTS power supply entity;
- Passenger/goods access - the boundary here is the entrance to a station or goods storage area under the control of/or is the responsibility the GTS entity;
- Infrastructure and structures - it is assumed that the track sub-structure, tunnels, some bridges (application specific), viaducts, equipment enclosures (containing signalling, power supply and other GTS equipment) and anything that is enclosed by

fences and other property boundaries belonging to the GTS are considered within the system boundary;

- Logistics - the boundary with respect to logistics will be the delivery/pickup points for spares and equipment at GTS maintenance and storage areas;
- All Communications equipment and infrastructure installed within GTS boundaries to the supply point of any outside supplier (such as internet provider) are within the system boundary;
- All other GTS components, emissions etc. are bounded at the edge of GTS property; and
- Rolling stock and signalling and control systems are completely within the GTS boundary.

2.5 Systems Engineering and Modelling

2.5.1 Introduction

Just as Elphick (2010) pointed out with definitions of a system, 10 systems engineers will give 20 definitions of a system, so the same can be said for Systems Engineering itself, this is also demonstrated in chapter 6.3.3, some example definitions are given below:

- Systems engineering is a discipline that concentrates on the design and application of the whole system, as opposed to distinct parts (Booton jr & Ramo, 1984);
- System engineering is a methodical disciplined approach to design, realisation, technical management, operations and retirement of a system (NASA, 2009);

- A logical sequence of activities and decisions that transform an operational need into a description of system performance parameters and a preferred system configuration (Department of Defence, 1974); and
- Systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, lifecycle balanced set of system solutions that satisfy customer needs (Department of Defence, 2001).

There are many others of course but the above demonstrates the breadth of thinking on just the definition. Chapter 6.3.2 also demonstrates this breadth of thinking through survey questions to a representative body of engineers. However, from the above definitions, and others (not listed), the author has derived the following:

- systems engineering is a methodical approach to dealing with systems;
- systems engineering is instrumental in defining what a system is actually required to do (in terms of what it should actually deliver when it is operational);
- systems engineering facilitates the verification and validation of systems against requirements at various stages in the system lifecycle beginning at the very concept and finishing at retirement and disposal;
- It is implicit that systems engineering is a multidisciplinary pursuit and requires collaboration;
- systems engineering approaches involve bringing diverse subsystems (whether technical, human, environmental, procedural etc.) together to work toward a particular outcome or outcomes; and

- systems tend to have a life cycle (they are required, designed, built, operated and maintained and finally retired) all of which the systems engineer needs to be cognisant of.

It is noticeable that nearly all the definitions of systems engineering are very focused on design and build etc. of new systems and not so much on the analysis of existing systems.

One of the ways identified early in the evolution of systems engineering for understanding what should be done and when, was the life cycle model. There have been a number of these over the years, however these have coalesced to three main types: Royce’s waterfall model (Royce, 1970); Boehm’s spiral model (Boehm, 1988) and Forsberg and Moog’s Vee model (Forsberg & Mooz, 1998). In the Railway industry the use of Forsberg and Moog’s Vee model, as in the implementation demonstrated in EN50126 part 1 (2010) (British Standards Institute, 2010), which is shown in Figure 4 is the most common:

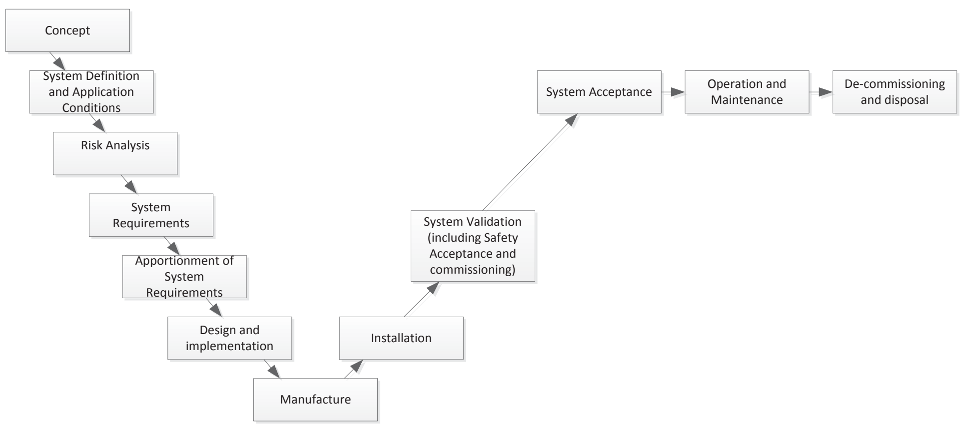

Figure 4 Forsberg and Moog's Vee Model as defined in (British Standards Institute, 2010)

Having a lifecycle gives the systems engineer a framework to work within. However, knowing what should be done where is really only the tip of the iceberg. The system stakeholders have to truly understand what it is the system is going to be and deliver. A key to this is to have a good representation of the system that shows the system in a way each stakeholder needs to see it without it being cluttered up with aspects they don't need to see. Of course each stakeholder will need a different view, but not necessarily a different model. A good systems model has views that are simply a representation of a system for some purpose, with extraneous detail removed. A good example would be the schematic map of the London Underground system. It certainly does not represent the system as it is implemented in reality, either technically or topographically, in fact it has all those (and other) details removed. It just shows stations and their relationships to each other in terms of lines connecting them. This is because its purpose to show the stakeholder group, that are called passengers, how to get from one station to another, so it needs no other detail. This is just one view of the system. If maintenance of the London underground infrastructure was the issue an entirely different view would be required, which would be another view of the same system.

The question now is how to usefully represent a whole GTS, with its many subsystems, subsystems of subsystems, complex environment, operations/missions and stakeholder interactions clearly enough to be able to understand it? This used to be achieved by the production of many paper based artefacts such as: concepts of operations documents; requirements specifications; requirements traceability and verification matrices; interface definitions documents; architectural description documents; system design specifications; test case specifications and speciality engineering analysis (this list is not exhaustive as specific projects may have different specific documentary requirements). This is all extremely difficult to manage and error prone (Delligatti, 2014, pp. 2 - 3). Whether this approach actually led to a system representation that could be fully understood is also debatable. One antidote to this is to find some modelling approach that integrates these all in one place and also manages re-use, change and documentation. One such general approach is Model based Systems Engineering (MBSE).

2.5.2 Model Base Systems Engineering

Computer modelling and simulation is increasingly used during the earliest system design phases (when the cost of change is relatively low) to define a robust system architecture and predict the emergent behaviour and properties of a design, (Pearce & Friedenthal, 2013).

MBSE is a methodology for doing systems engineering extensively using computer modelling. For the purposes of this project the author is using the definition of a methodology as defined by Martin (1996), "*as a collection of related processes, methods, and tools*". Estefan (2008) in his survey of MBSE tools for the International Council on Systems Engineering (INCOSE) characterises an MBSE methodology "*as the collection of related processes, methods, and tools used that support the discipline of systems engineering in a model based or model driven*" context.

INCOSE (2007) state "*that in many respects the future of systems engineering can be said to be model based, a key driver for this will be the fact systems are becoming too complex and intelligent for the humans who design them to comprehend and control all aspects of the systems they are creating*", this is certainly true for a GTS. MBSE is exactly what the name suggests, it's a way of approaching the problems hi-lighted in the preceding chapters regarding understanding: system structure, system behaviour, requirements, document production and management, managing interfaces, managing change and helping stakeholders to understand their proposed system. Its main point is, at its heart is an underlying system model that represents the whole system and is able to show a number of different aspects or views of itself.

With an MBSE approach systems engineers carry out all of the same life cycle processes that other approaches use, as detailed in Haskins (2006), see Figure 4, except that these are not the primary output or artefact. The primary artefact (and the word artefact is chosen carefully) is an integrated, coherent and consistent system model created by using a dedicated systems modelling tool. All the other artefacts can be said to be secondary, automatically produced by and through the system model (depending on the tool used), using the same system modelling tool, (Delligatti, 2014, p. 3). Although this author does not necessarily agree with the statement it should be all one modelling tool. It is sometimes desirable to use one tool for representing the system and others for doing analysis, this is demonstrated by Ferrogalini (2015), Sana et al (2010) and Bousse et al (2012), who have all, with varying degrees of success used more than one modelling tool to achieve their required results although not strictly within an MBSE approach.

Each design decision is captured as a model element in a single place within the underlying system model. Within an MBSE approach all diagrams and automatically generated text artefacts are merely views of the underlying system model, they are not a model in and of themselves (Delligatti, 2014, pp. 2 - 4).

Delligatti (2014, pp. 4 - 7) states that there are “*three pillars to MBSE*” as shown in Figure 5

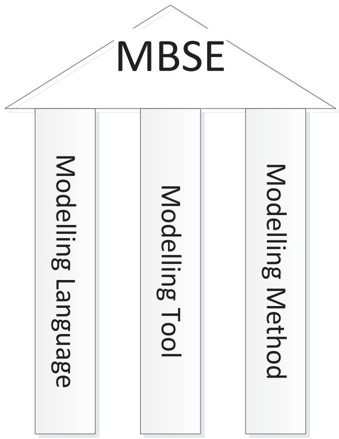

Figure 5 Pillars of MBSE (author)

The Modelling Language defines the grammar of the approach, (Delligatti, 2014, p. 5). When something is being described it is being described in some sort of language, for example a natural language (e.g. English) is often used to define hi-level requirements, formal mathematical notation is often used for verification (B method, Z, etc.), sets of equations are used to describe physical entities and their interactions, logic is used to define relationships and so on. Delligatti (2014, p. 5) recommends the graphical language SysML, Weilkens (2007, p. 16) recommends UML and Friedenthal et al (2015) speaks about both, but there are many others including but not limited to:

- UPDM, Unified Profile for Department of Defence Architecture Framework (DoDAF)/Ministry of Defence Architecture Framework (MoDAF) is the product of an Object Management Group (OMG) initiative to develop a modelling standard that supports the USA’s DoDAF and the UK’s MoDAF;
- BPMN, Business Process Model and Notation, another OMG initiative, is a standard for business process modelling that provides a graphical notation for specifying business processes;

- MARTE, is an extension of the UML profile for model driven development of real time and embedded systems, it provides support for specification, design, verification and validation.

Delligatti (2014, p. 6), describes a modelling method as a kind of recipe, a documented set of design tasks that a modelling team performs to create a system model.

Estefan (2008) identified numerous manifestations of MBSE methods in his work for INCOSE, such as IBM Telelogic Harmony-SE, INCOSE Object Oriented Systems Engineering Method (OOSEM), IBM Rational Unified Process for System Engineering (RUP SE) for Model Driven Systems Development, Vitech Model Based System Engineering Methodology, JPL State Analysis (SA), Dori Object Process Methodolgy (OPM) and add to that SysMOD (Weilkens, 2007), all variations on the theme, trying to capture system structure, behaviour, interfaces and produce views thereof, many of which were developed for use with proprietary languages and tools such as SysML, UML or Core.

The 3rd pillar identified by Delligatti (2014, p. 4) is modelling tools, they are a special class of tools that are designed and implemented to comply with the rules of one or more modelling languages. These enable development of well-formed models in the language that a particular tool supports. What is the difference between a modelling language and a tool? An example is the Enterprise Architect for SysML used later in this thesis; it supports the use of SysML and is therefore a tool and SysML is the language. It is possible to use any diagramming package to produce these diagrams, however if a change is made to an element in one of those diagrams the modeller needs to go back through all of the other diagrams that might have that element in them and change those too (very time consuming and error prone). Using a modelling tool such as Enterprise Architect means when the modeller changes an element in a diagram, he/she is actually changing that element in the underlying model and it is therefore changed in every other diagram and view of the model.

As Estefan (2008) points out "*In a nutshell, model-based engineering (MBE) is about elevating models in the engineering process to a central and governing role in specification, design, integration, validation, and operation of the system*".

2.5.3 Enterprise Architecture (EA)

Enterprise architecture (EA) is a description of an enterprise from an integrated business management and IT perspective intended to improve business and IT alignment and handle evolution and change (Kotusev, 2017). It 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 (Umiliacchi, et al, 2018).

EA began as an attempt to handle complexity in large businesses/organisations with particular reference to the IT systems that support them (Cox, 2014). EA documents the structural and behavioural building blocks that make up the overall information system together with their relationships, with respect to the enterprise in question (University of Birmingham, 2018). These are then principally used to manage change and evolution of the enterprise with reference the IT systems that support that change (Kotusev, 2017). This documentation of behaviour and structure has some commonality with SysML and UML, which can be used in the implementation of EA as will be shown later with respect to Network Rail's Digital Railway project (Umiliacchi, Bhatia, Brownlee, & Brown, 2018).

An EA is usually based on a framework within which to model an enterprise. There are many frameworks in existence a generic framework is given in Figure 6.

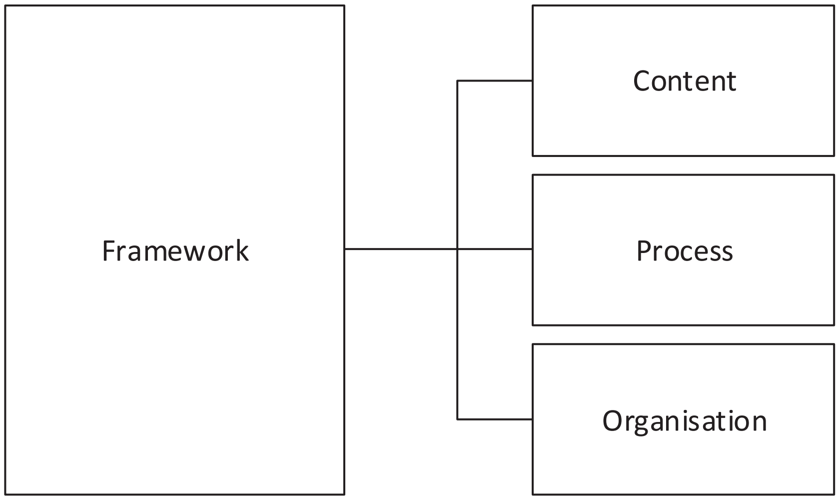

Figure 6 General EA framework (Cox, 2014)

The content part of the framework is the organisations structure (sometimes called the meta-model), the processes are the activities of the company and the organisation represents the people, roles etc involved in carrying out the organisations activities (sometimes called doing the architecture) (Cox, 2014).

The content part of the framework can be very complex as indeed large enterprises often are. In order to handle this complexity EA tends to segment the organisation into domains. There are number of ways this can be achieved, for example if the enterprise is a large government department it can segmented into departments or if it's a large organisation made up of subsidiary companies it can segmented by company each with their own domain. Cox (2014) suggests a generic segmentation into a number of logical domains (generally 4) as shown in Figure 7.

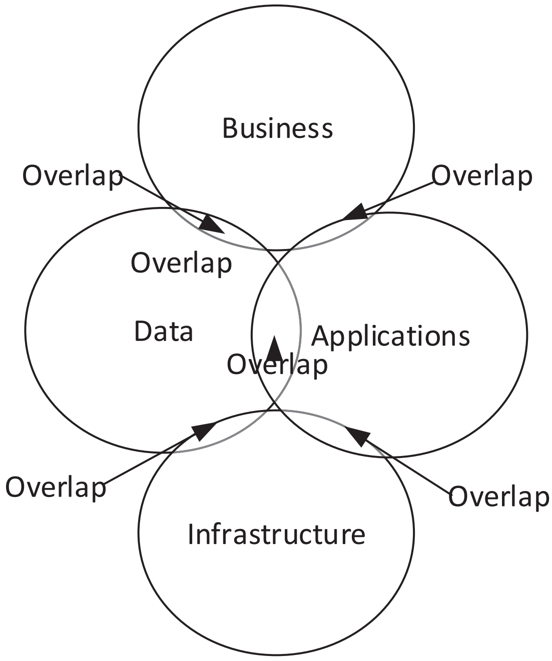

Figure 7 Generic EA domains (Cox, 2014)

The domains in Figure 7 are described as:

- A business architecture domain, which is basically the reason the organisation exists, this will typically include the organisations structure; obligations; goals; strategic thinking; capabilities; processes; and functions;
- A data domain, this has an overlap with the business architecture domain as the business will have a view on its own data, but the data domain takes a more in depth view at a lower level;
- An applications domain that focuses on the applications which offer services to the business, it manipulates the data/information in the data domain. Hence there is an overlap with both the business and data domains;
- An infrastructure domain, which from an IT point of view is the platform, software and connections upon which the application and data domains are dependant. Hence

there is are overlaps between the infrastructure domain and the application and data domains. This can be expanded to encompass non IT systems.

There are a number of EA frameworks in existence that are either general or industry specific some of the major frameworks are:

ToGAF: a generic framework designed to be applied to different businesses and their needs produced by the Open Group as standard for EA (Weisman, 2011);

DoDAF : the overarching framework developed for the US Department of Defence to support decision making at all levels in terms of planning and change (Department of Defence, 2018)

MoDAF: MOD architecture framework to support defence planning and change management activities (Ministry of Defence, 2018)

Rail Functional Architecture: Developed to assist in the identification and delivery of railway network strategic goals (Rail Safety and Standards Board, 2011); and

TRAK: originally commissioned by London Underground to assist with change management (Plum N. , 2016).

As can be seen from the above EA is expanding away from a business/IT oriented modelling approach to encompass wider organisational issues.

2.5.4 UML/SysML system modelling

For the last 25 years, since Booch and Rumbaugh began to develop UML in the mid-1990s (Pomona Colleage, 2017), the development of systems engineering modelling has increasingly been dominated by the UML/SysML languages, for software and systems engineering respectively, although other more discipline specific languages are available (Weilkens, 2007, pp. 3 - 7) and (Delligatti, 2014, pp. 1 - 10). Weilkens (2007), Paredis (2011), Ferrogalini (2015) and Delligatti (2014) all make the point that UML and SysML are languages not modelling approaches so therefore can support an integrated systems engineering approach to MSBE such as ASAP, SYSMOD or even EA approaches.

The UML and SysML languages involve describing the system of interest with a number of specific and related views, represented by various diagrams, of structural and behavioural aspects of the system, in order to describe its makeup and behaviour in its entirety. In SysML these diagrams can be divided into 4 main areas: Structural diagrams; Behaviour diagrams; Requirements and Parametric Diagrams (Freidenthal, et al, 2009 and Kamours & Boulet, 2007). Relating behaviour to structure is useful for this project as it is looking at a whole railway and its operation within its given environment. Having a parametric capability also allows for mathematical modelling and simulation later on, once the representation is developed.

UML and SysML are described as object oriented languages, they allow encapsulation through the use of ports to manage interaction between blocks and allow inheritance through decomposition and re-use (Delligatti, 2014, pp. 34 - 39 and Peak, et al, 2007).

Developing models with UML/SysML tends to be "*Use Case driven*", which captures the idea of a system performing a number of services or missions, which is helpful when thinking about a railway at its highest level, say a passenger wanting to make a journey, but not so much when thinking about performance related issues such as reliability, safety and punctuality. However, the main difference between more traditional approaches and the object oriented approach is the way the systems are decomposed, traditional methods are either data or process centric, whereas object oriented approaches (UML/SysML) contain objects that capture both data and process (ANON, Chapter 2, 2005). In fact, according the inventors of UML, Grady Booch, Ivar Jacobson and James Rumbaugh, any object oriented approach to systems development should be Use Case driven, architecture centric, interactive and incremental (ANON, Chapter 2, 2005).

Another useful advantage is that UML/SysML support the idea of MBSE, which cuts down on the amount of documentation required (as its generated through the diagrams), the repetition needed to manage relationships and change across large integrated multi-disciplined models and the associated errors caused by missing references etc. (Weilkens, 2007, pp. 3 - 7 and Delligatti, 2014, pp. 1 - 10).

A GTS, like any other complex useful system includes people, hardware, software and processes. The SysML modelling language has been proposed, by OMG and INCOSE, as a language that allows a system to be described correctly and consistently among various disciplines and stakeholders of the same project (Linhares, et al, 2006; Freidenthal, et al, 2009 and Balmelli L, 2007). It also supports the specification, analysis, design, verification and validation of systems (Kamours & Boulet, 2007 and Sana, et al, 2010).

The structural view provides the advantage of showing a traditional architectural view of a system and its related subsystems and components, which is easily communicated to other users. However, through the use of ports and behavioural/functional modelling other views can be developed including business relationships, financial information etc. (Weilkens, 2007, pp. 3 - 7 and Delligatti, 2014, pp. 1 - 10).

The use of parametric diagrams has the potential to be particularly useful in a very complex project (e.g. a whole railway system) as it allows modellers to express constraints, develop quite complex mathematical relationships and the assignment of types and values that can be managed and reused within the model (Freidenthal & Wolform, 2010 and Paredis, 2011). Peak, et al, (2007) in their papers, Simulation Based Design using SysML Part 1 and Part 2, have gone some way to demonstrate how SysML uses parameters and captures engineering knowledge. They show that the primary focus for parametrics is to support engineering analysis of critical system parameters including evaluation of performance, reliability, and physical characteristics, which again is particularly useful in modelling a GTS. (as performance and reliability are key performance indicators that feed into a number of stakeholder views). It is noted that SysML/UML does not actually do simulations or run models, it must work with other tools, such as MatLab, MathCad etc.

A further key point with models of complex systems such as GTS' is that complexity, diversity and the sheer number of engineering models and analysis required for complex system of systems models drives a lack of integration and synchronization (Peak, et al., 2007) and (Mori, et al., 2018). Parametrics provides a mechanism to address this gap and integrate engineering analysis models with system requirements and design models for behaviour and structure.

Flexibility in application is another key advantage of the SysML/UML modelling languages. Even though there are many diagrams that can be produced, SysML has 9, all of which are used in the submarine design discussed in Pearce and Friedenthal (2013), not all of them need to be produced for a particular modelling task. This is demonstrated in the survey of projects/research in the following section particularly in the work by Sana, et al (2010) and Bousse, et al (2012).

The remainder of this section is devoted to a review of some of the more promising or interesting UML/SysML research/projects particularly within the railway industry.

The EU has sponsored the development of a Europe wide rail traffic management system, ERTMS (European Rail Traffic Management System), to ensure safe operation and interoperability across the European railway system. The EU has produced specifications for ERTMS, its components and functions that manufacturers, suppliers and operators need to follow to implement and use the system (UNISIG ERTMS User Group, 1999 and UNISIG ERTMS User Group, 2008). ERTMS elements are certified for compliance and safety against these standards.

ERTMS is a distributed system; it has components that are in control centres, at the wayside and on actual trains. It consists of radio systems, train protection system, train control systems and interacts with signalling, interlocking, rolling stock and human operators. It can therefore be considered as a reasonably complex railway system, even though it is not a whole GTS system. See Abed (2010) for more information on ERTMS, its structure and application. An overview of a level 2 implementation of ERTMS is given in Figure 8.

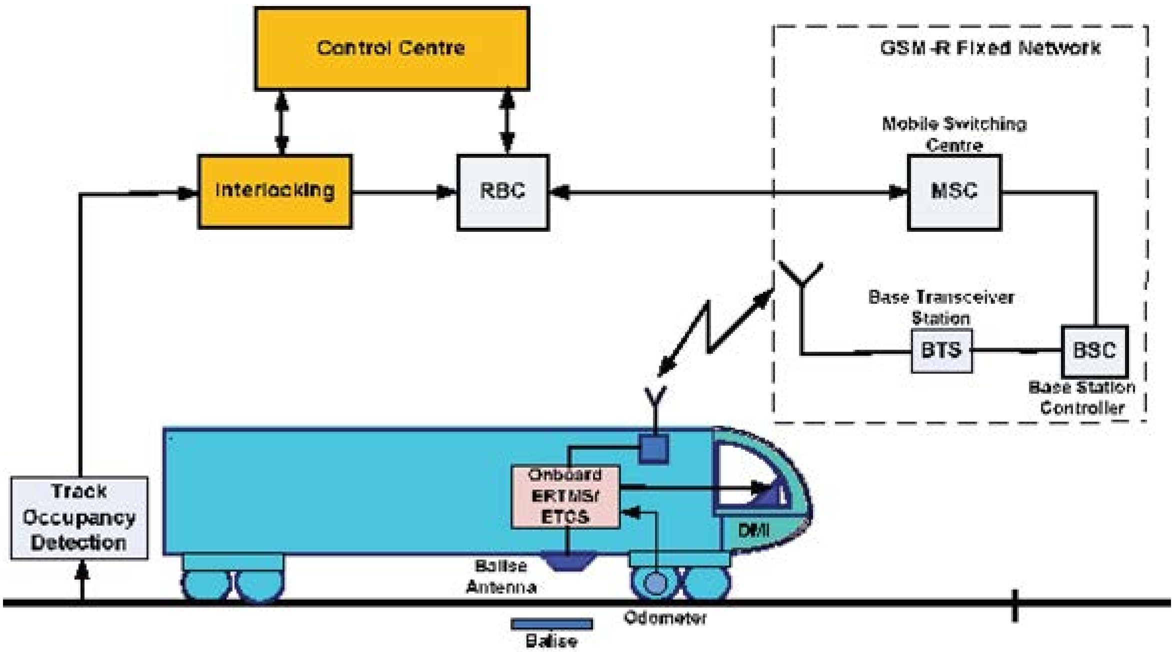

Figure 8 Overview of a level 2 ERTMS implementation (Abed, 2010)

Sana, et al (2010) have set out to determine how to use the ERTMS specifications to produce test scenarios for the validation of ERTMS components based on functional requirements.

Their chosen tools to achieve this are UML and Petri nets. Using UML or SysML type diagrams with other more formal tools is something that is advocated by other researchers in this area such as Dghaym et al (2018), Bousse et al (2012) and Farrari et al (circa 2014). This is because formal methods have been and still are a corner stone for the modelling of signalling and control systems for at least 25 years, particularly where safety is concerned (Berger, Lawrence, Roggenbach, & Seisenberger, 2017), (Reichl, Fischer, & Tummeltshammer, 2016) and (Haxthausen, Peleska, & Kinder, 2011). However, both Sana et al (2010) and Bousse et al (2012) state that the tools associated with formal methods are not very flexible and easy to update or easy for others to understand. The UML on the other hand is easier to understand and is a widely accepted modelling standard in industry with a much wider base of users. Reichl et al (2016) use UML diagrams to help explain how a railway control system fits together even though their work is on the subject formal methods. However, UML is a semi-formal language and it does not allow verification of system dynamic behaviour. So Sana et al (2010) use formal models in the form of Petri Nets that are developed from the UML diagrams.

The objective of the research work was to facilitate interoperability through the mutual recognition of the ERTMS components between the member states of the EU by proposing test scenarios enabling validation against the ERTMS specifications (SRS and FRS).

Sana, et al (2010) argue that the structure of the specifications is not formal enough and therefore not really appropriate for the production of general test scenarios. They are looking to produce UML models of each functional requirement and then to transform these into formalised models. This is carried out in three stages the first being to create semi- formal specifications using UML the second being to transform the semi-formal UML models into more formalised Petri nets and finally to automatically generate the conformity test scenarios.

The fact that the ERTMS specifications are constantly evolving means that any modelling language and approach needs to be easily changed, so UML was chosen principally because of its object oriented nature. It also drives an object view with structure and behaviour captured. This is an advantage that has also been underlined by Peak, et al, (2007). An object oriented approach can yield the following benefits:

- Maintainability through mapping to the real world which in turn leads to reduced design effort, less complexity and easier usability;
- Reusability of analysis artefacts, which saves time and costs; and
- There could also be some saving in later use of Object Oriented programming languages for final implementation.

ERTMS requirements specifications tend to specify behaviour, which lends its self to translation in UML/SysML type behavioural diagrams. For their work Sana, et al (2010) chose to manually translate the ERTMS European Vital Computer (EVC) specification into UML state machines and sequence diagrams. They hi-lighted the fact that both diagrams are views of the same model and therefore they could use same entities in both diagrams.

The approach was based on a decomposition of UML State Machines into its basic elements, such as states, pseudostates, and transitions. For each element, transformation rules from State Machines into Petri Net fragments were developed. This allows time constraints on transitions to be taken into account. Sequence diagrams, were used to guide the connection of these Petri Net fragments so as to produce a single Petri net to represent the system of interest.

It is not possible to apply mathematical techniques directly on UML diagrams because it lacks a formal semantics (SysML with Parametrics may indeed have been a better choice). The authors cite a number of other studies where UML diagrams need to be translated into other formats to achieve some sort of formality.

This does however display some of the versatility gain from using UML/SysML, in that it is not necessary to produce all of the diagrams to produce the systems model that is required (Weilkiens (2008) pp 65). In this case no object or Block diagrams (sometimes considered the starting point and heart for the system model in UML/SysML) were needed, just the behavioural diagrams.

Ferrari, et al, (circa 2014) reports the experience of a railway signalling manufacturer in introducing the SysML notation within its development process by means of the TOPCASED tool.

General Electric Transportation Systems (GETS), a company that produces safety critical railway signalling systems, in collaboration with the University of Florence set about trying to improve their development process with the application of formal methods. This development process has led to the introduction of formal modelling and code generation and produced a model based process compliant with CENELC Standards.

The Simulink/Stateflow tools used proved very powerful for formalising low level requirements, but less good at high level requirements specification and analysis. These high level tasks were usually carried out through a paper based approach in natural language. The natural language necessarily used in this approach proved to be too ambiguous even for high level requirements. As with Sana et al (2010) the idea of introducing the use of SysML to provide the bridge between natural language and formal methods was introduced.

TOPCASED version 3.0.13 (an open sourced tool), was introduced for the requirements specification of the Failsafe Data Transmission (FDT) system, a platform that manages the switching of the traffic direction between adjacent stations. This tool provides a model validation feature that allows the internal consistency of the produced model and its compliance to the SysML standard to be checked.

In common with Sana et al (2010) and Bousse et al (2012), only a subset of the diagrams available was employed, Use Case diagrams, Requirements diagrams, Sequence and Structure diagrams (package, BDD and IBD) were considered sufficient to specify the system to the right degree of detail. GETS used a single model structured into packages: each package corresponding to a CENELEC phase on the V process (see Figure 4) and including the diagrams to fulfil the standard prescriptions for that phase. For example, the requirements phase includes mainly Use Case and Requirement diagrams, while the architecture phase is essentially documented with Block Definition and Internal Block diagrams.

The model was built incrementally, and each artefact of each phase was traced to the elements coming from the previous one.

Today GETS uses SysML on large projects. However, all the official specifications required by the CENELEC norms are manually edited natural language documents. This is only because assessors who (they say) normally join a project at the end of the development process to validate compliance with the standards, require paper-like documents in order to have a complete picture of the activities performed by the company. While this implies there is still a major effort in terms of production and maintenance of the documentation, the authors report that the investment on SysML paid off in terms of increased confidence on the quality of the specifications. This project was deemed a partial success.

In research that is similar to that of Sana et al (2010), Bousse et al (2012) set out to identify a verifiable subset of SysML that is usable by systems engineers, while still being transformable into formal verification tools. This again deals with using SysML and formal methods for Verification and Validation as with Ferrari et al (2014) and Sana et al (2010). In this case the formal method is the B method.

The entry for the B method in Wikipedia states, "*the B method is a method of developing software based on B, a tool supported formal method based on abstract machine notation. It was originally developed by Jean-Raymond Abrial in France and the UK. B is related to the Z notation (also originated by Abrial) and supports development of programming language code from specifications. B has been used in major safety-critical system applications in Europe (such as the Paris Métro Line 14). It has robust, commercially available tool support for specification, design, proof and code generation (ANON, B-Method, 2018).*"

The process developed was applied to a simplified railway crossing controller supplied by Mitsubishi Electric. A SysML model was designed with safety properties, then automatically transformed into B, and finally imported into Atelier-B for automated proof of the properties. As with Sana et al (2010), Bousse et al (2012) noted that formal methods although ideal for Verification and Validation of safety critical systems and showing compliance with standards such as EN50128 (British Standards Institute, 2011), IEC 61508 (British Standards Institute, 2016) etc., they can be difficult for others to read and understand. To address this they wanted to align formal methods with SysML using what we now call MBSE techniques (Delligatti, 2014, pp. 2 - 9). The main SysML diagrams applied are Block Definition Diagrams and in common with Sana et al (2010) State Machine Diagrams. The resulting process that was developed was then applied to the simplified railway crossing controller specification.

SysML was chosen because Bousse et al (2012) wanted to use a language that can link requirements and elements of the model, as they needed a language that was complete in that it provides ways to design systems at a number of levels of abstraction.

In order to align the B method with SysML Bousse et al (2012) looked for what they call semantic similarities between SysML and the B method, because they wanted to construct models in SysML that could be translated into the B method while preserving the semantics. Altogether 12 semantic similarities were identified as shown in the table below:

B Method	SysML
Project	SE of blocks
Module	Block
Imports link	Part property
Sees link	Reference property
Basic type	Value type
Basic data	Value properties
Constraints	isReadOnly meta-attribute
Enumerated set	Enumeration
Component parameters	Default values
Initialization and valuing	Default values
Operation	Operation (with behaviour)
Invariant	Constraint property

Table 1 List of semantic similarities

Using the similarities, the authors defined a subset of SysML which allows only a few constructs. Each model was a collection of blocks with primitive data in value properties, linked by part property links for the system decomposition, with test operations to define their behaviour and required constraints over variables. The UML profile was used to extend the SysML language to create stereo types that were required to complete the set of similarities.

Alstom have introduced an MBSE approach to their railway development system called ASAP. They develop systems views with a top down approach which begins with Requirements. An approach similar to that adopted for this thesis.

The system objects are stored in two synchronised databases, the Requirements are stored in a Telelogic systems DOORS database and the other database is a system architecture database implemented in SysML a tool provided by Atego Artisan.

The requirements drive Use Case diagrams which drive train functional architecture diagrams which in turn drive the sub system architecture.

Ferrogalini (2015) states that this approach drove the following improvements in design quality:

- Rigorous traceability between requirements, their implementation into the system architecture, verification and validation;
- Enhanced coherence and consistency (interfaces) between all the different subsystems; and
- Rigorous management of design change and system architecture configuration.

There are also reported improvements in productivity through:

- Reuse of existing models to support design evolution;
- Reduced errors and time during integration and verification and validation;
- Enabled concurrent system architecture definition; and
- Documents being generated automatically out of the model.

Other gains were enhanced communication in terms of a shared understanding of the system and its analysis across the development team and other stakeholders, which was greatly enhanced by the ability within SysML to integrate a number of views of the system based on the underlying model. Knowledge transfer was also improved because design could be captured in a standard format that could be easily accessed. These improvements confirm the conclusions made by Beasley (2012), Dunford et al (2012) and others with reference to the gains that can be made where these attributes of systems engineering are applied.

The presentation gives the impression that the application of SysML and MBSE is still embryonic within Alstom but it is defiantly expanding and the results are demonstrable.

There are a number of other projects/research regarding the use of SysML and UML that are worth mentioning:

Richards et al (2009) produced a paper for INCOSE detailing the use of SysML highlighting how UML and in particular, improvements introduced by SysML can aid the testing process in terms of verification, validation and simulation of software, firmware and mechanical systems. This again has much in common with the work of Sana et al (2010) and Bousse et al (2012), in terms of using these techniques to drive the development of testing scenarios. It demonstrated the use of SysML through the medium of a cruise control system.

It comes to the conclusion that Test Plans can and should be modelled. The test plans should define the nature and purpose of the whole range of testing. Parametrics can also be specified and modelled in terms of input-parameter-to-output-results tuples and automatically verified within the model itself (or an external script). The logical progression from Test Plan, with associated models to suites of tests and their associated models to Test case and models to actual test scripts is a refinement and decomposition process that lends its self to modelling in SysML rather than traditional systems and software engineering analysis and design. It should be noted that this demonstration was produced by Artisan employees who market UML and SysML tools, having said that, it gives clear evidence of use and versatility of SysML.

Erd (2007), presents a rail vehicle maintenance system modelled with UML, whose intended use is to generate software code automatically and then to extend it to use in other industries such as ship maintenance or power generation.

The approach was different to any of the others described; the system modelling is broken down into three elements:

- Information elements allowing a full description of concepts (including database construction);
- Expected output information (user interface); and
- Data processing (application logic).

This approach naturally divides into three layers, i.e. data access, logic and presentation. Erd's idea was to make the functions of individual layers independent from one another and to prevent changes in one layer from affecting the other layers.

As this was UML, the base elements of the model are the classes which represent its components or concepts. Like SysML blocks, the classes possess attributes (parameter values) and methods, which operate on these attributes. Classes also described concepts such as measurements e.g. oil analysis, wheelset size measurements etc.

The Class diagram was able to model aggregation through relationships as with SysML blocks.

The maintenance system being modelled involved the use a large amount of data. This data was divided into three main groups:

- Non-changeable (Constant) data, including number of units in stock, configuration data;

- Measured data, including oil consumption, wheel set measurements, analysis results etc.; and
- Changeable (variables) data including register of vehicles in rolling stock library, usage of spares etc.

Use Cases were created to show how the various actors in the system preformed tasks and entered and received data. The Use Case diagrams were particularly useful when it came to specifications of system element behaviour as seen from outside.

A representation of the various states that the system would be in or transfer to as a result of individual actions was developed.

The model developed by Erd was clearly a starting point for creating a model design of the maintenance system. It is a model with a single purpose that is to significantly shorten the development time for diagnosis and maintenance systems. The idea was to create a design pattern (model) which could be used in a possibly broad range of applications, although the paper does not really demonstrate this. This work does however demonstrate that the UML modelling language is a useful tool for showing the interdependencies between the system elements (static model) and its operations (dynamic model).

This section is not intended to be an exhaustive survey of all the rail biased SysML/UML projects and research there is, it provided to give a flavour of some of the most interesting, from a perspective of modelling a GTS.

2.5.5 Ontological Approach

A way of approaching the difference in understanding between the disciplines involved in GTS type systems is to find some way of communicating data, knowledge and information in terms that are understood by all those involved. This can be achieved by using ontology, (Gruber, 1993; Borst & and Akkermans, 1997 and Noy & McGuiness, 2001).

Ontology defines a common vocabulary for those who need to share information in a domain. It can include machine-interpretable definitions of basic concepts in the domain of interest and the relations among them, (Noy & McGuiness, 2001). Ontology also provides criteria for distinguishing various types of objects (concrete and abstract, existent and non-existent, real and ideal, independent and dependent) and their ties (relations, dependencies and predication), (Corazzan, 2010).

A GTS is an extremely complex system which requires and generates vast amounts of heterogeneous data that has to be understood and used by the various models that are already in existence. This can be dealt with by an ontology implemented in software such as Protégé OWL. Ontologies can also be implemented in UML/SysML (Umiliacchi, Bhatia, Brownlee, & Brown, 2018). It should be noted however at the time of writing (2019) there are some limitations on the size of data sets that reasoners can deal with (Tutcher, et al, 2017). The ability to use existing ontologies gives the opportunity for ontologies in the various areas of the railway to be developed (with rules, Ontological commitment etc.), (Umiliacchi, et al, 2011). These can be developed and maintained by field experts and then imported into a general ontology. For example, the signalling part of the railway can have its own ontology developed with signalling experts. This then holds the relevant signalling information (but adhering to the Ontology rules about vocabulary etc.) and it can then be imported into a railway ontology for use in modelling the whole system.

The semantic web uses Ontologies for modelling shared meanings, concepts and theories, (Neaga, et al, 2006). The search for faster and optimised methods of design and manufacture has recently resulted in the need for greater information exchange or even knowledge sharing between users and designers of the various system entities.

Getting the right information to the right people at the right time is extremely important, (Umiliacchi, et al, 2011), for example the failure to recognise and correct an error in the transfer of information between the Mars Climate Obiter space craft team in Colorado and the mission navigation team in California led to the loss of the space craft (Mars Climate Orbiter Team Finds Likely Cause of Loss, 1999). Now if there was a clear pattern to the information being supplied this could have been caught (Neaga, et al, 2006).

Noy and McGuiness (2001) stated that there was no right way or wrong way to build a ontology, but other authors recommend a number of criteria that should be taken into account. These are best summarised by Gruber (1993) who asserts, "*the design criteria that should be applied when developing an ontology are:*

- *Clarity: an ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective and should avoid being too context dependant. In the case of this thesis, a general railway GTS definition can be used to define any GTS regardless of the type of GTS or its environment. It can then be specialised by a user adding more component classes and individuals as required under the given hierarchy, within the proposed rules and agreeing to use the suggested vocabulary;*
- *Coherence: an ontology should be coherent, that is it should sanction inferences that are consistent with the definitions. At the least defining axioms should be logically consistent. If a sentence that can be inferred from axioms contradicts the axioms, then the ontology is incoherent;*

- *Extendibility: an ontology should be designed to anticipate the uses of the shared vocabulary. It should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialise the ontology monotonically;*
- *Minimal encoding bias: The conceptualisation should be specified at the knowledge level without depending on a particular symbol level encoding. An encoding bias results when representation choices are made purely for the convenience of notation or implementation and this inhibits knowledge sharing;*
- *Minimal ontological commitment: An ontology should require minimal ontological commitment sufficient to support the intended knowledge sharing activities. Ontological commitment is when different users/agents agree to use the same vocabulary to represent a universe of discourse. An ontology should make as few claims as possible about the world being modelled, allowing the parties committed to the ontology freedom to specialise and instantiate the ontology as needed. This can be achieved by specifying weakest theory (allowing for the most models) and defining only those terms that are essential*."

In recent times ontological approaches have increasingly been used for various modelling and data integration projects across the railway industry. These include:

- Requirements specification and validation (Hoinaru, et al, circa 2013);
- Maintenance and condition monitoring, (Umiliacchi, et al, 2011; Lewis, et al, 2008 and Easton, et al, 2011);
- Data integration and interoperability (Umiliacchi, et al, 2008; Langer, et al, 2008; Moris, et al, 2015; Tutcher, et al, 2017 and Gould, et al, 2017);

- development of customer querying facilities (Mohan & Arumugam, circa 2012); and
- planning processes (Lodemann & Luttenberger, 2010).

The following section provides an overview of these approaches and looks at where these can be taken in the future, in the context of this project.

Hoinaru, et, al, 2013 were at the very early (feasibility/initial modelling) stages of a project to develop an ERTMS/ETCS Ontology aimed at modelling and formalising the ERTMS specifications and other recorded (written) knowledge, in order to obtain a data structure that is reusable in other ERTMS work, particularly for software development, this is similar to the work carried out by Sana, et, al, (2010) and Bousse et, al (2012) using UML and SysML based approaches.

Hoinaru et al (circa 2013) use the Web Ontology Language (OWL) as implemented in the Protégé modelling tool developed at Stanford and Manchester universities.

At the initial stage, their Ontology is based on three base ERTMS documents as follows:

- ERTMS System Requirements Specification (SRS) produced by the European Railway Agency (ERA);
- ERTMS Glossary; and
- ETCS implementation handbook published by the International Union of Railways (UIC).

The ontology itself is a semantic model extracted from the above documents. The knowledge extraction is to be carried out manually after studying the documents. There is an ambition that this work would be a precursor to developing methods of carrying out this extraction automatically.

Given that the key function of ERTMS is information exchange Hoinaru et al (circa 2013) concentrated their initial efforts on this part of the system as a demonstration of its fitness for purpose.

The SRS specifies ERTMS at a high level. The idea is to take the concepts expressed in the documents and model them as a superclass, subclasses taxonomy with their various characteristics modelled as properties. Relationships between the various concepts and data will be modelled through concept and data type relationships along particular properties.

Hoinaru et al (circa 2013) divided their ontology into a number of modules:

- Entity module, a super class containing several concepts such as Driver, ERTMS and Procedure. It describes individuals that are used to define the required system behaviour;
- OSI (Open System Interconnection) Module, which is a sibling of the Entity module, which describes the concepts that characterise and standardise the internal functions of the communications system by partitioning it into layers;
- Source, another sibling of Entity it formalises the documents mentioned above to form the core of the ontology; and
- TrainCategories module is also a sibling of Entity and describes concepts supporting different types of rolling stock.

In the SRS, the procedures associated with mode transitions and dynamic behaviour of the ERTMS system are defined by flow charts an example of which is shown in Figure 9.

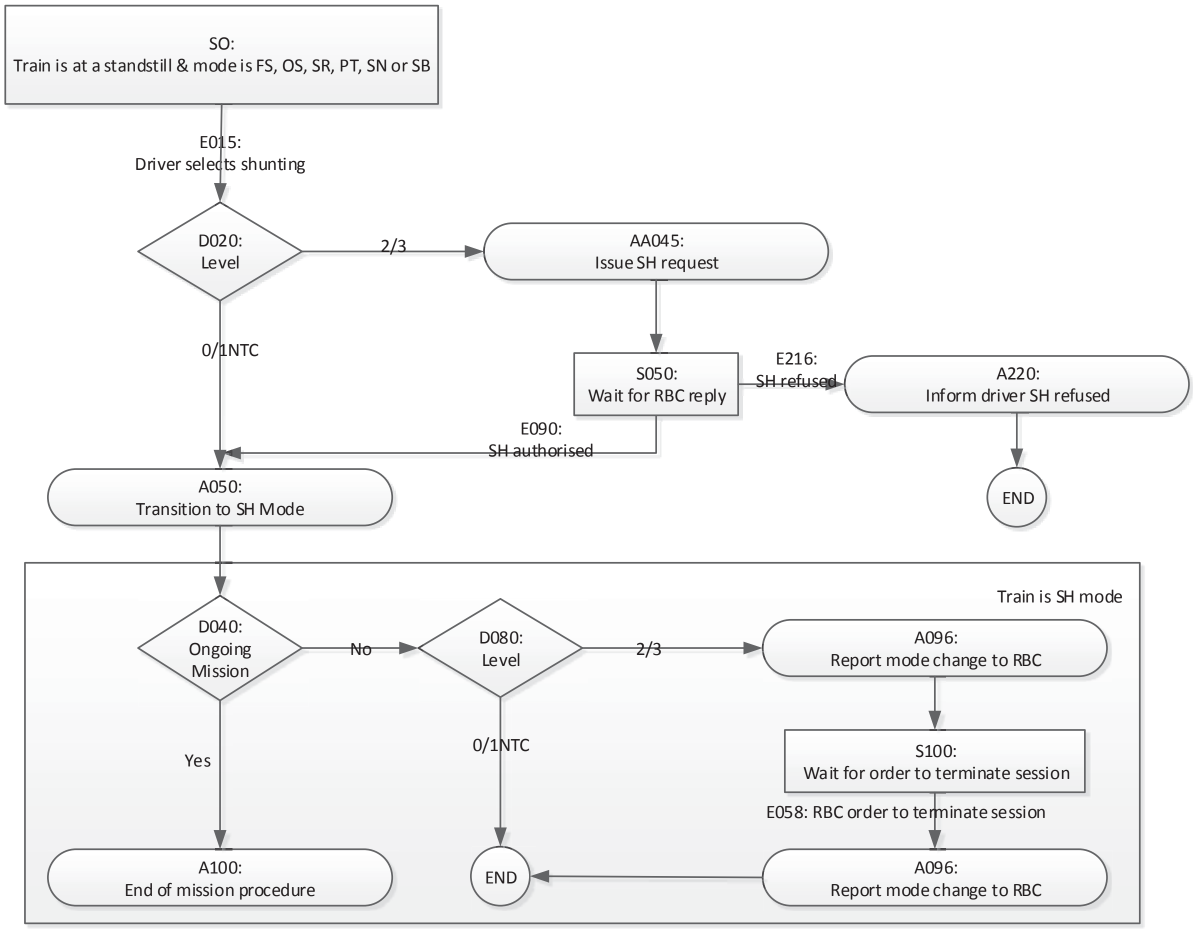

Figure 9 Flow chart of the shunting procedure (Hoinaru, et al, 2013)

In order to capture the semantics of these functions the flow charts were translated into Semantic Web Rule Language (SWRL) rules in much the same way as Lodemann and Luttenberger (2010). Sana et al (2010) and Bousse (2012) accomplished something very similar with UML and SysML behavioural diagrams. Hoinaru (circa 2013) also mention that these flow charts could have been implemented with UML state machines. This is a good idea as it would allow the use of an initial systems modelling approach through UML/SysML etc., allowing the rules in question to be visualised. SWRL can be difficult to understand for those who do not have experience with logical expressions. This would also provide a series of diagrams from which to build the ontology and capture the dynamics of the system at the same time.

As mentioned above ERTMS is a system that relies on information exchange so the Hoinaru et al (circa 2013) concentrated their efforts on the OSI module. The OSI model is a conceptual model that characterizes and standardizes the internal functions of a communication system by partitioning it into abstraction layers. The model is a product of the OSI project at the International Organization for Standardization (ISO), maintained by the identification ISO/IEC 7498-1. The model groups communication functions into seven logical layers. A layer serves the layer above it and is served by the layer below it. For example, a layer that provides error-free communications across a network provides the path needed by applications above it, while it calls the next lower layer to send and receive packets that make up the contents of that path. A graphical representation is given in Figure 10

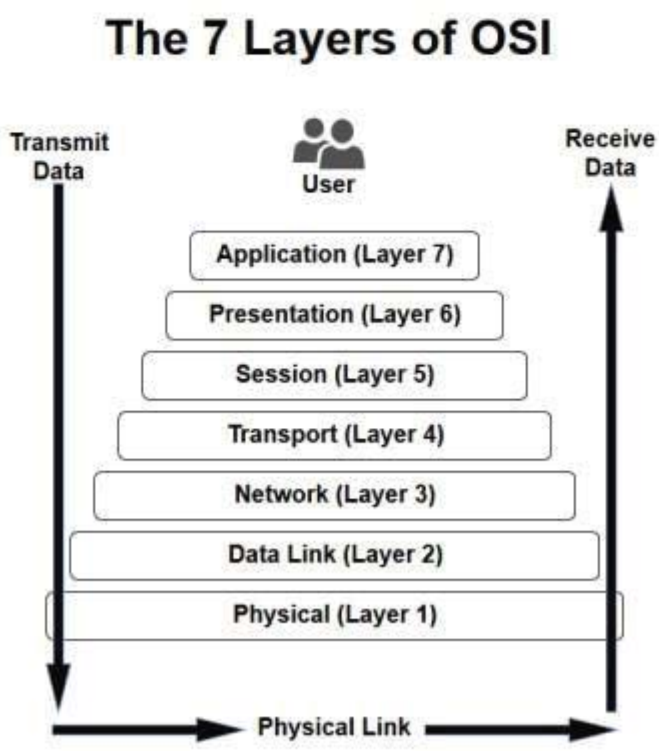

Figure 10 OSI seven segment model (ANON, OSI Seven Segment Model, 2019)

OSI model is a generic 7 layered model whereas Hoinaru, et, al, (circa 2013) have condensed the ERTMS communication system as a three layered model as described below:

- GSM-R Layer (Transports data packets via the cellular network between the train and radio block centre);
- Euro Radio Layer (Handles end to end communications between train onboard applications and ground system applications); and
- ETCS Layer (Manages the messages at the application of ETCS level, e.g. communications between and EVC and RBD for a Movement Authority).

Hoinaru et al (circa 2013) used the inherent reasoning within OWL enabled by the relationships hasUpperLayer and hasDownLayer. A class at the top has no hasUpperLayer relationship and if it is at the bottom of the hierarchy it has no hasDownLayer.

Hoinaru et al (circa 2013) were clearly at the very beginnings of their work and so far have dealt with possible approaches as to how they might model an ERTMS system with an ontology. Their paper demonstrates that ontology modelling is beginning to be applied to areas that are not just concerned with the semantic web or data interoperability, but as a system modelling tool in its own right. They have demonstrated the feasibility of their proposed modelling approach to a complex systems problem. It is interesting to note that they are also looking at linking their modelling to other approaches, such as state machines etc., in order to help them derive a better representation within their ontology of the system of interest. This could be yet another example of the SysML/UML approach working with other tools, approaches. This idea is taken even further by Umiliacchi et al (2018) in their paper on Network Rail's EA approach where they represent their ontology in a UML diagram.

Lodemann and Luttenberger, (2010) are further ahead of Hoinaru et al (circa 2013) in terms of their model development. They looked at how new railway lines (infrastructure only) were planned, verified and certified in Germany, including:

- the deployment of the physical elements (tracks, switches etc.) and their alignment with reference to each other;
- how these physical objects are viewed in terms of more abstract elements such as routes;
- how legal (in this case German) rules and legislation relate to the new planned route in terms of safety (how the interlocking scheme demonstrates adequate safety); and
- how this information is used to provide approvals and certification.

The problem they are trying to solve is how to streamline or semi-automate the very inefficient task of planning a railway lay out and then comparing it against the legal and safety requirements (for interlocking, train detection etc.) and achieving approvals and certification, again this is very similar to some of the UML/SysML approaches described above Sana et al (2010) and Bousse (2012) particularly. It also has similarities the work of Hoinaru et al (circa 2013) in terms of trying to get a more efficient validation against specifications. At the time of their writing this process was inefficient because there were/are very few computerised tools to aid the process, particularly those which represent the railway infrastructure (although this has been changing very recently with work on railway domain ontologies (Moris, Easton, & Roberts, 2015) and RailML (Nash, et al, 2010)), so the majority of the work is achieved by the exchange of paper based documents. The company planning the railway had to transform their (usually computerised) plans into paper based tabular form, these are then passed to the authorities, who apply the rules manually. An error report is then produced and all the paper work is returned to the company. The work is then re-digitised and corrected and process starts again. This results in a large overhead with very little re-usability between jobs.

Lodemann and Luttenberger (2010) aim to achieve this semi-automation by the development of an ontological framework where the implicit expert knowledge of the railway infrastructure and the German guidelines for Railway Infrastructure and their relationships are made explicit and stored in a concept model. The modelling language chosen is OWL and rules themselves are applied by using SWRL which were enriched with railway specific built-ins, these rules then reside in the ontology. The attempt was not to develop a complete model only a proof of concept model demonstrating the possibilities going forward.

The ontological approach was used by Lodemann and Luttenberger (2010) because of its ability to expressively represent the semantics of the railway domain when compared with other approaches such as syntax only XML representation, a point also made by Tutcher et al (2017). However, Lodemann and Luttenberger (2010) did not use ontology for whole modelling approach, they followed a bottom up process for developing their ontology based on the railway specific XML schema, RailML which was then extended to cover all the required infrastructure classes. The Lodemann and Luttenberger (2010) also interviewed experts in the field to capture knowledge. This approach to knowledge capture was part of the approach adopted by Tutcher et al (2017) and Hoinaru et al (circa 2013).

Lodemann and Luttenberger (2010) began their infrastructure model with the concept of track and that track has a spatial extent and a beginning and an end. These can then be interconnected with each other and other concepts such as switches. All the other concepts in their model e.g. signals, balises, train detection etc. are related to track. This approach can also handle virtual elements such as routes, which can subsume the other concepts. Similar ideas are now part of the RaCoON Ontology (Tutcher, et al, 2017).

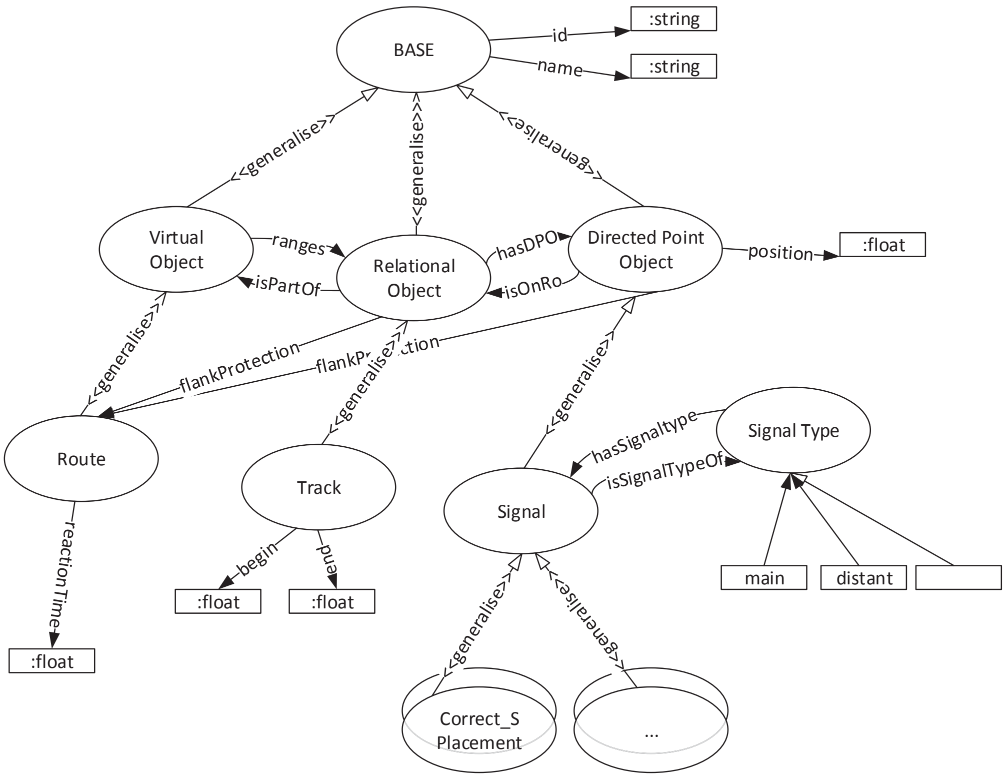

Figure 11 An excerpt of a Class Ontology (Lodemann & Luttenberger, 2010)

A portion of Lodemann and Luttenburger's model is illustrated in Figure 11. The classes are represented by ellipses, individuals by rectangles (e.g. main, distant), datatype properties by named connections from a class to rectangle (e.g. :float) which show the actual data type and object properties by named connections between classes:

- The top level class, Base, contains two properties *id* and *name* to provide a unique identifier and a human readable name, these are inherited by all the other subclasses.

- The RelationalObject class contains all subclasses for the physical elements of the system such as Track.
- The DirectPointObject which contains subclasses such as signalling are aligned with RelationalObjects by the properties isOnRO (meaning is on Relational Object) and hasDPO (meaning has a DirectPointObject).
- The VirtualObject subclass contains all the virtual concepts such as route.
- The subclass of DirectPointObject, Signal is related to the subclass SignalType which contains individuals representing the various signal types, through the properties hasSignalType and its inverse isSignalTypeOf.
- The subclass signal also has subclasses for verification. These subclasses are populated as a result of classification (and the application of SWRL rules). After classification verification is achieved by classifying individuals a members of the class Correct_S_Placement if they agree with German rules and guidelines as translated into SWRL. An example of sort of sematic SWRL rule used is shown below. The rule formalises the following directive from the German rules: "*A signal needs to be placed within the range of the corresponding track it is assigned to*".
- Signal (?) ^ Track (?t) ^ to(?t, ?t0) ^ signalIsOnTrack(?s, ?t) ^ from(?t, ?from) ^ SignalPosition (?s, ?pos) ^ swrl:greaterThanOrEqual (?pos, ?from) ^ swrl:lessThanOrEqual (?pos, ?to) -> Correct_S_Placement (?s)

It can also be seen from Figure 11 that members of subclass Route have direct relationships with RelationalObject and DirectPointObject showing how a route is made of individuals that are members of both.

The complete ontological verification system developed by Lodemann and Luttenberger is shown below in Figure 12. The system has been segmented into various logical and physical parts. The lowest level contains the class ontology which contains the railway concepts taken from the railML schema, extended with real knowledge from interviews with experts. The class ontology also contains the categorisation classes as shown in Figure 11. The rule ontology includes the class ontology and the SWRL rules. The highest level ontology is the individual ontology; it is generated by applying XSLT scripts to a document in railML format. This railML document contains the actual planning data in terms of precisely defined infrastructure objects.

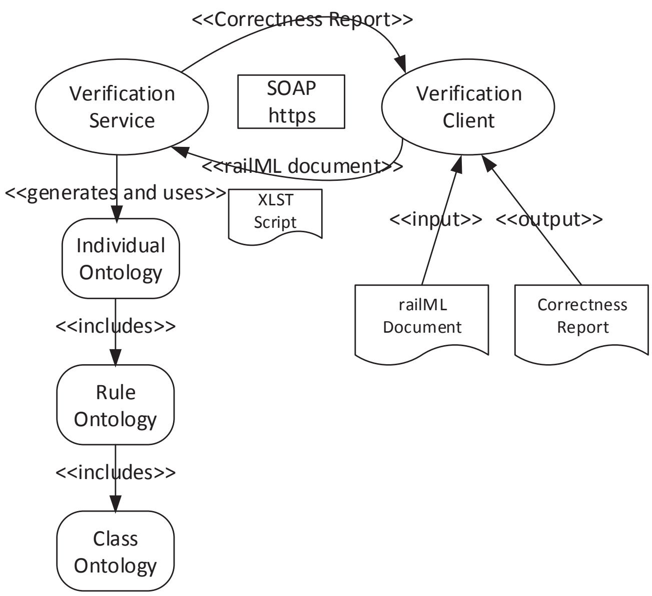

Figure 12 Ontological Verification System (Lodemann & Luttenberger, 2010)

The individual ontology is data specific whereas the class and rule ontologies form an unmodifiable knowledge base. Invoking the verification service, the objects within the individual ontology are to be verified automatically against the German railway rules modelled in the class and rule ontologies.

As an application the verification service is a web service which wraps around the ontology and reasoning framework. It is invoked by a client who communicates via the standard Simple Object Access Protocol (SOAP) mechanisms. The user uploads the planning data in the form of an extended railML document; this is then transformed by the XSLT script into the individual ontology, which via the import facility includes the rule and class ontologies. The verification is then carried out as described above and a correctness report is generated, this is then transferred back to the client via the SOAP mechanisms.

This work shows very well how a complex railway system ontology may be implemented. It is interesting however that the actual representation of the railway system starts with a railML description of the concepts which is then used to build the ontology. This was also used for the RaCoON Ontology (Tutcher, et al, 2017). This is something that may well need to be taken onboard for building a more complex whole railway system model, using a tool such as sysML/UML.

Umilacchi, et, al, (2008), state that there were a number of problems with information systems in the railway environment, (also noted by Lewis, et al, (2008), Langer et al (2009), Easton et al (2011) and Tutcher et al, (2017) these are:

- There is a lot of data available, but it can be difficult to retrieve easily;
- Once data is extracted it is not in a state where it can be easily processed further by other systems or used in conjunction with data from other sources;
- Maintenance of the information systems themselves is expensive in terms of updating, checking consistency and expansion/modification; and
- Data sharing is not always easy as it is usually aimed at a specific user group within a specific area

Umiliacchi et al (2008) and Tutcher et al, (2017), also argued that data sharing is becoming increasingly more important in the railway industry, driven by the commercial and operational trends towards interoperability particularly in the European Union but also the knock on effect this is having on the rest of the industry worldwide, an example of this being the number of non-European implementations of ERTMS e.g. China, Brazil and even North America. Now in 2018/19 this argument has been proved correct, with even greater strides towards interoperability particularly within the EU, TSI compliance is now mandatory within all major rail projects within the UK, recent examples being, the Crossrail project, the Thameslink upgrade and the introduction of new high speed trains for East Cost Main Line and Great Western Main Line.

Work has been carried out by Umilacchi et al (2008), Langer et al (2009), Lewis et al (2008) and Easton et al (2011) on an EU sponsored approach called, Integrated Railway Information System (IGRIS), developed by the InteGRail European research project. These are efforts to define a set of open industry standards to support interoperability between new and legacy information systems. The structure of the inteGRail project is shown below in Figure 13

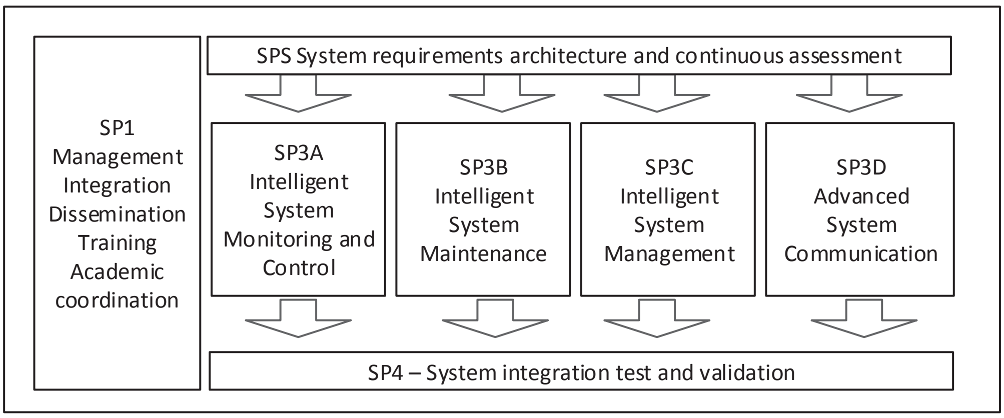

Figure 13 Structure of the InteGRail Project (Umiliacchi, et al, 2008)

The objective was to guarantee unambiguous data exchange, flexibility and scalability. This was to be made possible by having data matched to a railway based data model implemented as an ontology. This model would be able to check its own consistency and be easily expanded as new concepts or data are introduced.

The IGRIS project was designed to:

- Organise knowledge in conceptual spaces, according to its context and by using a standard model to avoid ambiguity through the use of a shared standard data model and vocabulary;
- Use of standard languages (such as SPARQL) to make queries and to represent information, which can be automatically processed;
- Improve system maintenance using tools to automatically check system consistency;
- Allow easily automated data extraction; and
- Deploy differing views of the model to different users and user groups.

The key problem to be solved was the transformation of raw data received from various sensors into information. Umiliacchi et al (2008) called this moving from a data centric approach to a knowledge and information based approach, while Langer et al (2008) call it mediation and population, they all however implement this approach through an ontology. In any case the key point is that the ontology is designed to represent knowledge about the railway assets (in these cases rolling stock and infrastructure components).

As with Hoinaru, et al (circa 2013) and Lodemann and Luttenburger (2010) discussed above the starting point for the development of the ontology was another modelling tool, in all these cases a railway domain specific XML tool such as the one developed during the EuRoMain project was used to develop the initial taxonomy. Tutcher et al (2017) use RailML for the purpose. It was noted that the need to move away from plain XML based representations was driven by the fact that no inference can be made automatically on XML data and it required domain experts to carry out this work. A key feature about the use of Ontologies is that they allow new information to be inferred from existing information automatically by the use of a reasoner and rules (SWRL).

The basic layered architecture of the InteGRail model is given below in Figure 14. It shows at a high level how the requirements mentioned would be implemented on a physical system. At the lower levels data is transformed between the sensors and adaptors. The next level is responsible for embedding the data into its proper formal relationships in a storage fusion layer. At the next layer automatic reasoning allows a very fine grained on-line analysis of this information/data. The fourth level shows that the reasoning capability will be distributed across the system of interest at various reasoning nodes which implement the same ontology model. At the 5th layer software agents can collaborate using semantic annotated interfaces and combine the knowledge represented in the integrated railway information system and the railway ontology.

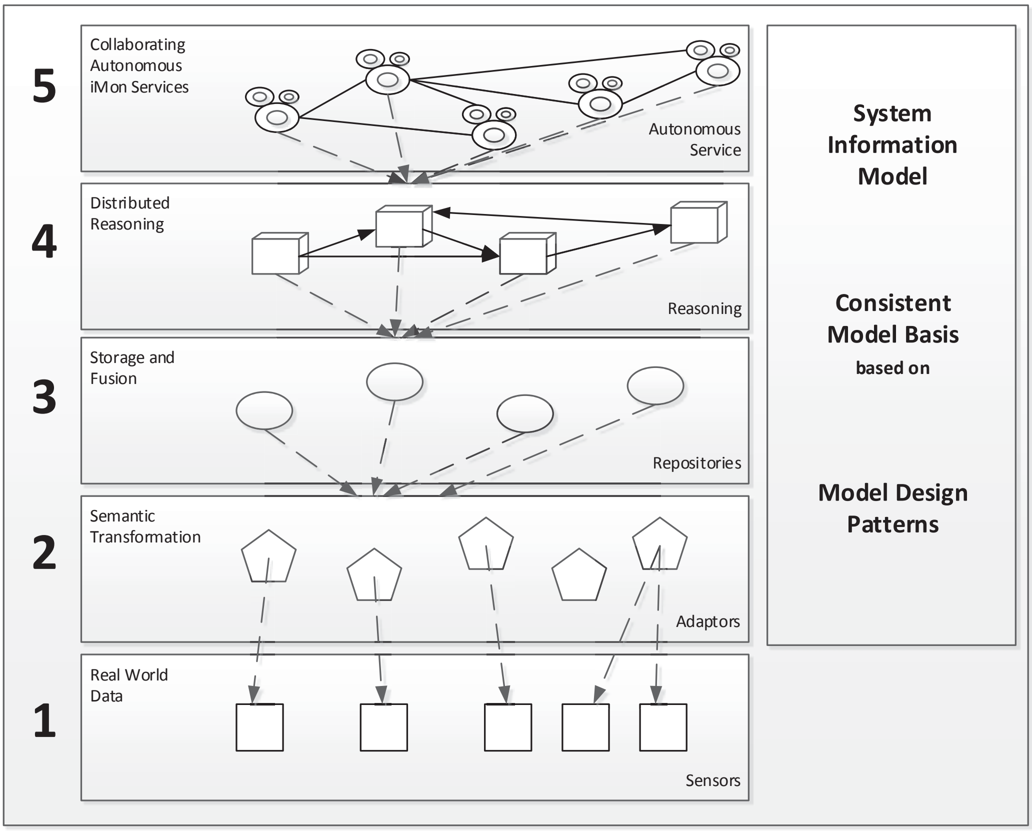

Figure 14 General InteGRail layered Architecture (Umiliacchi, et al, 2008)

This approach together with improved definitions of the measured data allows many views of the system such as mechanical dependencies, electrical properties as well as the spatial and topographical relationships that affect the system behaviour and (in this case) maintenance processes. The views referred to here are not the same as those from an EA or MBSE approach, they are machine readable views made up of relationships between objects.

As can be seen from the distributed reasoning layer in Figure 14 the railway ontology is distributed and can be spread over many different nodes. Two main ontology development steps were employed to build this structure:

- Concepts needed to describe the railway are defined in terms of their properties and their relationships. This creates the structure of the model (known as T-Box), an empty ontology or framework; and

- Generate real data applied as instances (known as A box), the form of these instances is driven by the T-Box structure of the railway ontology, this then allows data extraction with a single query language.

The implementation stage of the InteGRail project began with the definition of a Core ontology which included all the defined railway concepts and their relationships. An important aspect of this stage is that the intrinsic modular nature of ontologies allows the use non-domain specific concepts from other ontologies, concepts such as space and time. This saves time and also encourages re-use and by-in to existing terms and definitions (vocabulary). The InteGrail railway ontology is not intended to be a complete railway system model, however, again because of the modular nature of ontologies it can be extended to include more of the railway as time goes on or indeed be included in a larger railway system ontology. Although the A-box framework does not offer guidance on the railway subsystems that should/could be included.

In order to prove the overall concept, the InteGRail Project produced three demonstration scenarios (Umiliacchi, et al, 2008):

DS1. Set up and operate a new freight service between two European countries;

DS2. Cooperation between a rolling stock undertaking and an infrastructure manager by exchanging asset condition data; and

DS3. Decision support in the event of an incipient (hidden) fault on a passenger train in service.

The output requirements for the demonstration were to:

1. Prove the architecture works;

2. Prove the functionality, e.g. the information can be disseminated over the InteGRail designated infrastructure; and
3. Prove an increase in performance of the railway system based on timely information delivery and analysis.

The scenario DS3 is designed to show the improvement in efficiency of the overall railway system in the management of an unexpected event detected on a train in passenger service through the introduction of the InteGRail approach. The event in question cuts across a number of railway disciplines (operations, traffic management, rolling stock maintenance). The traffic manager has to manage the traffic on the railway with the objective of reducing the impact on the services currently in operation and those which could be operating in the future as result of a train failure. He/She must also mitigate the immediate local situation on the train itself. Detailed information has to be available to enable the operator to plan the best options for intervention and allow the rolling stock maintainer to evaluate the impact of the fault on the train performance, in order to allow the traffic manager to find the most appropriate solution (including re-configuring the routes available to other trains, detraining passengers etc.).

The fault should be detected by the Onboard Train Control and Monitoring System (TCMS). This is then communicated to the Maintenance Organisation, Traffic Manager and the Operator.

The situation is evaluated and notified to the Operational Decision Support System (ODSS) together with an evaluation of its impact on train performance.

The ODSS can combine this and other information in order to evaluate the impact of any available solution on the overall performance of the affected part of the system, so as to suggest to the traffic manager and train operator the most effective decision (Umiliacchi, et al, 2008). For revealed faults, (called Generic Faults), the rolling stock maintainer can identify preventative actions that reduce the effect on the overall service. An unrevealed fault (called insipient) that remains hidden until other circumstances combine to produce a train failure, result in the worse consequences, as preventative interventions are not planned. This process is shown below in Figure 15. The decisions available and consequences are represented in ellipses, the concepts affected are shown in rounded rectangles and the technical fault on the train is the central rectangle.

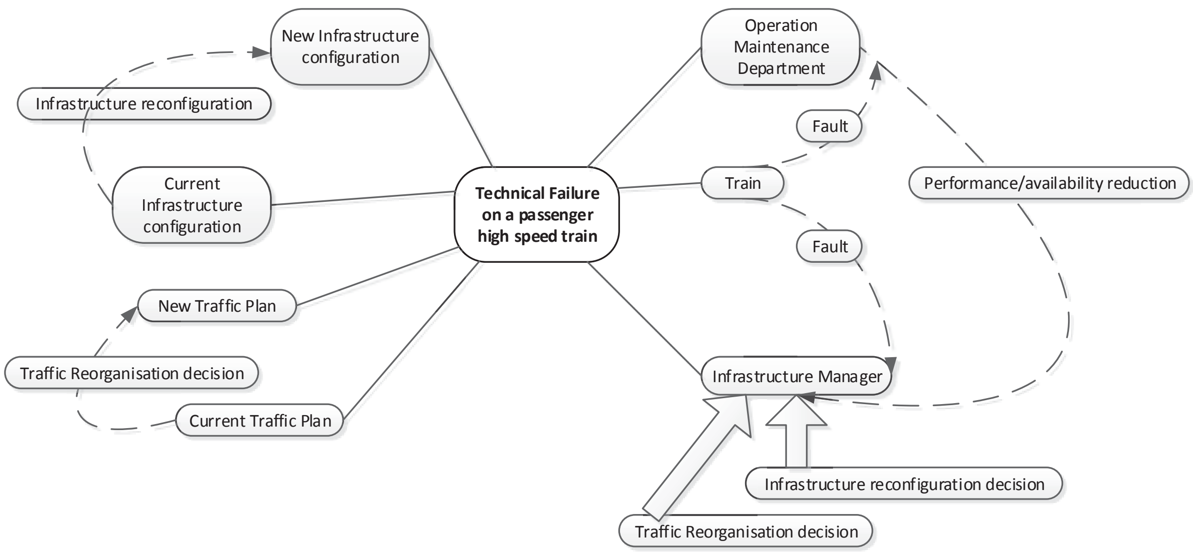

Figure 15 Principles of 3rd Scenario (Umiliacchi, et al, 2008)

There are a number of benefits from the application of systems like the InteGRail system. Enabling unified access to information and adding automated reasoning allows

previously impossible scenarios to be considered. Reducing the reliance on manual intervention can lead to better control of asset maintenance and therefore less variability. Langer, et al, *(2008)* and Lewis, et al, *(2008)* describe how OWL enables the capture of the semantics of railway RCM (Remote Condition Monitoring) data structures to represent actual and historic RCM information. They detail how an ontology (Railway Domain Ontology (RDO)) was created with a structured design methodology using design patterns extended by railway fault concepts and using the extended features of Description Logic (OWL-DL). They demonstrate how data is transformed into semantic information which can then be used as described above at a higher level by Umiliacchi et, al, *(2008)*. The objectives within the InteGRail project were *(Langer, et al, 2008)*:

- To provide a standard for information interchange that can be shared by the producer and consumer; and
- To provide a mechanism for recognising context and creating opportunities for improved performance, to be achieved while recognising and addressing issues of IPR (Intellectual Property Rights) and security.

The initial design of the RDO was driven by other parts of the InteGRail Project which were addressing the scope of the "kind of" data to integrate and the "kind of" querying to be performed. Once this was understood the types of model pattern were developed as a way of realising the ontology modelling requirements, this was based on four criteria:

- Representing the sequential relationship between physical components (to represent how infrastructure elements are connected together or how trains are made up);
- The abstract dependencies between them (to show the various elements throughout the railway are related to each other, e.g. through communication of component status);
- Their relationships with external observations (to demonstrate how measurements can be taken etc.); and

- Their mapping to concrete data types for historical analysis (to support further reasoning and inference).

Both Langer et al (2008) and Lewis et al (2008) defined their models by means of design patterns that support the specification of core ontology concepts. This lends itself to domain knowledge capture, because it enables domain experts to incrementally specialise the components in the model. An example of one of the design patterns is given below. Figure 16 illustrates the pattern for sequential component modelling. This pattern is used to represent the physical railway concepts such as tracks and routes, note it has startsAt and endsAt properties which capture its spatial properties.

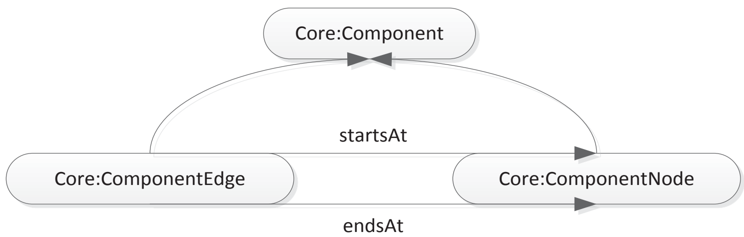

Figure 16 Sequential component design pattern (Langer, et al, 2008)

The core ontology also requires the ability to attach measurement concepts and some overall status condition. The pattern developed to achieve this is shown below in Figure 17. The same approach is used by both Langer et al (2008) and Lewis et al (2008). The measurement pattern was required to show measurement concepts. It is based on a requirement to capture implicit content and use it to infer new information; this is explained in greater detail later in this section.

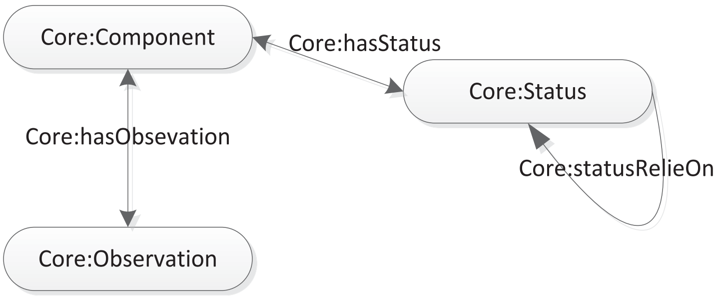

Figure 17 measurement pattern (Langer, et al, 2008)

The measurement pattern was further extended to represent relationships with symptom, fault and event level concepts. The event level concept is important because it is used to interpret quantitative data into qualitative information, it is used to provide semantic information such as high, low, hot, cold etc. (Langer, et al, 2008), which can then be reasoned over, this is illustrated by the pattern diagram below in Figure 18.

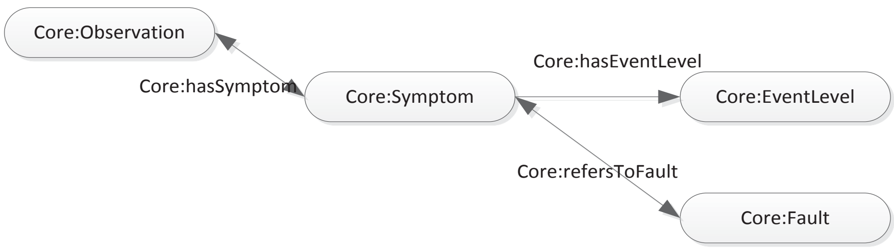

Figure 18 Extended measurement design pattern (Langer, et al, 2008)

The symptom concept is used to capture domain knowledge from experts. This enables a measurement that is outside of certain levels to infer some symptom. A second inference can then be made to some associated fault. This process is based on gathering information from domain experts.

The work described covered two key railway operations, traffic management and maintenance, a subset and specialisation of that described (Umiliacchi, et al, 2008) above. The traffic manager needs to know if a system is faulty in order to make decisions on the overall service and the maintainer requires a specialised version of the same information to plan maintenance tasks (Umiliacchi, et al, 2008) and (Langer, et al, 2008). The traffic manager only needs fault information at a high level, e.g. engine fault, wheel fault etc. It can then be implied what sort of action needs to be taken, e.g. remove a train service at the next opportunity. The maintainer however requires the information in more depth for example the specific component that is faulty, to enable maintenance to be planned.

This more detailed scenario is modelled by using a track based measuring system, which measures the wheel impact on the track and hot axles detectors which detect if an axle box has exceeded a certain temperature. Trains which have a too high axle box temperature or wheel impact force are deemed a high priority for removal from service because of the potential impact on safety and damage to the infrastructure and rolling stock.

This idea is implemented by Langer et al (2008) and Lewis et al (2008) by extending the concept of status, to PriorityStatus and NonPriorityStatus relating to the need to remove a train from service. Explicit information about the dependencies between components was captured using the statusReliesOn property as shown below in Figure 19 by Langer et al (2009) which represents a model proposed to capture the dependencies between components of a railway vehicle and Lewis et al (2008) which demonstrates a more general approach.

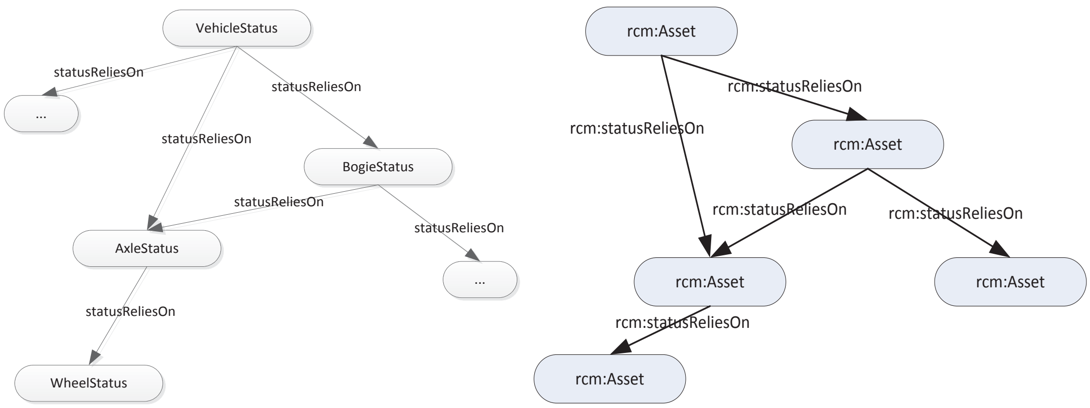

Figure 19 Dependencies between concepts (Langer, et al, 2009 and Lewis, et al, 2008)

Taking the Langer et al (2008) example, there are potential problems of having a correct vehicle status associated with a non-priority component relying on a priority component. This is elegantly solved through the "open world assumption feature" provided by ontologies. The open world assumption means that this unknown aspect of status can be used in the inference process to entail that there is a non-priority that relies on a priority. When a fact is missing it not considered to be false or an error, the component in question is thought of as being in either state (Priority and non-priority) until enough facts are available to "close the world".

When developing the System Architecture, as with Lodemann and Luttenburger (2010), Umiliacchi et al (2008) and Hoinaru et al (circa 2013), the process starts with acquiring data structures and data from legacy systems. As in other cases legacy data was mapped to some XML schema as an intermediate format. The diagram in Figure 20 is an example of such a schema, WILM (Wheel Impact Load Monitor) system. The solid rectangles represent elements that are mandatory in every data message the dotted rectangles are optional.

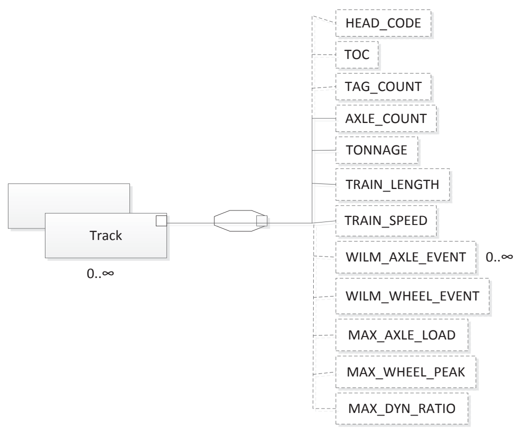

Figure 20 WILM system XML Excerpt (Langer, et al, 2008)

In the case where a WILM_Wheel_Event data element is updated, the appropriate wheel instance in the repository is updated and its status can be reasoned over. The associated

WILM_Wheel_Peak data is then used to update an associated data concept. In the case where the WheelStatus concept shown in Figure 19 is updated and if the status happens to be inferred as a priority then the associated axiom becomes true and new inference is made on the Vehicle status.

The same sort of processes are used for a Hot Axel Box Detector (HABD). The WILM and HABD both update separate ontology storage facilities. The only common aspect is that they are using the same ontology model to update their respective data stores (Langer, et al, 2008) as shown in the layer diagram in Figure 14. This is important when it comes to querying the status of a common railway vehicle. The querying application has the same ontology model and uses the result to perform its own inference.

The inteGRail project proposed distributed reasoning where each node in a network infers the status of its components in the way described above. This approach is favoured because even though the overhead is increased because it is distributed it is reduced for the querying module. An InteGRail reasoning node is shown diagrammatically below in Figure 21.

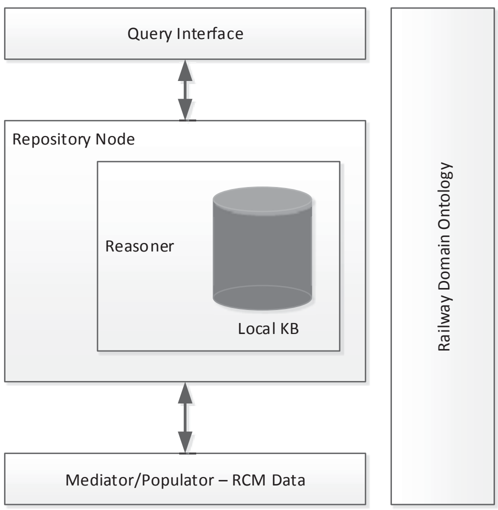

Figure 21 InteGRail Reasoning Node (Umiliacchi, et al, 2008)

The results are in RDF/XML and returned to an external application. One of these reasoning nodes is deployed at every information source, resulting in a network of interconnected nodes using a shared Railway Domain Ontology. This can be related to the overall InteGRail layered model for Umiliacchi et al (2008) shown in Figure 14.

The modelling described shows the application of an ontological approach as part of the solution to real industry (railway) problems, in terms of data interoperability. They show how a reasoner can use an ontology to integrate data from heterogeneous sources (in these cases RCM data) and also how an ontology provides a strong, semantically rich specification for the integration of existing data exchange formats.

The Open World Assumption inherent within ontologies coupled with their inferencing abilities show that where data is missing or unavailable, the reasoner will simply make the best judgement it can based on the data available.

The open world assumption means that it cannot assumed something does not exist until it is explicitly stated that it does not exist. In other words, because something hasn't been stated to be true, it cannot be assumed to be false. It is assumed that 'the knowledge just hasn't been added to the knowledge base' (Horridge, et al, 2007).

Ontology modelling within the railway industry is still current, Tutcher et al (2017) details "*a system based on semantic modelling techniques to allow integration of information from diverse and heterogeneous sources*". Which is addressing the same issues as those addressed during the InteGRail project (Umiliacchi, et al, 2008). They note in that few of the knowledge management and data modelling initiatives have met with very much commercial success, with exception of RailML and inteGRail. The main difference is it is based on a domain ontology for railways the Rail Core Ontology (RoCoOn), which has expanded from being infrastructure and signalling centric to take in more subsystems such as rolling stock and time tabling. The ontology is based on 3 use cases, which is a slightly different starting point than the previous work by Langer et al (2008), Hoinaru et al (2013), Lewis et al (2008) and Umiliacchi et al (2008). It is more in line with object oriented modelling. However, the data was still extracted from RailML as with InteGRail (Umiliacchi, et al, 2008), but this was added to with operational data from the Association of Train Operatoring Companies (ATOC) and Siemens Rail Automation in the UK. They demonstrate the applicability of their work through a passenger information system, where they show how using inferencing data from new and legacy systems can still be used, without the need for costly ITC system upgrades. It demonstrates the use of both ERTMS and legacy track circuit/Axle counter data. They demonstrate that their system can provide train position data either using ERTMS equipment (accurate) or track circuit data (can only show a train is in a section). A case is made for large cost savings with application of this sort of technology to the industry in terms system upgrades and maintenance.

Another way of using rail data is discussed by Mohan and Arumugam (circa 2012), they have produced a railway ontology to deal with the extraction and display of various operational data, that would be of use to their travelling public in India. It is based on the services offered by the Southern Railway and is intended to help the less literate of the population get information about the operation of the railway in order to purchase a ticket, based on attributes such as distance, station, arrival time etc.

Mohan and Arumugam (circa 2012) have not created a system model that covers very much of the railway, however, it does demonstrate the applicability of this sort of modelling to a very diverse stakeholder group and its ability cover different aspects of the system.

Their model is a very compartmentalised view of a railway based on a pentad (meaning group of five), SRO = <RC, RP, RI, RR, RA>, where:

- SRO is the ontology that describes the concepts and their relationships in the railway domain;
- RC is a collection of concepts;
- RP is a collection of attributes related to concepts in RC;
- RI is a collection of individuals/instances of the concepts in RC;
- RR is the collection of relations between the concepts in RC; and
- RA is the collection of axioms which are used to restrict the attributes and relations.

The concepts of the domain are the types of trains such as mail/express, rajdhani, shatabdi and fare details. The taxonomy of their ontology is shown below in Figure 22. The information in this ontology is purely for running information, as the only 2 subclasses of Southern_Railway are Fare_details and Train_Schedule. The only subclasses of Train_Schedule are List_of_Trains which leads to classes of types of trains.

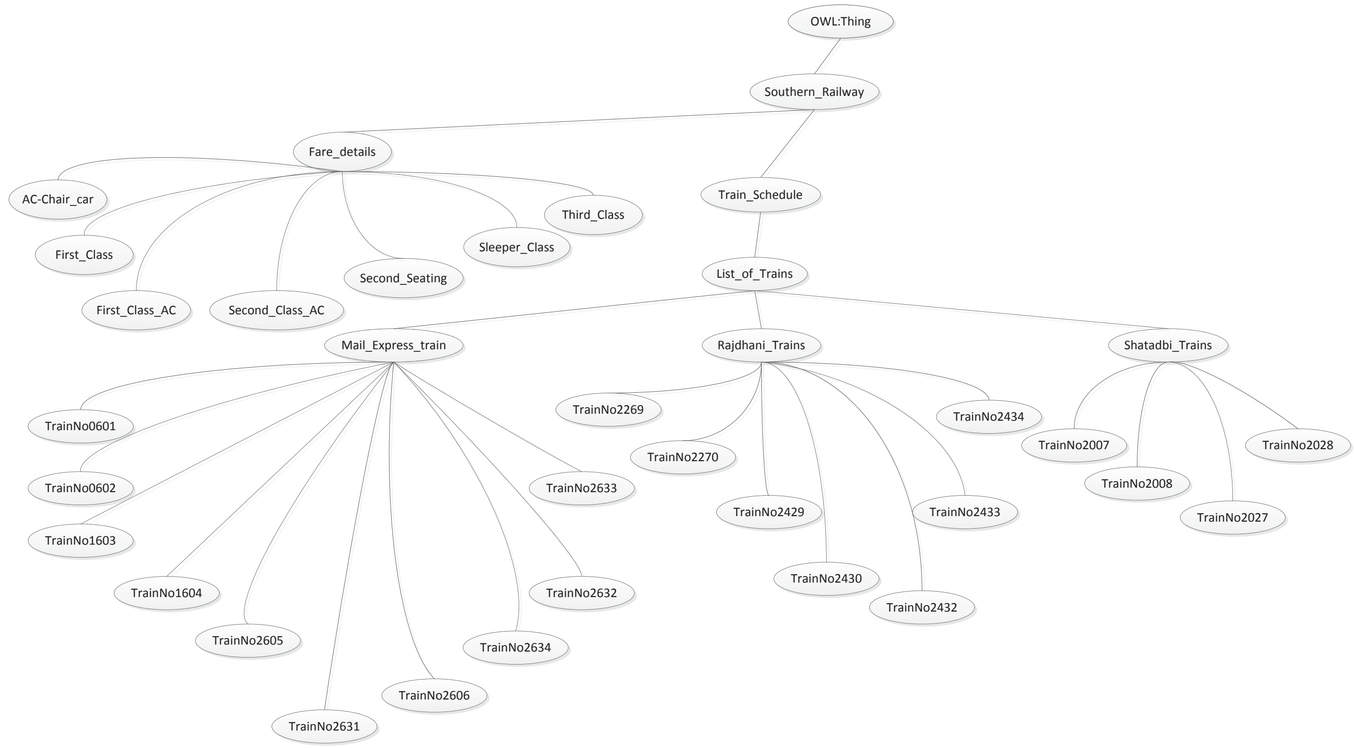

Figure 22 Railway Taxonomy for a Timetable (Mohan and Arumugam (2013)

The ontology data type properties are: hasArrivalTime; hasDay, hasDepartureTime, hasDestinationStation, hasDistance, hasRouteNo, hasRunsOn, hasSourceStation, hasStationCode, hasStationName, hasTrainName and hasTrainNo.

The object properties are: hasACChairCar_Fare; hasFirstClassFare; hasFirstClassAC_Fare; hasSecondClassAC_Fare; hasThirdClassAC_Fare; hasSleeperClass_Fare; hasSecondClassSeating_Fare; hasGeneralInformations and hasRunningInformations.

Typical individuals are: TN0601; RI0601_S01; RI0601_S02; RI0601_S03 and General_Information_of_0601.

They use a reasoner to:

- check consistency, used to identify any semantic contradiction caused by ambiguities within the description; and

- classify the ontology, to test the subsumption relationships within the class hierarchy and that classes of individuals are classified into the correct classes.

2.5.6 Enterprise Modelling

The Railway Architecture framework (TRAK)

In terms of its application in the railway industry EA is quite new the first application in the UK was TRAK. TRAK was originally commissioned by London Underground Limited. Development started in 2009 and was based on the then current views of architectural descriptions within London Underground, which were based on ISO/IEC 42010 (International Organisation for Standardisation, 2011) and the Ministry of Defence Architecture Framework (Ministry of Defence, 2018) and also tied to the systems engineering life cycle defined in ISO/IEC 15288 (International Organisation for Standardisation, 2015) (Rail Safety and Standards Board, 2011). As an EA Model TRAK is a means of describing the architecture of systems and is based on the requirements of ISO/IEC 42010 (DfT 2013). This framework allows a system architect to describe an enterprise, a concept, a solution (and its procurement) and an architecture tasks. An Enterprise has stakeholders who have concerns that need addressing and these are addressed through viewpoints which make up the architecture description. As mentioned above this approach comes predominantly from business and business systems point of view, with particular attention to managing change and major projects.

TRAK is based on 21 viewpoints associated with five perspectives. The perspectives and their viewpoints are as follows:

1. Enterprise: This perspective describes the enterprise in terms of its goals and the enduring capabilities that support those goals. These are high level business

requirements that everything else contributes to and they should form part of the long term strategic thinking of an enterprise. It is interesting that over the recent past, particularly in large projects (like infrastructure upgrades) the industry has been looking for ways of reliably relating solutions/technical requirements back to business requirements. The viewpoints that together describe this perspective are:

a. Enterprise goal;
b. Capability Hierarchy; and
c. Capability Phasing.

Typical stakeholders whose concerns would be addressed here are:

d. Owner;
e. Developer;
f. Planner; and
g. Maintainer (of the enterprise).

2. Concept: provides a logical view of what is needed in response to the capabilities required by the enterprise in question and identified through the enterprise perspective. This perspective does not dictate a solution (it is necessarily technology free), it describes the logical connection between various parts of the system, an example being the connections between a service control centre and other parts of the infrastructure without actually saying how these can be achieved. It also does not describe any particular lifecycle; it covers everything from conception to disposal. The viewpoints associated with this perspective are:

a. Concept need;

b. Concept;

c. Concept item exchange;

d. Concept Activity to Capability mapping;

e. Concept activity; and

f. Concept sequence.

Some typical stakeholders here are:

g. User;

h. operator of the concept; and

i. solution and enterprise stakeholders.

3. Procurement: this perspective gives a top level view of the procurement of a solution to satisfy the enterprise capability needs outlined in the enterprise perspective and developed in the concept perspective. It shows how projects might deliver solutions that are described in the solution perspective. Time dependencies between projects owing to dependencies on systems being introduced or removed are also captured here. It also shows changes in responsibilities over time. This is a good example of the change management bias of this approach. The viewpoints associated with this perspective are:

    a. Procurement structure;

    b. Procurement timeline; and

    c. Procurement responsibility.

    The stakeholders are likely to be

d. Acquirer;

e. Developer;

f. Builder.

4. Solution: describes the solution, it covers the parts of the system be they human or machine, their exchanges and protocols. It also provides a view on how organisations and equipment are organised and their governance. It provides a description of how logical requirements defined in the concept perspective and the capability requirements defined in the enterprise perspective are realised. The language and terms used here are oriented towards business systems and software development. The viewpoints associated with this perspective are:

   a. Solution structure;

   b. Solution resource interaction;

   c. Solution resource interaction to function mapping;

   d. Solution function;

   e. Solution function to concept activity mapping;

   f. Solution competence; and

   g. Solution viewpoints.

   The stakeholders are:

   h. Owner;

   i. Acquirer;

   j. Developer;

k. Builder;

l. Maintainer; and

m. Trainer.

5. Management: describes the architectural tasks and those relationships that are common across the other perspectives. It provides a way of representing the scope and findings of an architectural task and provides structure to the modelling. This is also the place where any requirements and nominative standards that apply are described. All of the stakeholders described above are also stakeholders in the management perspective. The following viewpoints are associated with the management perspective:

    a. Architecture description dictionary;

    b. Architecture description design record; and

    c. Requirements and standards.

Redacted

Figure 23 Structure of TRAK - formed from 1 metamodel, 5 architecture perspectives and 21 architecture viewpoints (Wikipedia, 2018)

The basic structure of TRAK is shown above in Figure 23.

Although TRAK does not mandate a detailed modelling process or the minimum views that must be produced for any architectural task, it does have a bare bones process to conform to TRAK as follows:

- Agree architectural task scope with task sponsor and stakeholders and record in the appropriate viewpoint, the viewpoint includes the constructs needed to capture the task scope (this again demonstrates TRAKs origins in business system change management, which is not to say it is not useful in other areas);
- Chose the appropriate TRAK architecture views for the task (it is up to each organisation or project to choose those they think most appropriate);
- Create Architecture description; and

- Close out architecture description with the appropriate management viewpoints to capture the findings of the modelling.

It is important to say here, that TRAK (or any other framework) will not specify:

- How to plan for the models needed or to be developed for the task;
- How to organise the structure of the repository;
- How to model; and
- How to organise views for readability or ease of navigation.

Part of the logical definition of TRAK is the metamodel (Plum N. , 2018). The TRAK metamodel defines the object types (stereotypes) and the relationships that can appear within the TRAK architecture viewpoints and therefore the architecture views. It defines the language TRAK uses to describe the real world. The metamodel is shown below in Figure 24.

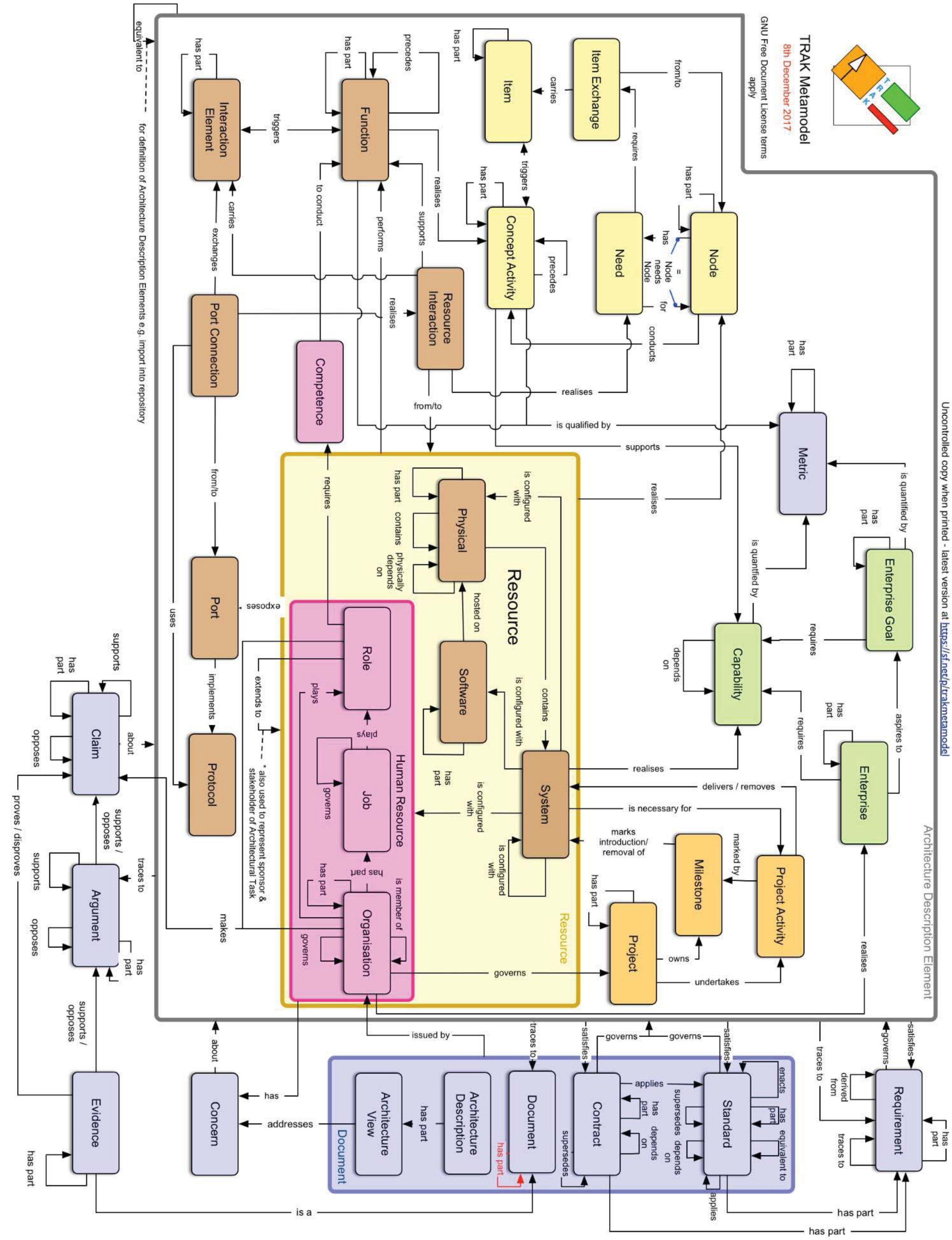

Figure 24 The TRAK Metamodel - defines Allowed Object Types and Relationships for Use in TRAK Architecture Descriptions (Plum N. , 2018)

TRAK has been formally adopted by the UK Department for Transport who chair the TRAK Steering Group that manages the overall direction, strategy and formal releases of TRAK it has been used on the following projects (Wikipedia, 2018):

- Sub Surface Upgrade Programme (SSUP). Upgrade of signalling and rolling stock for Circle, Hammersmith, Metropolitan and District lines on London Underground;
- Technical Strategy Leadership Group (TSLG). Railway Functional Architecture; and
- Rail Safety & Standards Board (RSSB). UK Railway Functional Architecture.

It should be noted that, although TRAK is implementation agnostic, RSSB notes that support is available through Sparx System Enterprise Architecture UML modelling tool (Rail Safety and Standards Board, 2011). UML support, rather than SysML points to EAs origins in business system change management as opposed to systems engineering.

2.5.7 Railway Functional Architecture (RFA)

The Railway Function Architecture (RFA) research project was commissioned by RSSB on behalf of the Technical Strategy Advisory Group (TSAG). TSAG commissioned this work as part of the development of Technical Route Maps which began 2008 the goal of which was to identify potential improvements in cost, capacity, carbon emissions and customer focus. The work was conducted between December 2009 and September 2010 under project T912 (Rail Safety and Standards Board, 2011).

The main objectives of the Railway Functional Architecture were to:

- Identify the technology based functions that must be performed for a modern railway to operate;
- To support elements of the railway that were amenable to the adoption of
    - Greater use of COTS technology;

  - Greater use of plug and play technology;
  - Open systems architectures;
  - Lean systems;
  - Smaller systems;
  - Rationalising system procurement; and
  - Shorter product development lifecycles.

The project was biased towards identifying improves that could be gained from better Information and communication (ICT). However, as the work progressed it began to look at railway processes, activities and standards.

The RFA model is a slimmed down version of TRAK it consists of primarily the following perspective:

- Enterprise perspective: primarily dealing with the goals of the railway system and the capabilities that support those goals. What does the system do and what does it need to able to do meet those goals;
- Concept perspective: deals with the logical functions that support the capabilities and the connections between them, this what the system has to able to do. It is solution agnostic and focus on the operation of the railway. It also does not focus on what organisations are delivering capabilities;
- Solution perspective: this perspective starts to become more technology and operationally specific as it deals with how a solution may be achieved.

The relationships between the 3 principle perspectives used in the RFA is given in Figure 25.

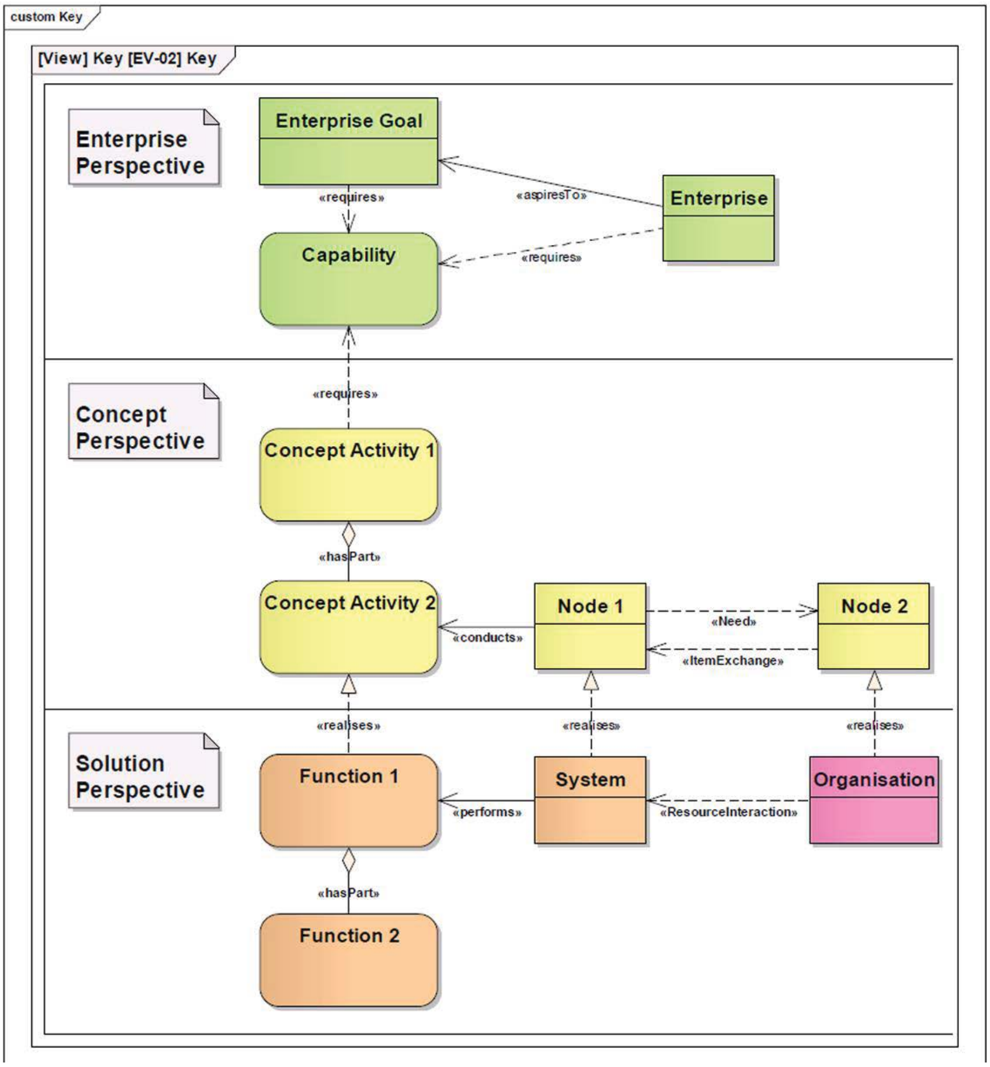

Figure 25 RFA Perspectives (Rail Safety and Standards Board, 2011)

The RFA also uses a subset of the TRAK diagrams, for more information on these see T912 Railway Functional Architecture Summary Report (Rail Safety and Standards Board, 2011) and (Plum N. , 2016).

The RFA summary support states "*it appreciates that the model is incomplete in certain areas, or not detailed to its full complexity*" it is expected that it will be added to and improved over time. It also states "*the solution perspective has been developed in order to demonstrate the benefits of architectural modelling with the RFA and to illustrate by way of examples the* *actual systems and organisations that deliver the railway today. It is not intended to be complete*", (Rail Safety and Standards Board, 2011).

An example of how the model handles the situation where a particular function can be implemented in more than one way is given below in Figure 26Figure 26 RFA example of optional functions , It shows the function required is to communicate a movement authority to a train and all of the options for delivering that function.

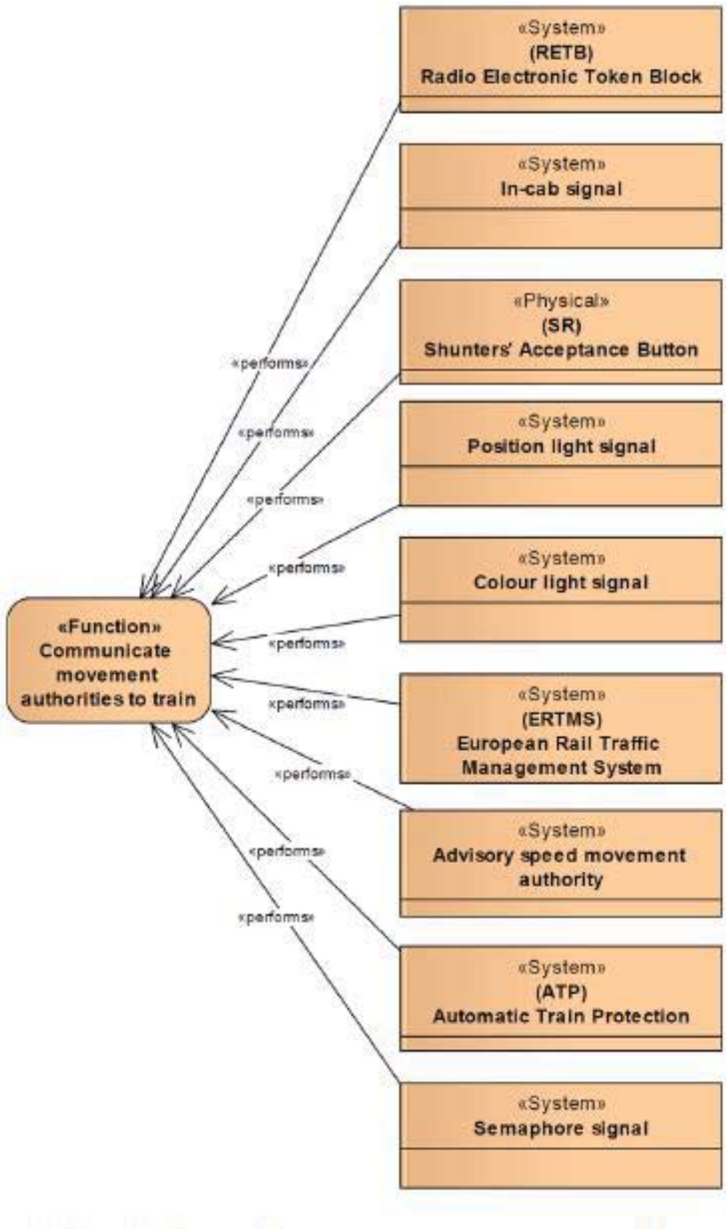

Figure 26 RFA example of optional functions (Rail Safety and Standards Board, 2011)

This above is very much an operational view another more technical view is given in Figure 27Figure 27 RFA view of subsystem configured with GSM-R . This shows a subsystem, in this case the GSM-R Subsystem and the other subsystems that it could possibly be configured with in a railway context.

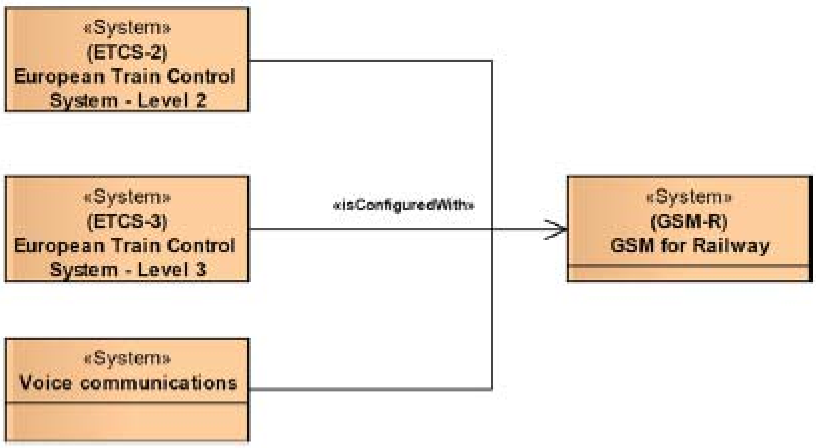

Figure 27 RFA view of subsystem configured with GSM-R (Rail Safety and Standards Board, 2011)

For other examples of the application of RFA can be found in T912 Railway Functional Architecture Summary Report, (Rail Safety and Standards Board, 2011).

2.5.8 Network Rail Digital Railway Enterprise Architecture

The Network Rail Digital Railway project is a programme of business change designed to improve and modernise the railway network in the UK. It requires a whole system approach because it needs to model the state of the system as it is and then a new modified state in order to understand how to get from one to the other. It is the subject a case study for the development of a whole railway system EA. Umiliacchi et al (2018) highlight that it is important to understand and design a system as a whole and in its environment and this includes business, processes, people and technology and all of their relationships. They propose, a framework for applying EA with systems engineering. As with both TRAK (Plum N. , 2016) and RFA (Rail Safety and Standards Board, 2011) Umiliacchi et al (2018) give the need to align solution/technology to business needs and the ability to assess the impact of changes to the system as the driving force for their work. The major difference between the framework proposed by Umiliacchi et al, (2018) and TRAK and RFA is their framework is ontology based. This brings together two emerging approaches, although both these have software/computer system/data modelling background in the rail industry at least. The Umiliacchi et al, (2018) framework proposes the ontology base to provide the precise language required when modelling such a complex system. A UML diagram describing the underlying ontology is given in Figure 28.

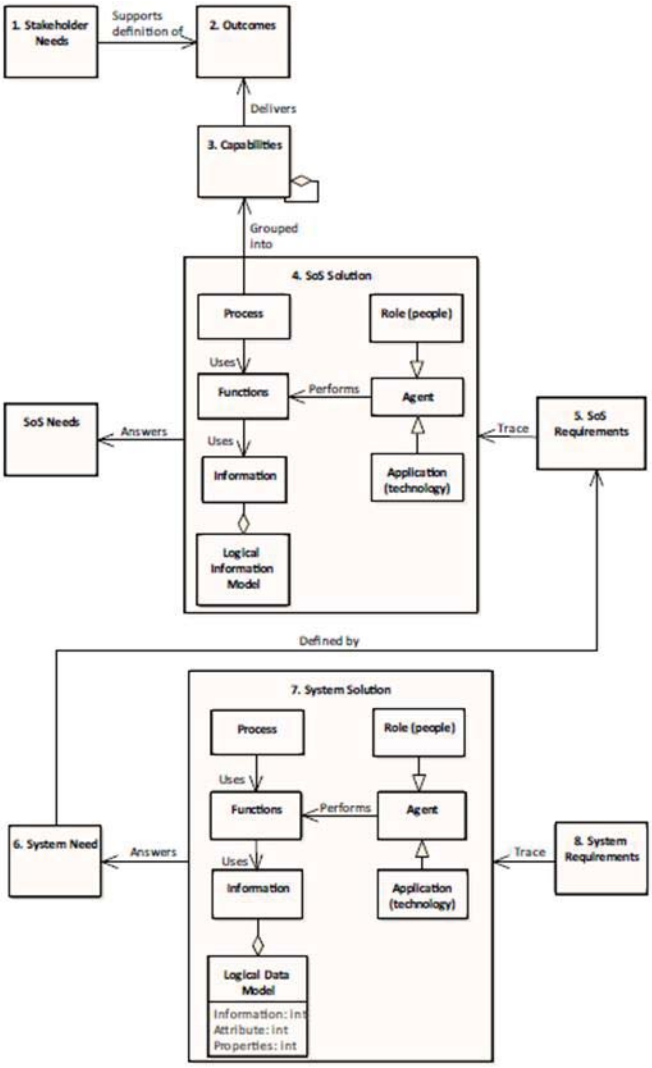

Figure 28 Ontology based EA framework for Network Rail Digital Railway (Umiliacchi, et al, 2018)

It is claimed that the ontology is a whole systems model that allows business needs to be transformed into system of system requirements (Umiliacchi, et al, 2018). Although the paper does not demonstrate the whole system model it does detail how this might be achieved. The ontology provided in Figure 28 is a meta model for the EA modelling process. The graphical (UML) description of the ontology helps to overcome the greatest obstacle this author found with ontology modelling in general, which is difficulty in visualising and therefore understanding large complex systems when using tools like Protégé.

Requirements (called capabilities) are generated from business needs as shown in Figure 28. Umiliacchi et al (2018) also note that these requirements are use case driven which are based on scenarios, this approach suits mission type requirements but is not as useful for non-functional requirements such as performance, safety etc. These types of requirements do not form part of this demonstration. Also the current demonstration does not yet model the dynamic behaviour of the system.

From Figure 28 it can be seen that the stakeholder needs are delivered through outcomes, which in turn are delivered through processes. Processes use functions which are performed either by human or computer agents. Processes are organised into groups. Umiliacchi et al (2018) also group process in a time horizon, which is a period of time over which a decision or other piece of information is expected to have a direct impact.

Clearly this work is at an early stage and so there are few demonstrations available.

2.6 Conclusions

The examples given earlier in this chapter demonstrate some of the versatility of the UML and SysML modelling languages and the overall object oriented approach to system modelling. They are versatile in terms of how they can be applied, modellers can use all or a subset of the diagrams in the creation of their systems model. They have the advantage of being able to capture both structure and behaviour/functionality as demonstrated particularly in Ferrogalini (2015). It also works well with other approaches, as shown in the Digital Railway EA (Umiliacchi, et al, 2018) where UML is used represent the underlying ontology method/meta-model. SysML is a particularly useful language as it can capture mathematical modelling and requirements. However, although it can model dynamic behaviour in terms of representing it with activity diagrams, state machines and sequence diagrams, it cannot on its own simulate that behaviour. It therefore needs to work with other tools such Matlab Simulink etc. UML particularly and to a certain extent SysML lack rigor particularly in terms of mathematics and logic in their language. This makes them less useful in verification and validation work against standards and specifications on their own. Although in SysML at least this has been addressed to some degree by parametrics (Peak, et al, 2007 and Ferrogalini, 2015). The above examples show this is overcome in all instances by the use of other formal methods such as the B method (Bousse, et al, 2012), or Petri nets, (Sana, et al, 2010) alongside UML and SysML. Also the examples have demonstrated how SysML in particular can be made to model complete systems, with many differing views. Another advantage of using SysML or UML is the large body of literature, tools and training that are commercially available.

Although there will always be the need for the use of other models, databases and tools SysML seems to be to provide the central underlying model that can be used to tie all of the others together.

On their own SysML and UML are just modelling tools and to be really useful in modelling large and complex systems they need to be part of a larger approach, such as MBSE or EA, (Delligatti, 2014, p. 5).

Research on the use of ontologies in the railway industry is ongoing at the time of completing this thesis (2019). The main thrust is still towards system and data interoperability and the advantages that can be gained from improvements in the way information systems are used within the industry (Tutcher, et al, 2017). The Railway Domain Ontology is a significant contribution as it provides base ontologies for others to work from (Moris, et al, 2015), although Umiliacchi et al (2018) query its readiness for more general use. There is also current research into how ontologies can tailor information for specific passengers (Gould, et al, 2017).

Research in this area is being driven by continued sponsorship of projects by government or government backed organisations such as the EUs ST4RT project. This research project is on the subject of sematic ontology based automation of transformations between hetrogeneous data formats. It is part of the Shift2Rail interoperability framework (Shift2Rail, 2018).

Undoubtedly all of the major contributions from ontologies are underpinned by some form of system model. Although not an Ontology, RailML is nearest to a whole systems model and is expanding (Nash, et al, 2010) and ontologies such as RaCoON use it. In the majority of cases the modelling has been done to achieve particular purposes, such as data interoperability, ERTMS requirements and test specifications etc. and their expansion has been driven by the need to solve these issues.

So far a complete whole railway system ontology model has not yet been completed. However, there seems to be enough evidence to suggest that using ontology modelling, as part of an integrated approach with other models it could be achieved. The use of other models with it would be particularly useful to give a graphically based easier to understand representation. The Digital Rail project is working on this aspect (Umiliacchi, et al, 2018).

As far as the railway industry is concerned EA is a very new approach. Although it does have a lot in common with UML/SysML type approaches. Its chief advantages are that it is an integrated approach much like MBSE and the demonstrations describe the whole railway from differing viewpoints (particularly as it is business centric). This is of particular interest to this project. At the time of writing this thesis EA is still very much a business/ICT based approach. This can make some of the language and terms used a little difficult for those who do not have that background. This could be a barrier to commercial success, as decision makers could be put off before they understand the advantages. Most of the demonstrations underline this. It is very likely that as the user group gets larger and more diverse this will change. This point is made by the three demonstrations above, TRAK, RFA and Digital Railway. They show how EA is being adapted to the railway industry over time as practitioners become more skilled in its use and the industry requirements become clearer. Using UML or SysML to represent EAs, as in Figure 26, Figure 27and Figure 28enlarges the group of people who can easily adapt to its use. None of the EA examples found demonstrate an attempt to actually model a whole railway system although all state such a model is their aim. All of them are also biased towards process or IT type systems, they are also missing a guide to what a railway is or the railway needs to be. They are developed for specific issues and therefore reflect just the particular railway or part of the railway affected.

3 Research Question and Requirements

It has been established in chapter 1 that there are clear benefits to the application of systems engineering, particularly in large complex projects including GTS' (Elliott, 2014), (Dunford, Yearworth, York, & Godfrey, 2012) and (Honour, Axelband, & Rhodes, 2004). However, there is still very little application of systems engineering on current major projects in the GTS environment. Although there has been little GTS specific work done on why this may be, Dunford et al (2012), Beasley (2012) and Davidz et al (2005) identified a number of reasons for this same problem in Rolls Royce and the US defence industry respectively, that are applicable here, including:

- Perceived costs and added effort with the associated additional time to complete a project;
- Logistical issues of applying processes and methods across departments, collaborating companies and national and international geographical separation;
- Lack of a flexible and appropriate approach to the application of systems engineering within a company;
- Lack of experience of engineers in the application of these techniques;
- Projects do not allow enough time for systems engineering to occur or don't allow a budget; and
- Resistance to change in companies, particularly if they are a successful company already.

The literature survey has also hi-lighted the need and the space, as none really exists at the moment, for developing a general whole system GTS modelling methodology and guide. This can then be used to overcome some of the difficulties identified by Dunford et al (2012), Beasley (2012) and Davidz et al (2005), such as creating a common understanding of the system and a common approach to systems engineering and therefore deal with the logistical issues mentioned above. Also by making the process guided and flexible it would help to overcome the issues of lack of experience and the perceived added costs and time to projects.

Although the literature survey covers a number of system approaches, they are driven by a particular need or problem to solve. They are also quite discipline specific and do not help understanding of the GTS as whole within its environment, or drive a common approach. This can lead to emerging properties of the system not being identified until later in a project, where they become much more time consuming and expensive to fix (Elliott, 2014).

The chief focus of this project and thesis is therefore, to produce a methodology for modelling whole GTS', that addresses some of the barriers identified above through having a common understanding of the system in general and a common approach to modelling it see Chapters 3 and 4.

The methodology developed for this thesis will be general so it can be of use to the largest possible group of stakeholders. If adopted by a large stakeholder group logistical barriers and barriers introduced by a lack of common understanding will be reduced and managed. It also should be able to represent a GTS from the broadest possible number of views, from business, finance, technical, logistics, passengers, governing bodies, neighbours, connecting systems and the environment. This would reduce the number of separate unconnected models required. The model will be expandable in terms of breadth (the amount of GTS covered) and depth (the detail covered) so as to manage the amount of systems engineering applied on a project and so reduce the perception that systems engineering adds cost and time to a project where it is not necessary. The methodology should be expressed in such a way as to reach the largest audience and be open to users. It should guide the user through the building of models to aid the capture of necessary building blocks and information and also help engineers with less experience.

The efficacy of the methodology developed will be demonstrated through partial modelling of various diverse elements of a GTS, against the methodology. The methodology will then be trialled on a representative body of railway biased engineers to ascertain its general usability and acceptability within the industry see chapter 6. The results of this empirical evaluation will then be discussed in the wider context of the barriers identified above in chapter 7.

To answer the questions posed above, the development of the methodology shall be guided by a number of high level requirements as follows:

1. The modelling methodology and models themselves shall be re-usable, e.g. applicable to different GTS' at different times. This will allow the approach to more easily be introduced across departments, companies etc and drive the common approach and understanding needed for successful application of systems engineering. Being reusable also reduces application cost and time overheads;
2. Both the methodology and the models themselves shall be extendable in length (be able to model more railway) and depth (greater levels of detail) in order to make them relevant as new systems become available and legacy systems expand. This also reinforces reusability and helps to manage the logistical, cost and time issues identified above;

3. Where possible the methodology shall allow the inclusion of existing quantitative and qualitative models from other sources, to allow complete descriptions to be formed within the models themselves. There are two advantages to this, firstly it promotes reuse of existing models making sure engineers see that their work is still valuable, which helps deal with resistance to change and secondly provides cost and time savings through less repetition;
4. The methodology shall allow and encourage the use of data from existing models to avoid re-inventing already existing and adequate models;
5. The methodology and the models themselves shall be sufficiently open and transparent to allow others to use and add to them to encourage users to “buy in” to the approach and feel ownership through being part of the development process; and
6. The methodology shall encourage and the models themselves shall enable the production of outputs that are readily understandable across disciplinary divides e.g. common representation.

To encourage the maximum use of the approach it is the intention that this modelling approach and generic model of a GTS will be of use to:

- those responsible for operating GTS’;
- those responsible for regulation;
- politicians and their appointees (looking at investment, social issues etc.);
- Engineering functions;
- Logistics functions;
- Day to day operations;

- Neighbours; and
- Commercial concerns (including spending and investment).

4 Design

4.1 Methodology

In order for a process to be used by others it needs to have a structure which, when worked through allows the user to arrive at the goal of that process. So to this end a 5 stage process was developed which is based on the various points on the left hand side of the V life cycle model (Forsberg & Mooz, 1998). The methodology does not always map directly because some stages, notably stage 2 model organisation, relate to what the author believes is good modelling practice. So whilst they are not system requirements as in that they do not refer to the system being modelled, they are modelling requirements.

Making sure that the model is set up with the right initial information is paramount for getting the most out it. The first stage is the setting up process. This stage can be thought of as the concept stage of the V life cycle, where thought is required about what is actually going to be modelled and who are the key customers/stakeholders. Therefore, this stage is designed to identify:

- the key stakeholders (others may follow during later stages);
- what kind of GTS is to be modelled (from a very high level e.g. is it a high speed service, suburban, metro etc); and
- the environment in which it will be operated (e.g. country, city, mountainous etc).

The next most important part of process is organising the model. A GTS is a large and complex system with many interacting systems, subsystems and components. It also requires many views of the model for all of the various stakeholders. Getting this in place early is going to dictate ease of navigation around the model, it's re-use and how easy it is to understand the model and system. Also reorganisation later on in the process, as the model is being developed, can be costly, time consuming and error prone. This stage also maps loosely to the concept stage of the V cycle. The output from this stage is a package diagram.

The third stage maps to the very beginning of the system requirements part of the V cycle. At this stage there will be enough information to set the context within which the GTS will operate. It identifies the systems, people, organisations and “things” that will interact with the GTS system. Also the GTS is identified as an Enterprise, which is the subsystems, people and processes that come together to allow the GTS to deliver its purpose, see Umiliacchi et al (2018). The Stakeholder list is revisited to allow for new Stakeholders to be identified from this more detailed view of the system and its operational environment. The system model boundaries are also identified at this stage. The outputs from this stage are a Block Definition Diagram called the Context diagram and the Views and Viewpoints for the Stakeholders.

The fourth stage is where the bulk of the Requirements work is done. This also maps to the system requirements stage of the V cycle. This is because the requirements process is an iterative process, that begins with understanding what is being delivered to whom and where (stage 3 of the process). Then the requirements elicitation work can properly begin (this stage). New Stakeholders are identified and their Requirements are recorded along with those already recorded. Operational scenarios are developed and critical system properties, design constraints and black box structure are driven from the identified Requirements. The Requirements process also starts the system decomposition process of the model. These processes are explained and demonstrated later in this and the next chapter.

The fifth stage is where the system is decomposed layer by layer until the correct level of abstraction is achieved. This stage maps to the apportionment of system requirements and design and implementation stages of the V life cycle. By the correct levels of abstraction, it is meant that each of the Requirements has been met through a single atomic argument. At each level of abstraction, the critical system properties are examined to understand if a further level of abstraction is required to deliver the level of information needed. An example of this is system reliability, there are choices that can be made, reliability can be obtained at a system, subsystem or component level depending on the data available and the requirements of the particular Stakeholder.

The process encourages constant iteration and decomposition of the requirements. Each requirement is traceable back up to the top level or business requirements. This is analogous to the right side of the V life cycle model, ensuring validation against the original requirements thus ensuring that the right model of the right system is being delivered.

The methodological approach to the development of a generic model and any applications that follow it, is shown in the activity diagram below in Figure 29. The 5 stages are shown in the swim lanes. The derivation and purpose of each stage are explained in the following chapters.

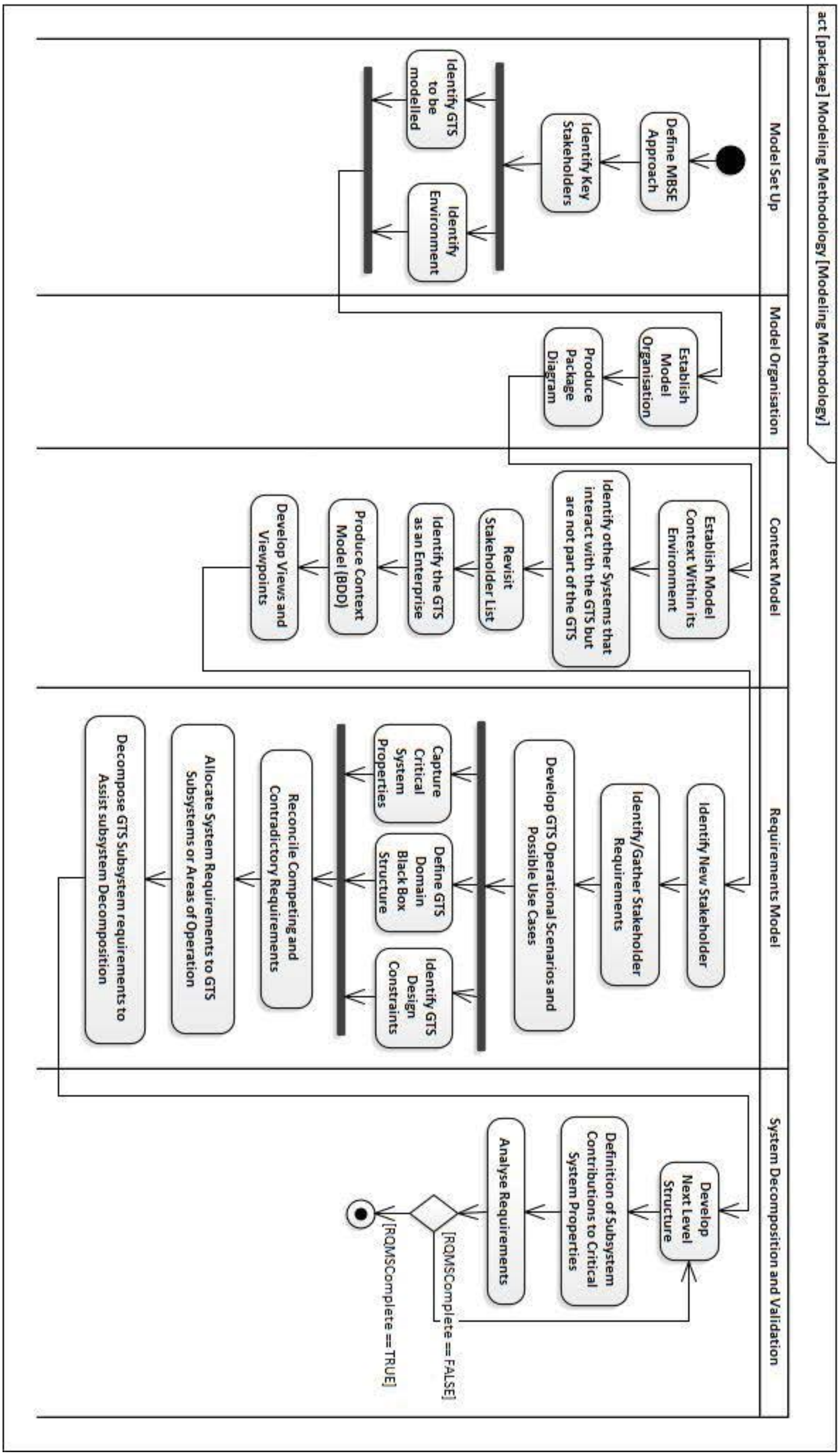

Figure 29 MBSE Modelling Approach (Author)

4.2 Model organisation

4.2.1 Purpose of model organisation

Due to the number of subsystems, stakeholders, operations, logistical requirements, public and political interest and amount of data generated and required, it is imperative that the model is organised in some way to facilitate:

- Understanding of the whole model including how the various elements can and do work together;
- Reuse and sharing of model elements including mathematical models, value types, definitions, specifications and data;
- Navigability among model elements allowing easy access to elements that can be copied, reused or shared as well existing/persistent data;
- Support for configuration management;
- Exchange of modelling information with other tools; and
- Maintenance of the model itself.

4.2.2 Model Organisation as part of a Model Based Systems Engineering (MBSE) Approach

The MBSE approach drives model organisation, the most important output of an MBSE approach is a coherent model to ensure the benefits listed above are realised (Freidenthal, et al , 2015, pp. 428 – 431 and Estefan, 2008). Organising a complex model of this nature is really partitioning the various major parts or subject areas of interest, of the system of interest, into logical groups. This then allows the various stakeholders to understand their own part in relation to the whole and also the various development teams to work on the parts that are of interest to them, while understanding the impact on the whole. Defining this effective model organisation is an iterative process and it is also subjective, one person's idea of a logical and coherent organisation and partitioning may not be the same another person’s. Friedenthal et al (2015), suggest a number of possibilities as follows:

1. By System hierarchy (e.g. system level, subsystem level, component level etc.);
2. By process lifecycle, where each model sub package represents a stage in the process (e.g. requirements analysis, system design etc.);
3. By teams working on the model (e.g. requirements team, product design team, validation team etc.);
4. By model elements that are likely to change together;
5. By model elements organised to support reuse;
6. By other logical or cohesive groupings of model elements based on predefined modelling criteria; and
7. A combination of the above.

The MBSE approach developed for this project is a combination like that suggested in 7 above comprised of 1, Systems hierarchy; 5, model elements organised to support reuse, users, external systems, constraints and operations.

4.2.3 SysML Model Organisational Features

One of the key reasons for using SysML as the modelling language for this project is that it supports MBSE, therefore by definition the creation of coherent system models and therefore systematic organisation. The fundamental unit of organisation in SysML is the package. Packages contain model elements, such as diagrams, value types, constraints etc. allowing related elements to be housed in the same place. Packages can also be nested into a hierarchy further specialising the logical grouping of model elements. Packages also aid in the necessary practice of providing unique names for identifying individual elements. They do this by using what is called a fully qualified name of the form, package::sub package::element_name. This is vital for a system model that in reality would be used by many people from different disciplines. Thus giving rise to the possibility that they could use the same name to identify different model elements. An example is a Train Management System (TMS); for Rolling Stock it is the onboard computer subsystem that handles some control functions, fault recording and reporting etc; but for a Command and Control system it the subsystem that controls where trains are and where they will go. Even if an element's name is the same as another element contained in another package it will be unique because of the rest of the path name. Package hierarchies are very similar to the file structure that commonly exists on a computer. There are also special types of package predefined in SysML such a model library, where shared items such as value types and constraints can be kept and shared across the model. Views are another package type where stakeholder views of the system model are kept. These will be explained further, where they are applied.

4.2.4 Criteria for Model Organisation

There is no identified correct way to organise a SysML model as noted earlier. Therefore, a number of criteria for model partitioning will be derived. This will be based on the following analysis which is based on the competing needs for easy navigation, reuse and usability.

Some of the overall structure of a GTS is generally known, see section 2.3. Therefore, an element of the organisation of the model needs to consider the general physical structure of the system. This needs to be at the 'system of systems' level right down to the individual system, subsystem and component levels (where appropriate) as advocated by Mori et al (2018). However, there are a number of other important aspects of the model other than the physical structure that should be taken into account.

The GTS has to be understood in relation to the context within which it functions. It is therefore inevitable that the other systems and elements with which the system interacts should be gathered in one place.

In almost all instances a GTS is bound by legislation, standards etc. across its physical structure, its operations and its interfaces with the environment it operates within. As these elements of legislation and standards are related to model elements across the system (indeed many deal with interfaces between various elements) it is, therefore, desirable to gather all these items in one place, where they can be accessed from across the model.

There are Requirements, physical, functional and performance related to all levels of the model, from the GTS system level itself down to component level. The GTS level Requirements can influence more than one area of the GTS; therefore, these Requirements benefit from being organised in the same place. These system level Requirements will have relationships to Requirements at lower levels stored in association with the individual areas of the GTS Domain decomposition. This shows how these system level Requirements are complied with by complying with the lower level Requirements.

Stakeholders can have various system level and subsystem level concerns that need to be addressed. They also need to understand how their concerns can affect the rest of the system, so logically stakeholder views at least could be organised in the same place.

General system level safety issues also relate to all parts of the GTS, safety issues usually consist of requirements, the various ways of meeting these requirements, actual demonstrations of compliance, evidence of compliance and in the railway industry at least, some way of capturing lessons learned and past data. Again at a system level at least these can be organised together to aid understanding of how safe the system is as a whole and how the various parts of the system contribute to safety.

Finance, is another area of interest that can span the system and individual logical operational elements of a GTS. This is particularly apparent in situations like the UK mainline railway, where train operators, train owners, infrastructure managers and maintainers of both infrastructure and rolling stock can all be separate companies. They all need to manage their finances and do financial reporting. This is balanced against the needs of governing bodies and political entities to manage and report on the whole. There is an argument for keeping the system level information in one place, with relationships to the particular areas that make up the functioning GTS.

The environment within which the GTS exists is an area which the GTS has no control over, but it non-the-less exerts an influence in terms of Constraints and Requirements on the GTS and all of its components. The same environmental element often influences many different parts of the GTS (the weather, populations, topography, road access and even gravity being just a few examples). These issues should be gathered together in one place and then related consistently to other elements across the model.

Logistics in terms of maintenance, storage, access facilities and central management are issues that can be viewed both as a local issue, i.e. where does the infrastructure manager store their maintenance equipment or the location of rolling stock maintenance depots etc.; and as a GTS wide issue for example does one maintainer maintain track across a number of areas, a rolling stock operator may store and maintain rolling stock at depots that belong to others. There are many other examples of shared facilities, working and management.

Organising at least some of the higher level logistical issues in one place with, where appropriate, relationships to other elements within the model is desirable.

There are other considerations too, which relate to modelling itself rather than to the GTS system, such as reuse of modelling elements. Some elements will almost certainly have relevance across the whole model. Value types where a quantity and a kind is defined, such as power (measured in Watts) or current (measured in Amperes) etc. should be defined and stored in a central place so they can be shared. The same can apply to mathematical models, which in SysML are called constraint blocks, these and any other element likely to be shared should be collected in one place at the highest level so they are available to all.

Operations is an area of interest where almost everything comes together, where trains go, where they stop, how they are protected and what is transported to where and how. It is therefore desirable that all aspects of running the GTS are kept in the same place.

It should be noted that model organisation is likely to differ from project to project due to their size, shape, goals etc. The important point is the modeller should be producing a model organisation at this point, not later in the process. Given the analysis and discussion above the author has derived the following organisational criteria for a generic model of a whole GTS system:

Organisational Criteria 1: All of the elements associated with the GTS model, including those that are not actually part of the GTS, shall be contained within a GTS Domain package and organised in a sub package hierarchy.

Organisational Criteria 2: The GTS exists within the wider environment and must be understood in that context. Where systems and other elements interact with the GTS, these should be captured and contained in one place. Therefore, there shall be a package called

GTS Context which contains these model elements as sub packages. The GTS Context sub packages shall, for this demonstration, be Public Power Supply System, Neighbours, Public Access, Public Communications and Legislation. Others can be added if a particular application requires it.

Organisational Criteria 3: A GTS is comprised of a number of particular physical subsystems. The organisation of the GTS model should facilitate easy identification, access, navigation and understanding of these. Therefore, the model elements for each identified physical subsystem are contained within a particular sub package. These shall be sub packages of a GTS Subsystems package within the top level GTS Domain package structure.

Organisational Criteria 4: Each physical subsystem package will have a number of sub packages. The model elements that represent the system component hierarchy and the details of how the components will work together are contained in a GeneralComponents sub package. Requirements model elements for the individual subsystems that feed up to top level Requirements, or just address the functioning of the individual system or operation in question, will be contained in a Requirements sub package. The operational and maintenance model elements of the individual sub systems are contained within an Operations&Maintenance sub package. There are also likely to be different types associated with each individual subsystem, for example different rolling stock types (EMUs, electric locomotives, DMUs, Diesel locomotives etc.), signalling solutions (conventional, ERTMS, CBTC etc.) and traction power supplies (25kV AC or 1500 or 750V DC etc.). Model elements that represent these different types shall be stored in a Types sub package of each subsystem sub package. Elements that represent individual interfaces with other subsystems or operations will be contained in an Interface sub package. Also model elements relating to the financial aspects of the individual subsystem in question will be stored in a Finance sub package of the subsystem sub package.

Organisational Criteria 5: Reusable model elements such as mathematical models (constraint blocks), Behavioural Models (Activities) Definitions and Value Types shall be contained within a Model Library sub package in separate sub packages called Value Types, Activities, Definitions and Parametrics in order to aid reuse and sharing of concepts. There is also a package to contain elements that relate to the modelling process itself called ModellingProcess. The Model Library sub package is situated in the top layer of the GTS Domain package.

Organisational Criteria 6: Each stakeholder will have one or more viewpoints to represent their concerns in the system of interest, these shall be contained within a View and Viewpoints sub package at the top layer of the GTS Domain package hierarchy.

Organisational Criteria 7: Model elements that represent GTS system level Performance shall be contained in a separate GTSPerformance sub package at the top level of the GTS Domain package hierarchy. These behaviour elements shall be separate packages within the Behaviours package called Performance and Safety.

Organisational Criteria 8: In many cases GTS's have to be compliant with various legislation, regulations, standards and codes of practice. These standards emanate from various bodies and geographical locations, those of particular interest in the UK usually come from Europe and the UK, however on occasion other countries produce standards that are of interest, for example the USA. All the legislation, regulations, standards and codes of practice applicable to a GTS shall be contained within a first layer sub package of the GTS Domain package, called Standards&Legislation, with sub-packages for British, European, International and Other.

Organisational Criteria 9: The way the GTS is operated produces operational processes and a number of logistical solutions. These bring all the various subsystems together to achieve the common purpose. Therefore, all the model elements that represent operations and logistics aspects that apply to a GTS as a whole will be gathered in an Operations sub package. This Operations sub package shall be situated in the top level of the GTS Domain package hierarchy. The Operations sub package shall contain two sub packages called Operational Procedures and Logistics.

Organisational Criteria 10: For this project there will be two sets of Requirements, one set for the modelling process and another set that will reflect the general requirements for GTS's. These will both be sub packages of a Requirements package situated in the top layer of the GTS Domain package hierarchy called GTSRequirements and ModelRequirements. There is a third package in the Requirements package call Requirements Process, which holds elements that specify the approach to requirements.

Organisational Criteria 11: Model elements that represent finance associated with GTS's as a whole are stored in a Finance sub package situated in the top layer of the GTS Domain package.

Organisational Criteria 12: There are a number of Environmental aspects that impact on a GTS. Elements that represent the environment are contained within an Environment package stored in the top layer of the GTS Domain package.

The package diagram is given in Figure 30.

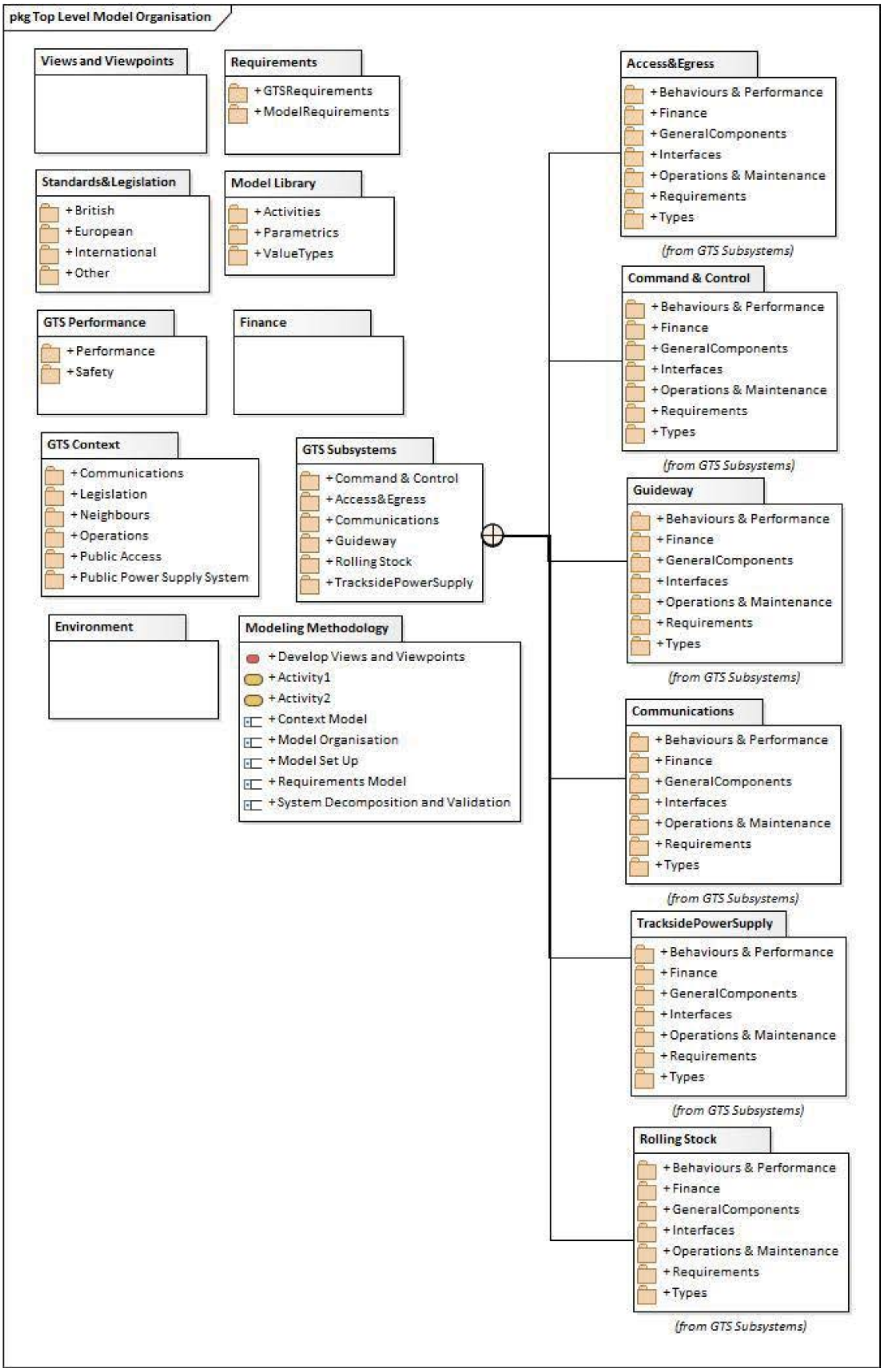

Figure 30 Package diagram (author)

4.3 Context Model

4.3.1 Purpose of the context model

All the stakeholders need to be identified and their concerns captured. A model based approach to this is to build a context model that represents the GTS as an Enterprise in the context of the world in which it is deployed (Freidenthal, et al, 2015, pp. 65, 66 and 302 and Weilkens, 2007, p. 5). By setting the model in the context of the outside world the context model helps to define the system modelling boundaries, the Stakeholders and the models relationships with the outside world. An Enterprise is defined as an aggregation of the systems elements, users and operations that collaborate together to achieve a set of mission objectives. Also a Stakeholder is defined as any person or organisation that has an interest in the system across its lifecycle (Freidenthal, et al, 2015, p. 433).

The purpose of this process therefore is to help a modeller or project identify all the Stakeholders and their possible concerns, to establish the system boundaries and the models relationship to the outside world. It achieves this by setting the system of interest in the context of its environment.

4.3.2 Context Diagram in SysML

The context diagram will take the form of a SysML Block Definition Diagram (BDD).

The top level block is GTSOperationalDomain, this represents a GTS in the context of its operational world. It is composed of a number of other blocks which together completely define a GTS context and represent various aspects of a GTS' world that are not directly part of the GTS, or are part of the model, but impact on it.

4.3.3 GTS Enterprise

The context diagram as a whole is shown in Figure 32. For convenience the GTS Enterprise block is enlarged and shown Figure 31.

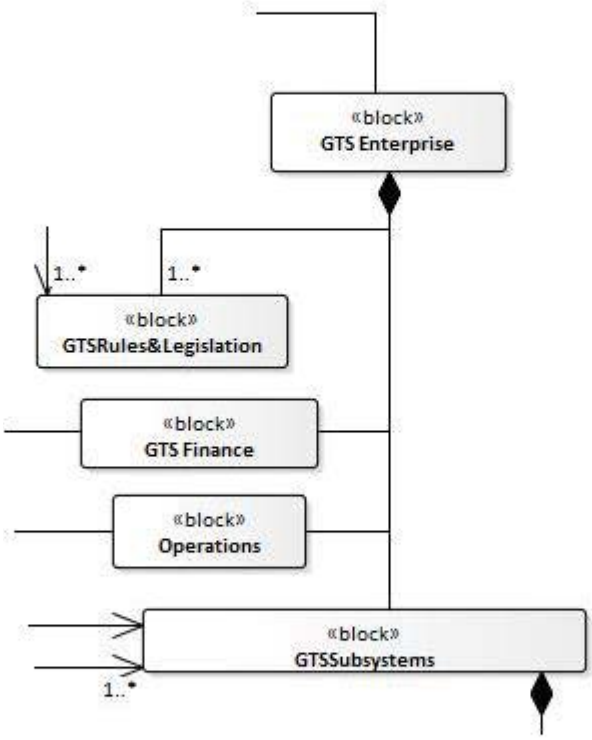

Figure 31 GTS Enterprise block (author)

GTSFinance represents the financial aspects of the GTS subsystem and is therefore part of the GTS enterprise. It also has a bi-directional relationship with the Stakeholders block. This represents the fact that Stakeholders are interested in the finances of the GTS. The relationship is shown with the Stakeholders block and not with the blocks that represent the actual stakeholders themselves lower down the block hierarchy. This is to capture the fact that many of the stakeholders have a relationship with the GTSFinance block and make sure that the diagram is still readable. The actual nature of the relationships is not elaborated on at this stage.

The GTSOperations block represents all those aspects of a GTS that enable its operation as a whole. It is part of the GTS enterprise as it is a key enabler for the GTS to meet its mission. It also has associations with the Stakeholder Block in terms of the interest many of the stakeholders have in its operation:

- various railway undertakings, such as infrastructure managers and train operators, represented through the RailwayUndertakings Block as part of the Users Block. These are interested in terms of how they can run their businesses;
- other businesses who may use the GTS to transport products or whose customers and workforce use the GTS to get to their premises. They are represented by the Businesses Block as part of the Users Block;
- A Passengers Block that represents other passengers is also part of the Users Block;
- Political entities such as:
    - local politicians who may see the GTS as a local asset (or not);
    - National politicians who may see the GTS as part of a larger transport policy;
    - Pressure groups interested in everything from green transport, the country side and passenger experience.

  They are represented by the NationalPolitician, LocalPolitician and PressureGroup Blocks which are all part of the Political Block;

- local residents can be interested from a quality of life point of view. Is there excess noise, does the service improve their access to other centres, does having a transport link close by improve the value of their property. There may also be other GTS' close by which can have their operations affected by a reduced customer base or EMI effecting their systems. These are represented by Residents and OtherGTS Blocks that are part of the Neighbours Block;

- Governing bodies also have an interest in how the GTS is operated in terms of the service it provides and safety and customer satisfaction. They are represented by the GoverningBodies Block;
- Manufactures and developers may also be interested in how the service will operate so they can better design products for a particular service pattern. They are represented by the Manufactures&Developers Block.

The relationship is therefore shown, for the purposes of the context diagram between the Stakeholders Block and the GTSOperations Block. It is a 1 to many (1..*) uni-directional relationship because many stakeholders can have a relationship with any one GTS.

There is a uni-directional relationship between the GTSRules&Legislation Block that is part of the GTSEnterprise and the CountryRules&Legislation Block. This is to capture the fact that the GTS rules and legislation are usually a subset of the country wide rules and legislation. Its direction indicates that rule changes at the country level can and do have an effect on the rules and legislation at the GTS level. The multiplicities are one or more at both ends to represent the number of rules, standards, codes of practice and other legislation that interact at both ends of the relationship.

The GTS enterprise is composed of physical subsystems as defined in 2.3. The GTSSubsystems Block is part of the GTSEnterprise Block. The GTSSubsystems Block is composed of a number of Blocks based on the necessary and optional subsystems identified previously.

The GTSSubsystems Block owns a Command&Control Block that represents signalling, train protection and control, the multiplicity of this relationship is zero or more. This is to enforce the generality of the model and to capture the fact that a GTS in its simplest form does not need a signalling system to be a GTS and only a GTS. However, it is likely to have one in a commercial setting. Also it is zero or more rather than zero or one because at any point a larger GTS may have more than one command and control system in place due to upgrades, maintenance and renewals.

A necessary subsystem of a GTS is its guideway system. This is captured by the GuidewaySubsystem Block being part of the GTSSubsystem Block. It has a multiplicity of one at each end of the composite relationship because a single GTS is unlikely to have more than one guideway subsystem, although it is possible to change this if the situation changed (part of the model flexibility Requirement). To ensure generality the Block is called GuidewaySubsystem and not TrackSubsystem, this way the model can be applied to trolley bus and people mover type GTS' as well as conventional railways and Tramways.

A GTS must have rolling stock in order for it to be a GTS and only a GTS, this is represented by the RollingStockSubsystem Block being part of the GTS Subsystems Block. It has multiplicities of one at the system end and one or more at the composite end. This arrangement captures the fact that one GTS can have many rolling stock assets, but must have at least one. The various types of rolling stock and other aspects are captured elsewhere as modelling progresses, all that is being captured here is that any GTS will have one or more rolling stock elements.

A communications subsystem is not essential for a system to be a GTS and only a GTS. In the modern world however, most GTS' have some form of communications subsystem, that is at least for internal GTS use. This is captured by the CommunicationsSubsystem Block, which is part of the GTSSubsytems Block, its multiplicity of zero or more captures the fact that it is not essential and also that there can be more than one communication subsystem deployed within the GTS.

There must be at least one access and egress point for a GTS even if it just goes around a single track and starts and finishes in the same place. This is captured by the Access&Egress Block that is part of the GTSSubsystems Block and its multiplicity of one or more, the one indicating that a GTS must have an access and egress point and the 'or more' indicating that there can be any number of further access and egress points belonging to the GTSSubsystem Block.

A trackside power supply system that provides traction power is not an essential system, this is because not all GTS' are electrified, some rely on diesel power for instance. It is possible in a depot situation, or in places like the Thameslink changeover point at City Thames Link Station in London (750V DC and 25kV AC) that there is more than one traction power supply, where a train arrives using one power supply, changes over and leaves on the other. Also there are other trackside power considerations, such as power for signalling equipment. This could be derived from a local provider or even solar power modules. It is represented by the TracksidePowerSupplySubsystem block that is part of the GTSSubsystems block. It has a multiplicity of one or more representing the fact that even if it is not actually essential, in reality it is and there could be more than one.

Although system boundaries will be different for every project, the generic context diagram given in Figure 32, will act as a guide to the modeller to where boundaries to the system might be and lead to further investigations. A generic set of systems boundaries based on the authors experience has been derived in 2.4.

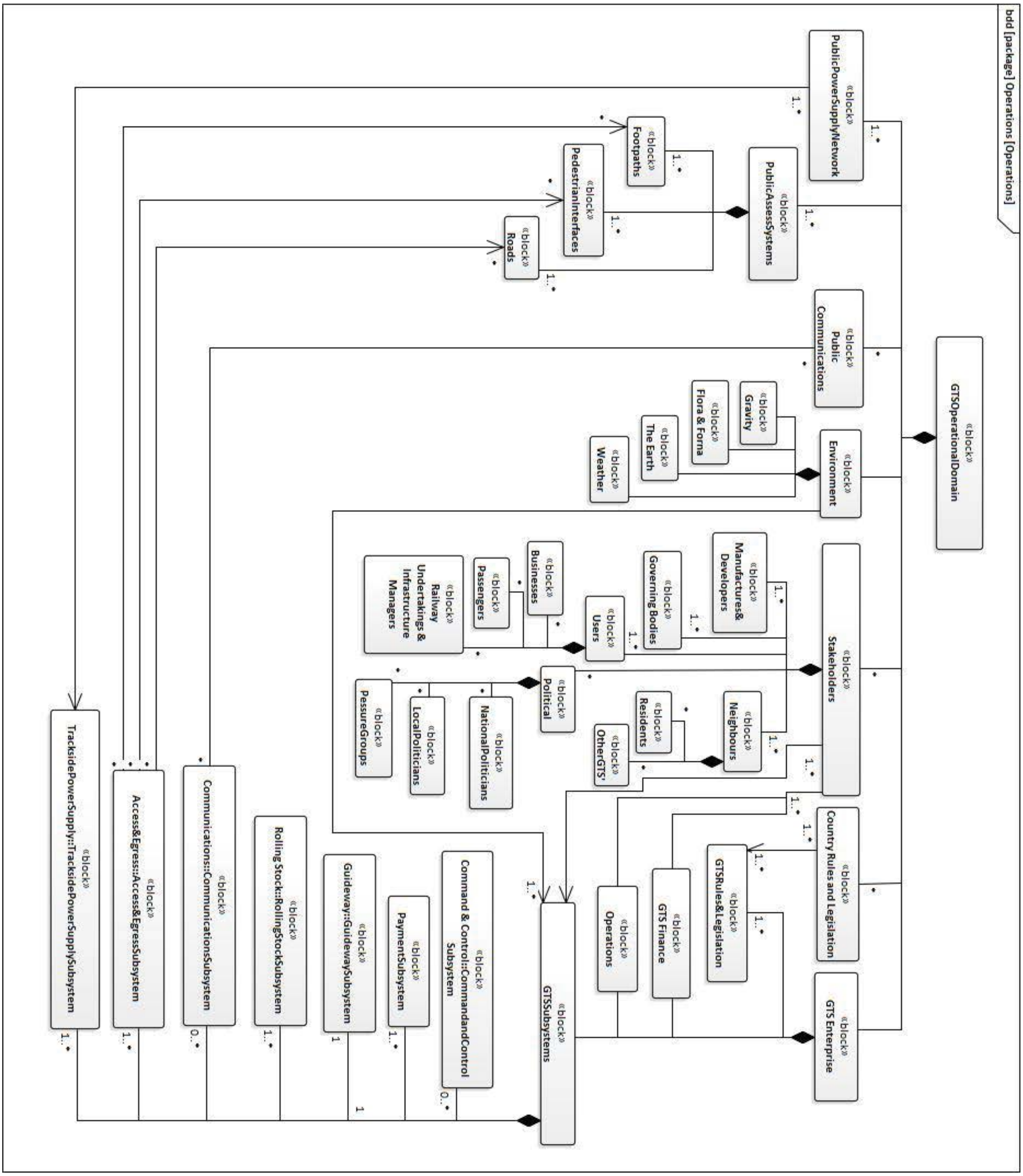

Figure 32 GTS Operational Domain Context Diagram (author)

4.3.4 Stakeholder Views

The other major output of stage 3 are stakeholder views. Stakeholder's have different concerns, requiring differing views of the underlying model. SysML allows a modeller to deal with these by Views and Viewpoints, which are consistent with ISO-42020 definitions. Using this definition allows a familiar starting point for modellers used to Enterprise Architecture (EA) approaches. This stage of the modelling process involves the modelling entity interacting with the identified stakeholders and developing Viewpoints and Views that cover each of the stakeholder's concerns.

A Viewpoint describes a view of the model that a Stakeholder or set of Stakeholders needs by articulating particular concerns they might have. This is done by providing a method that identifies the artefacts that represent the information that the Stakeholder needs.

The Viewpoint specifies:

- The purpose of the Viewpoint;
- The Stakeholders and concerns that are addressed;
- How the View content should be expressed, for example the modelling language used;
- If appropriate the file format required as an output;
- How the information is presented, for example text, graphs etc.;
- The method of producing the artefact.

A View specifies the model content that is to be presented to the Stakeholder. Views are said to expose model content and conform to a Viewpoint.

This is shown in Figure 33 with a dotted line from the FinanceView to the FinanceViewpoint Blocks, with an open arrow head at the Viewpoint end. The View is said to import the necessary SysML elements, although this may not be an automated process.

It is important to note that although a view is a SysML construct that exists within the SysML model the artefacts produced (and also how they are produced) from the View potentially live outside the modelling environment (Freidenthal, et al, 2015, pp. 98, 99 and 378 - 381). How an artefact is constructed is detailed in an owned behaviour belonging to the Viewpoint in the method section as shown on Figure 33. This can be informally expressed as a guide to producing the artefact manually as in Figure 33, or in a formal language that can automatically produce the required artefact. A view for a Finance function is given in Figure 33. The detail in the Viewpoint explains the concern that is being dealt with, the need for data for financial reporting, the languages of the final artefact will be English (to provide the narrative), SysML to get the values and Microsoft Excel for data analysis and graphing etc. The purpose is defined as; use data to create financial reporting in line with contractual requirements. The Stakeholders are the Company accountant and Finance director. Note there is no data as yet in this generic model. A SysML model can have as many Views and Viewpoints as required by Stakeholders, it could take many Views to satisfy Stakeholders with complex concerns, equally a number of Stakeholders can use the same View as in Figure 33.

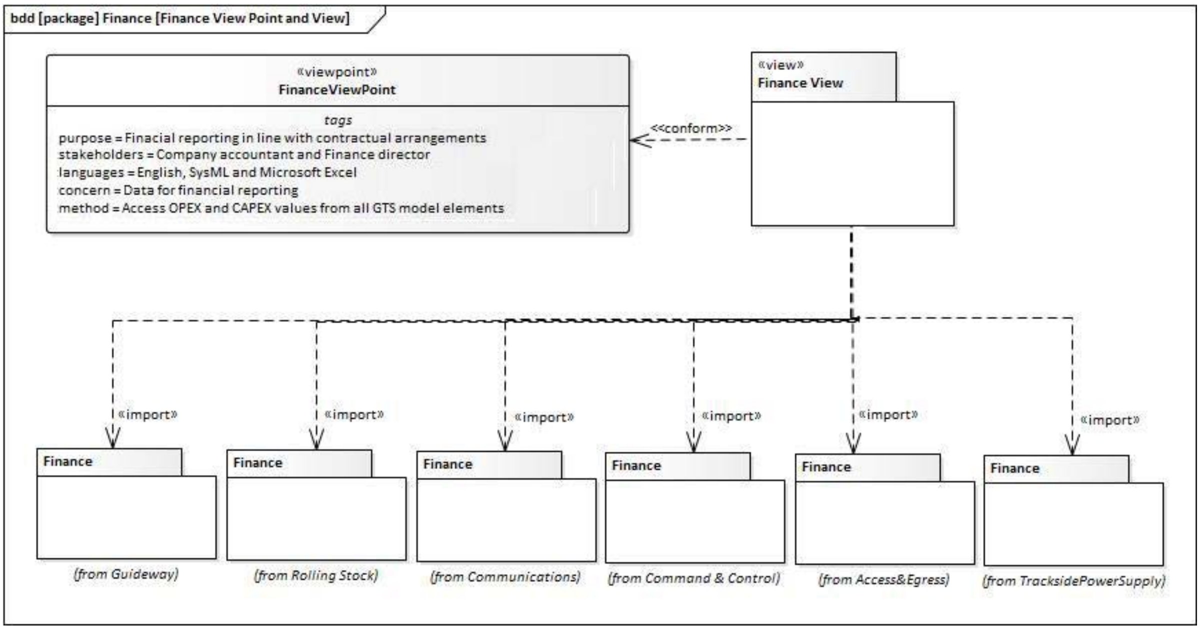

Figure 33 Finance Stakeholder View (author)

4.4 Requirements

4.4.1 General

Stage 4 of the process is where the requirements for the model and the GTS begin to be exposed. This is probably the most important phase in the development of any system, it is the process by which the requirements of the system are collected and developed. It is vital that these are captured and documented so they can be analysed and validated with stakeholders and allocated to parts of the system, to ensure that everyone involved knows what they are getting.

Requirements can be decomposed from very high level requirements (more like mission statements) to very low level technical requirements (for example those that detail how a component will deal with certain inputs or what it will be made from etc). The important thing is to be able trace that low level technical or operational requirement right back up to its associated top level requirement to ensure it is part of delivering the system that the stakeholders expect. These points are stressed in a majority of the literature that deals with requirements some that are particular relevant to this thesis are Freidenthal, et al, (2015), p. 309 and Weilkens, (2007), p. 46.

4.4.2 Requirements process

For reasons of time and space it is not the intention to define a complete and sufficient set of requirements for a complete GTS in this thesis, the number of requirements could reach thousands, see Department of Transport (2015), Department of Transport (2016), and Association of Train Operating Companies (2016). What is done in this chapter is to develop a process by which requirements can be elicited, modelled and tracked through a complex system, in the context of a whole system model supported through MBSE, with particular reference to the GTS environment. The objective is to provide a process to develop a generic set of requirements at a sufficiently generic level to guide a modeller who is applying the process to any GTS.

The process developed here is a systematic top down approach similar to those defined in Freidenthal, et al, (2015) p. 442 and Weilkiens (2007) p. 46 with differences driven by a GTS' structural, commercial and operational environment and the allocation of requirements to systems and operational areas. It details how the requirements are elicited, how they are organised, how they are traced through the system from top to bottom and how the MBSE approach supported by the SysML modelling language can support the process. The process advocated by Freidenthal, et al, (2015) p. 442, shown below in the SysML Activity Diagram Figure 34, is based on the Object Oriented System Engineering Method (OOSM) approach. This approach shows its origins in Object Oriented software development, developing requirements for a black box system and then defining system state machines. It focuses on mission scenarios, looking to identify system constraints and critical system properties straight away, the process can be applied effectively where the system environment is less complex than that of a GTS, which has many stakeholders with competing expectations and requirements. In the approach advocated for this thesis, shown below in Figure 36 Stakeholders are dealt with very early on as many of them can have a critical input into the operational scenarios applied to the design of the final GTS. There is also a focus on requirements and system decomposition this is to get traceability of the requirements from top to bottom through the many layers of sub systems and components that characterise a GTS. Also the focus is to be as generic as possible to allow the application of the process to any GTS project at any level/stage.

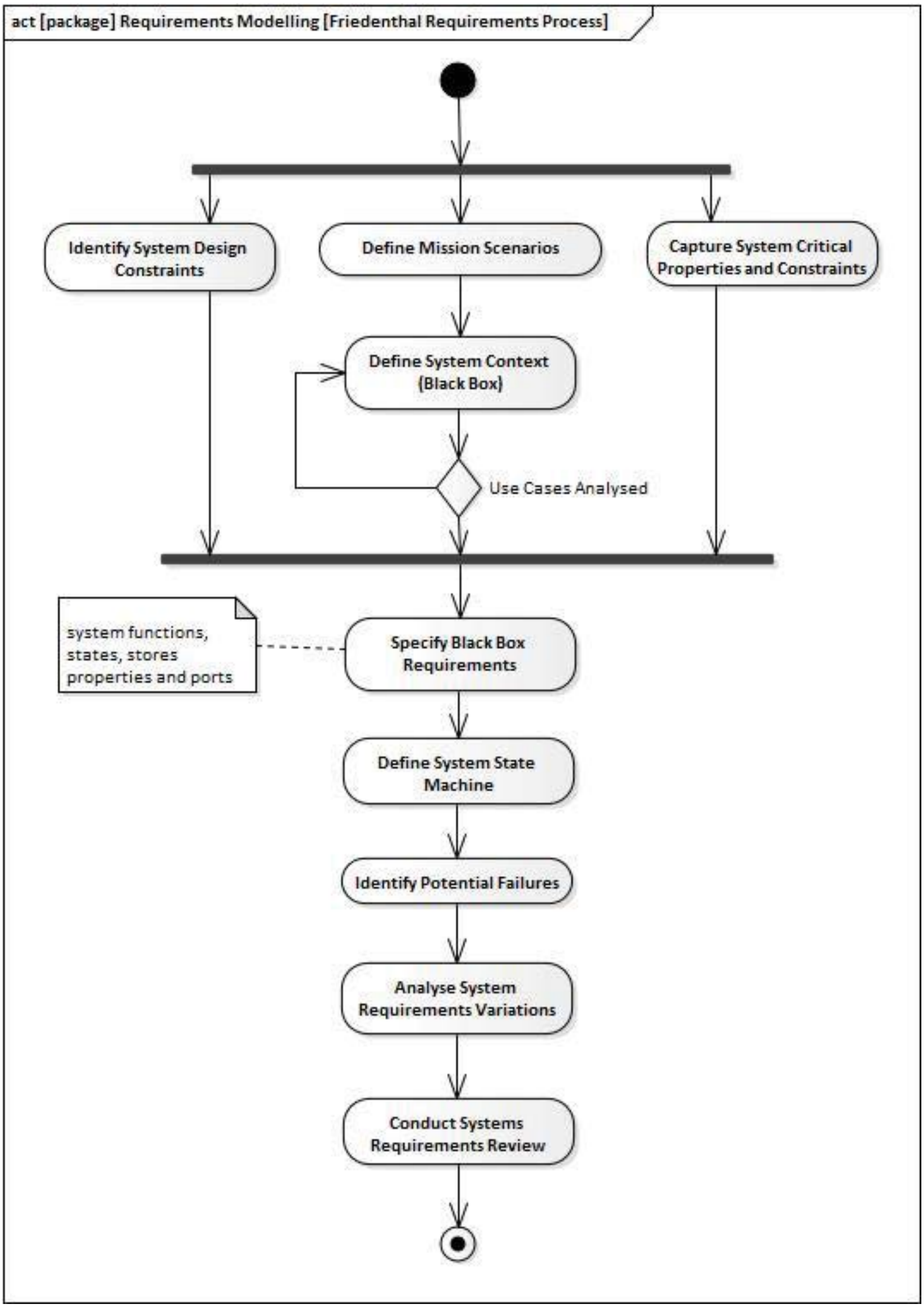

Figure 34 Friedenthal (2015) Requirements Process

The approach recommended by Weilkiens (2007), shown in the SysML Activity Diagram in Figure 35, has its origins in the automotive industry, this can be seen with the development of a project context and stakeholder identification at the very start of the process, this has more in common with a GTS type system. This approach has also been adopted for the approach developed for this thesis. However, the remaining phases of requirements development are treated somewhat in the detail, with particular application to the automotive industry.

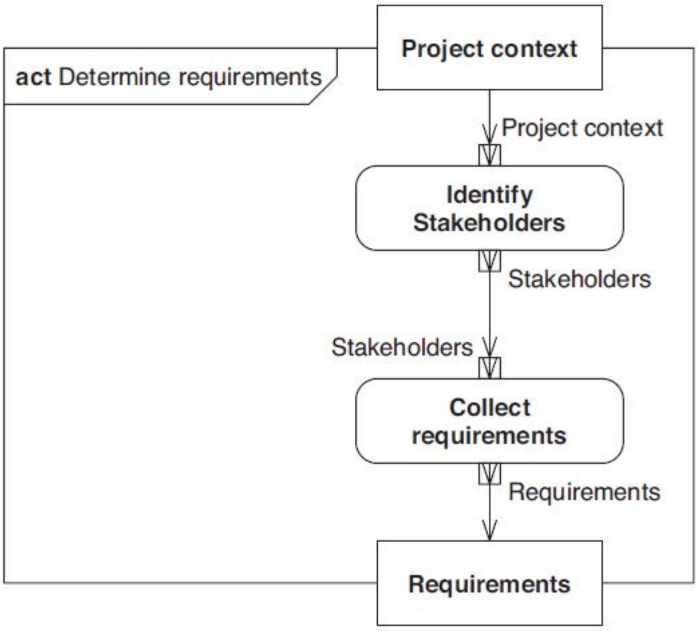

Figure 35 Weilkiens (2007) requirements process

The approach to requirements elicitation and development taken for this thesis is detailed in the SysML Activity Diagram below in Figure 36. Its main features are detailed in the following sections.

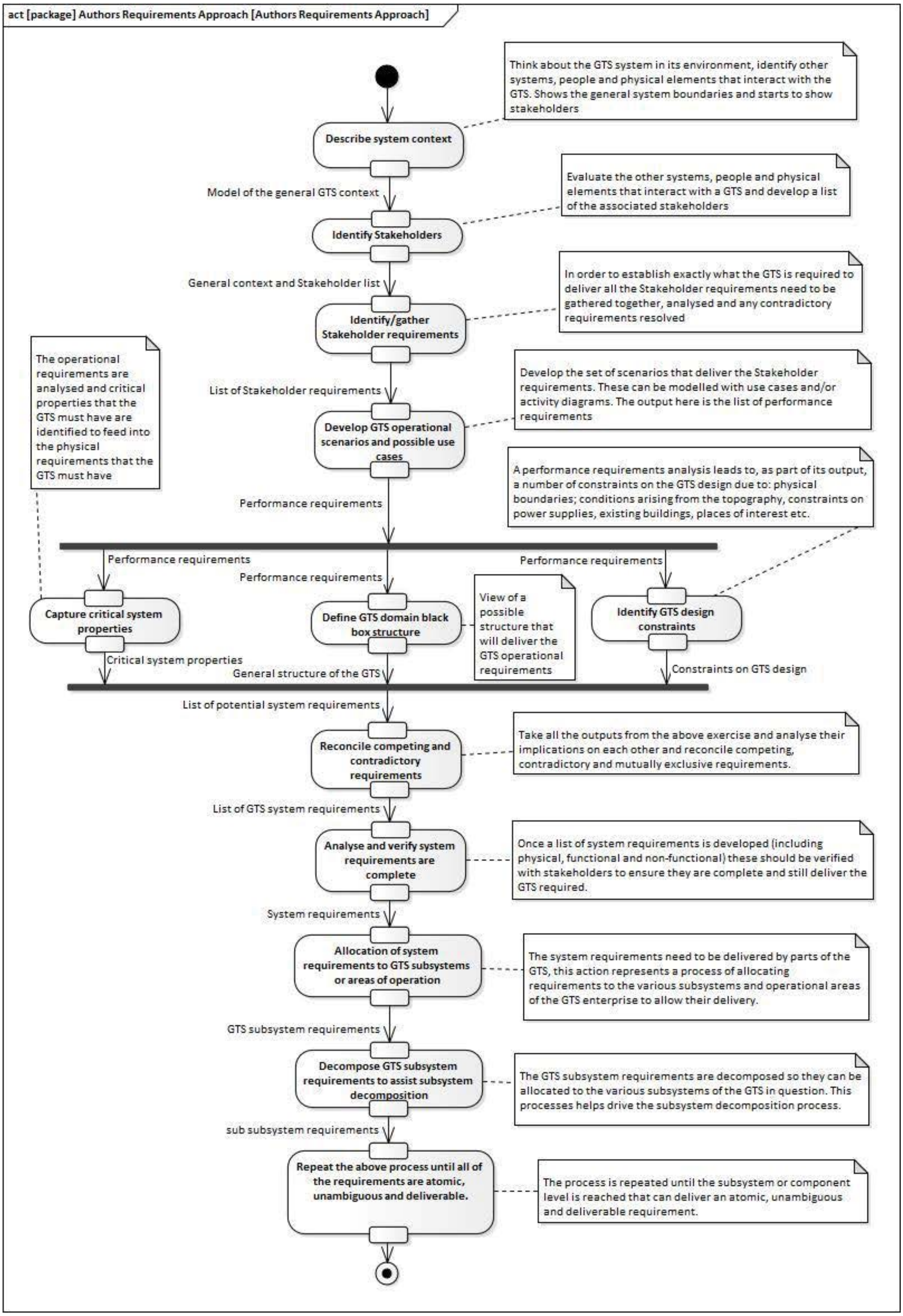

Figure 36 Requirements approach (author)

Stage 5 of the modelling methodology is the iterative part of the above where each requirement is decomposed until it can be answered with a single atomic argument. This is demonstrated in the next chapter through examples of dynamic and structural modelling. Verification of model is an ongoing process as at each iteration the requirements are traced back up to the highest level to ensure the right system is being delivered.

5 Development of Integrated GTS Model

5.1 Introduction

This section contains a demonstration of the process detailed in chapter 4. It is not the intention here to demonstrate a whole model of a GTS only a representative sample of the models that can be produced by the application of the methodology. The generic context model and package diagram are the artefacts that lead the modeller through the processes.

5.2 System Context and Stakeholder Identification and requirements

To begin with the system context is developed using the generic context model as a guide, as described in 4.3, this process also hi-lights a number of Stakeholders. The next stage is to evaluate the systems, organisations, people and physical elements identified during the specific context modelling that make up the GTS context and identify any new Stakeholders.

Once the Stakeholders are known their requirements can be identified, these can be obtained from asking the Stakeholders for their requirements. However, in the GTS world, there are also numerous other sources of Stakeholder requirements that are not necessarily got through conversation, these are things like franchise agreements, which specify exactly what a government agency expects from a GTS operator. Laws, standards and codes of practice are also providers of system requirements, particularly in terms of safety. However, they even effect how sub systems function together through standards like Technical Specifications for Interoperability (TSIs) from the European Union. A representative list of Stakeholders and possible requirements are given below in Table 2 (this is not supposed to be an exhaustive list of either Stakeholders or their requirements it is a demonstration). Also where a requirement, such as that the GTS shall comply with applicable standards and codes of practice, is in the interest of more than one Stakeholder it is only included once rather than for each Stakeholder, for brevity.

Stakeholder	Concerns/requirements
Manufacture & Developer	The GTS shall operate over defined routes to a defined service pattern.
Governing Bodies	The operator shall deliver the service as described in the franchise agreement and/or service level agreements.
	The GTS, its management, its finances and its operation shall comply with current legislation.
	The GTS shall implement and comply with all applicable standards and codes of practice.
	The service shall display reliability and availability as described in the service level or franchise agreement.
	The GTS shall operate within the constraints of its published budget.
	The GTS shall provide a demonstrably safe environment for its customers, employees and the general public.
	The GTS operator - management shall produce regular detailed financial reports as part of their demonstration of compliance with the franchise agreement/service level agreement.
	The GTS shall continue to provide services for a defined period of years (life time).
Business'	In the event that freight transport is part of the agreed service delivery, the GTS shall provide access for freight and storage of freight when not in transit at its freight yards.
	The GTS shall provide a regular service to a published timetable for both passengers and freight.
	The GTS shall have adequate access and egress points (stations) at each stopping point along the guideway for the predicted passenger numbers for a defined period (years).
Railway Undertakings	The Infrastructure shall be capable and have capacity to allow a train operator to provide its contracted service.
	The infrastructure shall have and keep to a defined level of reliability.
	Any changes to the infrastructure should not make rolling stock currently operating on that infrastructure incompatible.
	The operator shall be able to set fares, within current legislation, to enable it to cover its own cost (along with any subsidy) and produce an agreed level of profit.
	The infrastructure shall conform to and maintain its conformance to all applicable standards and codes of practice.
	The Train operator shall have access to the necessary stopping points along the guideway to enable the provision of the contracted services.
Infrastructure managers	All rolling stock using the infrastructure shall be demonstrated as compatible with the infrastructure.
	The rolling stock shall be compliant with and be maintained to stay compliant with all applicable standards and codes of practice.
	The Infrastructure Manager shall have the time and resources to maintain the infrastructure network to all applicable standards and codes of practice.
	The Infrastructure Manager shall negotiate service patterns with potential operators, proposers etc.

Stakeholder	Concerns/requirements
Passengers	Passengers shall be able to travel between any stopping points along the guideway subsystem, if they are in possession of a correct ticket/travel permit, in line with the published timetable.
	Passengers, including those with restricted movement, sight or hearing deficiencies, shall have the ability to access or egress the GTS in order to make a journey.
	Provision shall be in place for a passenger to purchase a journey (single, return, season ticket etc.).
	There shall be a published timetable that shows the departure times of rolling stock from all stopping points along the guideway. It should demonstrate the operators compliance with the service level or franchise agreement.
	The operator shall provide the capacity (with their rolling stock and timetable) to allow the predicted levels of passenger traffic.
National Politicians	The GTS shall perform to its Franchise/Service agreement.
	The GTS shall require no more funding than the level agreed at the time of the Franchise/Service level agreement.
	The GTS shall provide an acceptable level of passenger experience, the GTS management shall demonstrate this through agreed methods, e.g. surveys, secret shopper reports etc.
Local Politicians	The GTS shall satisfy the identified local need.
Pressure Groups	The GTS shall have a minimum impact on the local environment and people.
	The GTS shall demonstrate a defined level of energy efficiency, in the type of energy it uses and the amount of energy it uses.
	The GTS shall provide a defined minimum service outside peak times.
Residents	The GTS shall not adversely affect local residents in terms of noise, pollution and sites of interest/green spaces.
Other GTS'	The GTS shall not interfere with other GTS' in the area from an EMC point of view.
	The GTS shall not adversely affect any shared power sources.
	The GTS shall not adversely affect any shared command and control elements.
	The GTS shall not adversely affect any shared guideway way subsystems.

Table 2 Stakeholders and Stakeholder requirements

5.3 Operations scenarios

Now there is a list of Stakeholder requirements/expectations defined (they may not all be deliverable), the next stage is to analyse how they can be delivered operationally, still keeping the results as solution independent as possible and making sure the analysis is kept within the model. Again it is not the intention here to analyse all of the requirements identified above, just to show a representative example. A number of techniques are applied to understand what is required to meet the Stakeholder's requirements. As this is a generic process there are no real performance or stakeholder requirements in place. Therefore, a generic activity that applies to almost all passenger carrying GTS' is chosen to be representative. It is "Passengers shall be able to travel between any points along the guideway subsystem, if they are in possession of the correct ticket/travel permit, in line with the published timetable". This involves a Passenger wanting to make a Journey, checking that Journey is available and at the times required, purchasing a ticket and then making the Journey. As this requirement is about a goal of the system as seen from an Actor's point of view (a Passenger), the analysis starts with the development of a Use Case. A Use Case is used to describe the black box system functionality of the system in terms of how it is used to achieve the goals of its various users (Freidenthal, et al, 2015, p. 295). The process usually begins with a Use Case description or specification, which provides a narrative of what happens when an Actor (in this case a potential Passenger) invokes the Use Case, Cockburn (2000, pp. 74 - 75) provides a usefully guide to the form this specification should take, although I have taken only a subset of Cockburn's format as it does not all apply, the specification is in Table 3.

Specification heading	Comment	Use Case clause
Use Case name	A verb phrase.	Passenger wants to make Journey.
Use Case scope	What system is being considered as the black box.	GTS and its context as defined in Figure 32.
Primary Actor	A role name for the primary Actor that invokes the Use Case.	Passenger.
Supporting Actors	Actors that provide a service to the system (participate in the Use Case by performing actions).	Agent.
Stakeholders	Someone or something that have an interest in the behaviour of the system.	Train Operator; Governing Body.
Preconditions	The conditions that must be true for this Use Case to begin.	The GTS must be in operation; The Journey required must be available.
Trigger	The event that gets the Use Case started.	A Passenger decides to make a Journey on the GTS.
Main success criteria	The scenario (the sequence of steps) in which nothing goes wrong.	Passenger wants to make a Journey; Passenger ascertains that the Journey is available; The Passenger purchases the Journey; The Passenger makes the Journey; Passenger leaves the GTS.
Extensions	Alternative sequences of steps branching off of the main success criteria.	Journey unavailable or delayed due to fault on the GTS or engineering works; Provide alternative transport (bus) Postpone Journey.

Table 3 Use case specification for Passenger makes a journey

The Use Case model is shown below in Figure 37. It can be seen that there are two external Actors, the Passenger who wishes to make a Journey and an agent who can also arrange the Journey. The Use Case diagram contains three Use Cases (named ellipses) which represent the (arguably) main activities that go into making a Journey. From the diagram it can be seen that both external Actors ascertain if a particular Journey is available and both can also purchase it, the associations are represented by the lines between Actors and Use Cases.

Only the Passenger can make the Journey. There are choices to be made here between keeping the representation generic, how much detail to add and readability. Obviously there could be a number of other Actors involved here from a system point of view, there are the systems that print the tickets, take the money, reserve seats etc., not to mention other human Actors such as booking clerks and of course communications systems such as the internet. However, adding these activities would begin to make this part of the MBSE model less generic and defeat the purpose at this stage. Also all these detailed elements and the interactions between them would make the diagram difficult to read. The Use Case's worth is in isolating and understanding the main activities required to achieve the goal (the Use Cases), the main Actors involved and their relationships to the Use Cases and making these facts available for further analysis through other methods. The detail would be included in the model; it is just not shown in this view of the model. This is a reminder of a key element of MBSE modelling, the diagrams are not the model, they are simply views of the underlying model.

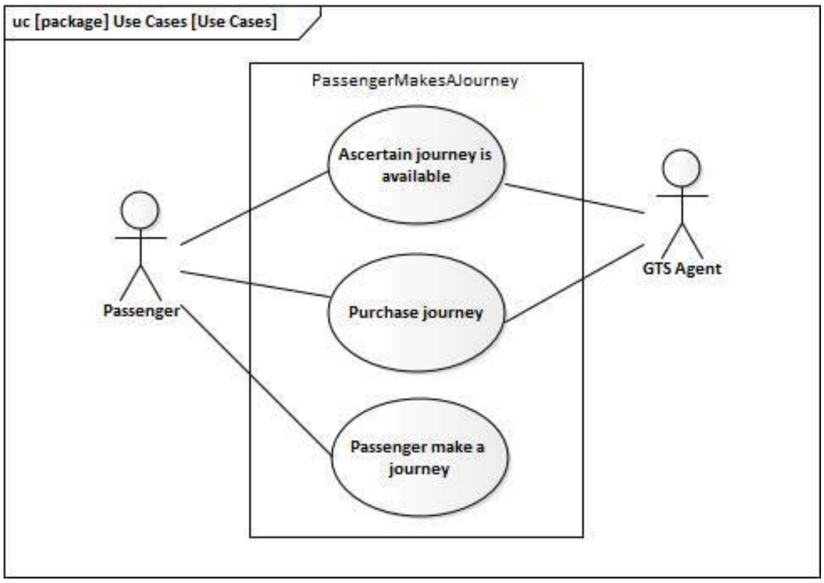

Figure 37 Use case for Passenger makes a journey (author)

The Use Case diagram identifies quite succinctly that there are three main Use Cases in a Passenger making a Journey and that there are two types of Actor that interact with the Use Cases. This is only part of the story and certainly not enough to capture Requirements to a level where they can be allocated to system structures or operations in a way that tracks their delivery. For this the Use Cases can be modelled in detail through SysML behavioural diagrams such as Activity Diagrams, State Machines and Interactions.

An Activity Diagram is an effective tool for the analysis of functional Requirements and is effective in communication with Stakeholders. It describes behaviour that specifies the transformation of inputs to outputs through a controlled sequence of actions (Freidenthal, et al, 2015). For the purposes of this process behavioural diagrams, of which Activity Diagrams are one, take the identified Use Cases and model them as Activities. Actions are the basic building blocks of Activities and describe how the Activities execute. The Activity Diagram below models the generic and high level activity of “a passenger takes a Journey”, expanded from the Use Case Diagram in Figure 37. This author decided to model the behaviour through an Activity Diagram rather than, say a State Machine, because the process from a behavioural stand point and the view of the passenger is a flow of elements of various descriptions. Information such as Journey details, tangible objects such as permits to travel and a Journey itself which can be thought of as a flow between points.

For more information see Appendix B Activity Diagrams, Freidenthal, et al, (2015), pp. 205 - 244 and Delligatti (2014) pp. 89 – 121.

In the Passenger makes a journey Activity Diagram below, the process of a passenger deciding to make a Journey, ascertaining if that Journey is available (offered by the GTS), getting permission to travel (ticket etc.), going to the Access and Egress point, making the

Journey itself and then leaving the GTS is modelled. It is modelled in a system/solution neutral way because this is still a generic guide, so it is not possible to say if tickets can be issued, can they be got from the operator, the internet etc., what sort of stopping point is available etc. Its role is still to drive out requirements. For the purposes of this demonstration a journey will be defined as a set of data that can be passed to various Actions. The data set will contain, information on start, finish, date, time, valid seat, exemptions etc.

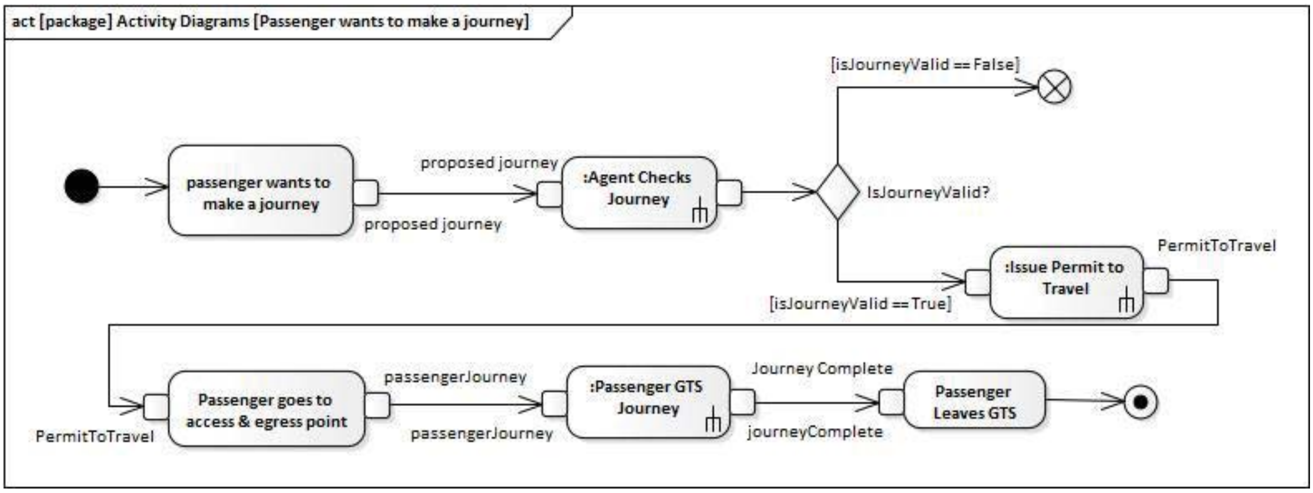

Figure 38 Passenger makes a journey Activity Diagram (author)

The Activity starts at the Initial Node, which invokes the action "passenger wants to take a journey". This is an example of an Action that does not need to be decomposed any further, it's not the intention to model peoples thought processes, it is therefore, an Opaque Action. The potential Passenger needs to know if that Journey is offered by the GTS and that information comes from some agent of the GTS, either human, online or some form of timetable. A proposedJourney of a type Journey Token is modelled leaving the Output Pin on "passenger wants to make a journey" Opaque Activity and arriving on the Input Pin of the Call Behaviour Action of Agent Checks Journey. This is a Call Behaviour Action because clearly this Action will consist of other Actions to implement it (although this is not further decomposed here). The output from "Agent Checks the Journey" is an isJourneyValid Token which is a Boolean. The Token can go either of the ways indicated by the Decision Node depending on the value of the isJourneyValid Token. It is controlled via the guard conditions on the Object Flows [isJourneyValid == False] and [isJourneyValid == True]. If the result is False, the flow ends. If the result is True it is put on the Input Pin of the “Issue Permit to Travel” Call Behaviour Action, again it is expected that further analysis will reveal a number of Ations beneath this Action. The output will be a permitToTravel of type PermitToTravel Token. The Passenger would then take this permit to travel to the Access and Egress point of the GTS (station or ticket machine etc.). This is modelled through the PermitToTravel Token being placed on the Input Pin of the “passenger goes to access & egress” Action, this is an Opaque Action as it requires no further decomposition for this model, although in the future this can be expanded if the intention was to model journeys to and from stopping points. At this point philosophically the PermitToTravel is exchanged for a Journey. This is modelled by a passengerJourney of type Journey Token being placed on the Output Pin of the passenger goes to access & egress point opaque action. It then transfers along the Object Flow to the Input Pin of the “Passenger GTS Journey” Call Behaviour Action, this Call Behaviour Action encapsulates the Passengers Journey along the GTS. It is a Call Behaviour Action because there are a number of Actions that go into making up a Journey on a GTS. Once the Passengers Journey is completed then the Passenger leaves the GTS. This is modelled by a journeyComplete Token of type Journey being placed on the Output Pin of the “Passenger GTS Journey” Call Behaviour Action and transferring along the Object Flow to the Input Pin on the passenger leaves GTS Opaque Action. The control flow to the Activity Final Node completes the Activity.

This relatively simple view of the underlying model starts to drive out operational Requirements and further model elements. It has been identified that there needs to be some element, Journey, which should have some attributes, such as start point, end point, start time, end time as indicated by another model element that has been identified, Timetable. Another entity that will need to be included is the permit to travel. Also the need for an access and egress point has been identified as well as Rolling Stock and if Rolling Stock is required then a Guideway. This confirms what was already being considered from Table 2 Stakeholders and Stakeholder requirements, and what was defined in section 2.3 with regard to the subsystems that are required for a system to be a GTS and only a GTS. Once this top level is correct the next task is to look at the Call Behaviour Actions to look for further structural and functional Requirements. This author has chosen Passenger GTS Journey as the representation of this stage of the process as it is the most generic and diverse behaviour identified so far. The Activity Diagram is shown below in Figure 39.

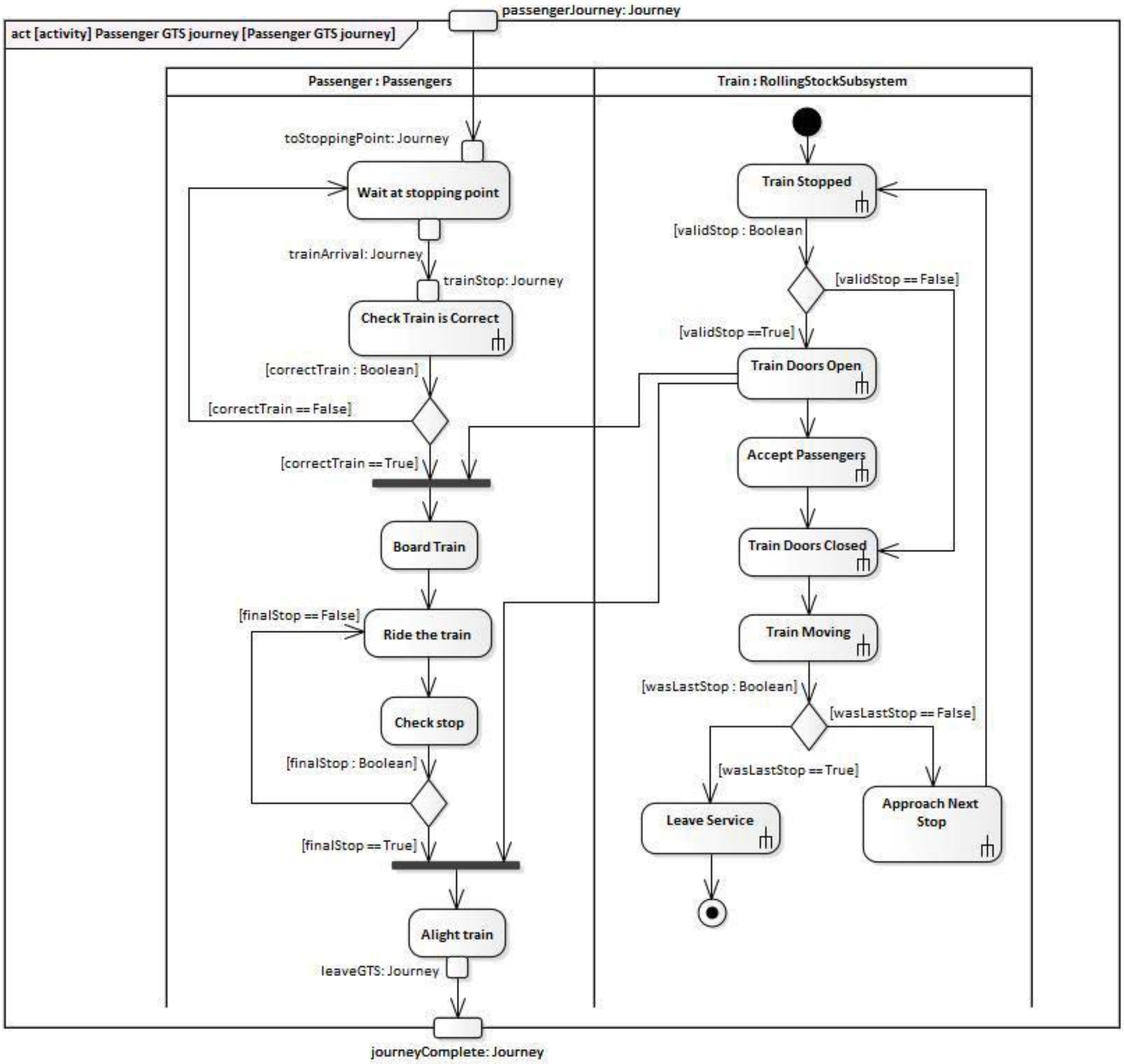

Figure 39 Passenger GTS Journey Activity Diagram (author)

This view of the underlying model is still purely from the passenger point of view representing a particular part of the service that the passenger requires from the GTS. On the boarders of the diagram frame are two rectangles, these are Activity Parameter Nodes and they represent parameters that are passed into and out of this Activity Diagram from the Activity Diagram in Figure 38. They are the Input Pin and Output Pin of the Passenger GTS Journey Call Behaviour Action, the input is passengerJourney of type Journey Tokens and the Output Pin generates journeyComplete of type Journey Tokens to the passenger leaves GTS Opaque

Action of Figure 38. These then are the inputs and outputs to the Activity Diagram in Figure 39. They clearly demonstrate the behavioural nesting and encapsulation that this approach is capable of. This is a vital part of this process as it allows modelling at different levels of abstraction and therefore detail can be added, elements specialised and/or added as they become known or needed throughout the model development. There are two behavioural flows modelling the passenger behaviour and the behaviour of the Rolling Stock, now called a Train. A Train is a specialised type of Rolling Stock, which is actually a collection of Rolling Stock elements, to be modelled later (see Figure 48). The focus of this view of the model is the Passenger, so the only Train behaviour modelled relates to what the passenger needs, so technical elements relating to a Train are not modelled. A further element related to travel has been identified: Route. The Route being something the Train does as opposed to Journey (identified above) which is something the passenger does. These are represented by Blocks which are parts of other structural elements of the model. A Train (not Rolling Stock) will own a Route. The GTS Operations subsystem will own a number of Journeys; a Passenger may join in a Journey. The other major difference between Figure 38 and Figure 39 is that there are two behavioural flows and they are each enclosed in rectangles, these are Activity Partitions, these allocate behaviours to structures and represent which structural elements of the GTS carry out the Actions modelled. The header of the right hand Activity Partition represents the fact that the behaviour within it is delivered by a Train of type RollingStock structure. The header of the left Activity represents the fact the behaviours within it are delivered by Passenger of type Passengers. All of the Train behaviours modelled are Call Behaviour Actions, because they would all be decomposed from a technical point of view in other views of the model. Also the only Flows here are Control Flows for the same reason, the actual passing of information, elements of matter etc. that invokes the various Rolling Stock behaviours would be examined elsewhere.

The Train Stopped Call Behaviour Action is invoked via a Control Token from the Initial Node. The Train Stopped Call Behaviour Action produces a Control Token validStop which is a Boolean that moves via the Control Flow to the Decision Node, if it evaluates to True then the Control Token goes to the Train Doors Open Call Behaviour Action, if it evaluates to False then the Control Token goes to the Train Doors Closed Call Behaviour Action. This is to model a train being stopped at a Stopping Point knowing if it is a valid stop on its Route or not (this can be "known" because the operator knows it, or because of some technical means such as ATO). If it is not a valid stop the doors should stay closed (or Passengers discouraged from boarding in some manner), if it is a valid stop then they can be opened. If the Train doors are open the Train can accept Passengers, at some point after that the Train will cease to accept Passengers and the doors can be closed in preparation for moving away from the Stopping Point. This is modelled via a Control Flow between the Train Doors Open and Accept Passengers Call Behaviours Actions. The control flow then moves to the Train Door Closed Call Behaviour Action. These three Call Behaviour Actions are named carefully because this is a generic model, the Train Doors Open Call Behaviour Action is simply the fact that the doors are open, it doesn't refer to how they open. The same is true for the Train Doors Closed Call Behaviour Action. This functionality can be examined and modelled at a later date when the detail of the actual application is known and the Activities called by the Call Behaviour Actions model this detail, this is called modelling for abstraction in MBSE. The Accept Passengers Call Behaviour Action does the same, it is not possible to know at this stage how the despatch of Trains will be managed. Once the Train doors are closed the Train can continue along its Route. This is modelled via the Control Flow from the Train Doors Closed

Call Behaviour Action to the Train Moving Call Behaviour Action. There is now a Decision Node representing two possibilities one to approach the next stop, the other to leave service. The Control Flow from the Train Moving Call Action Behaviour is a wasLastStop Boolean Control Token. This is evaluated, if true the Leave Service Call Behaviour Action is invoked, if it is false then the Approach Next Stop Call Behaviour Action is invoked, on completion of this Action the control loops back to Train Stopped.

The behaviours described above are all a passenger needs for the service required, it has a train stopped at a stopping point, the doors are open if it is available for service, closed if not. The train moves off from the stopping point to either the next stopping point or leaves service. All this behaviour is provided by a Train.

Passenger behaviour also needs to be modelled in this context. It begins with the Object Token toStoppingPoint of type Journey being passed to the Input Pin on the wait at stopping point Opaque Action, which simply models a passenger waiting for a Train. An Object Token is passed from the trainArrival Output Pin of the wait at stopping point Opaque Action to the Input Pin of the Check Train is Correct Call Behaviour Action. This models the behaviour of a Passenger when a Train is at the Stopping Point in terms of determining if it is the Train required to make the Journey. There is a Control Flow from the Check Train is Correct Call Behaviour Action which is a Boolean Control Token correctTrain. This is evaluated at the Decision Node, if it evaluates to false then control loops back to the wait at stopping point Action. If it evaluates to True it puts a Control Token onto the Join Node. The Join Node has another Control Flow from the Train Doors Open Call Behaviour Action in the Train Activity Partition. Once there are Tokens from both on the Join Node the board train Opaque Action is invoked, until both Tokens appear on the Join Node the Board Train Action cannot start. This series of actions models a Passenger waiting for a Train, checking if a Train that is in the stopping point, is the required service, then boarding the Train once the doors can be opened provided it is the right Train. If it is not the right Train the passenger carries on waiting. The Passenger riding the Train is modelled by the ride the train and the check stop Opaque Actions. The Control Flow from the check stop Opaque Action is a Boolean finalStop, which is evaluated. If it evaluates to false the control goes back to ride the train, if it evaluates to True then a Control Token is placed on the Join Node, the Join Node also has a Control Flow from the Train Door Open Call Behaviour Action. Once Tokens from both are present the Alight the Train Opaque Action is invoked and a leaveGTS Object Token is passed out of Passenger GTS Journey Activity Diagram to the Passenger Makes a Journey Activity Diagram. The Passenger GTS Journey activity is complete.

The process has identified a number of functional Requirements that need to be included in the Requirements model. From the high level Passenger wants to make a journey Activity Diagram there are Requirements for some form of information exchange between a Passenger and the GTS system. This is in the form of some sort of agent that can establish if a Journey is available. There is a Requirement to purchase a Journey and have some form of proof of purchase (permit to travel). There are also the obvious requirements for a Passenger to be able to access and egress the GTS. From the Passenger GTS Journey Activity Diagram a Requirement for some specialisation of the generic Rolling Stock subsystem into a Train that is a collection of Rolling Stock was identified. Also a train owns an entity called a Route. There is a requirement for the Passenger to be able to ascertain if a Train stopped at a Stopping Point is the one required or not. There is also a requirement for the Passenger to able to check the progress of the Journey in order to know when to alight the train.

Each of the Call Behaviour Actions would be analysed as described above until all the Actions are decomposed as much as possible, thus driving out all the available requirements for the

Use Cases involved in a Passenger making a journey. This process from Use Case to behaviour model should be applied to all the identified service type functional Requirements. In this way a set of functions and behaviours is not only identified but also they are traceable to the actual needs of the Stakeholders of the GTS.

5.4 Performance requirements

The list of Stakeholder requirements and the operational scenario modelling is now synthesised into a set of generic performance requirements, as shown below in Figure 40, through a process (usually) of Requirements analysis and negotiation with Stakeholders where Requirements are conflicting or competing. Some Requirements have been subsumed into others, for example, Passengers, included those with restricted movement, sight or hearing deficiencies, shall have the ability to access or egress the GTS in order to make a Journey, is subsumed into the Requirements to comply with legislation and standards as compliance covers access for persons of restricted movement and there is also the disability national legislation.

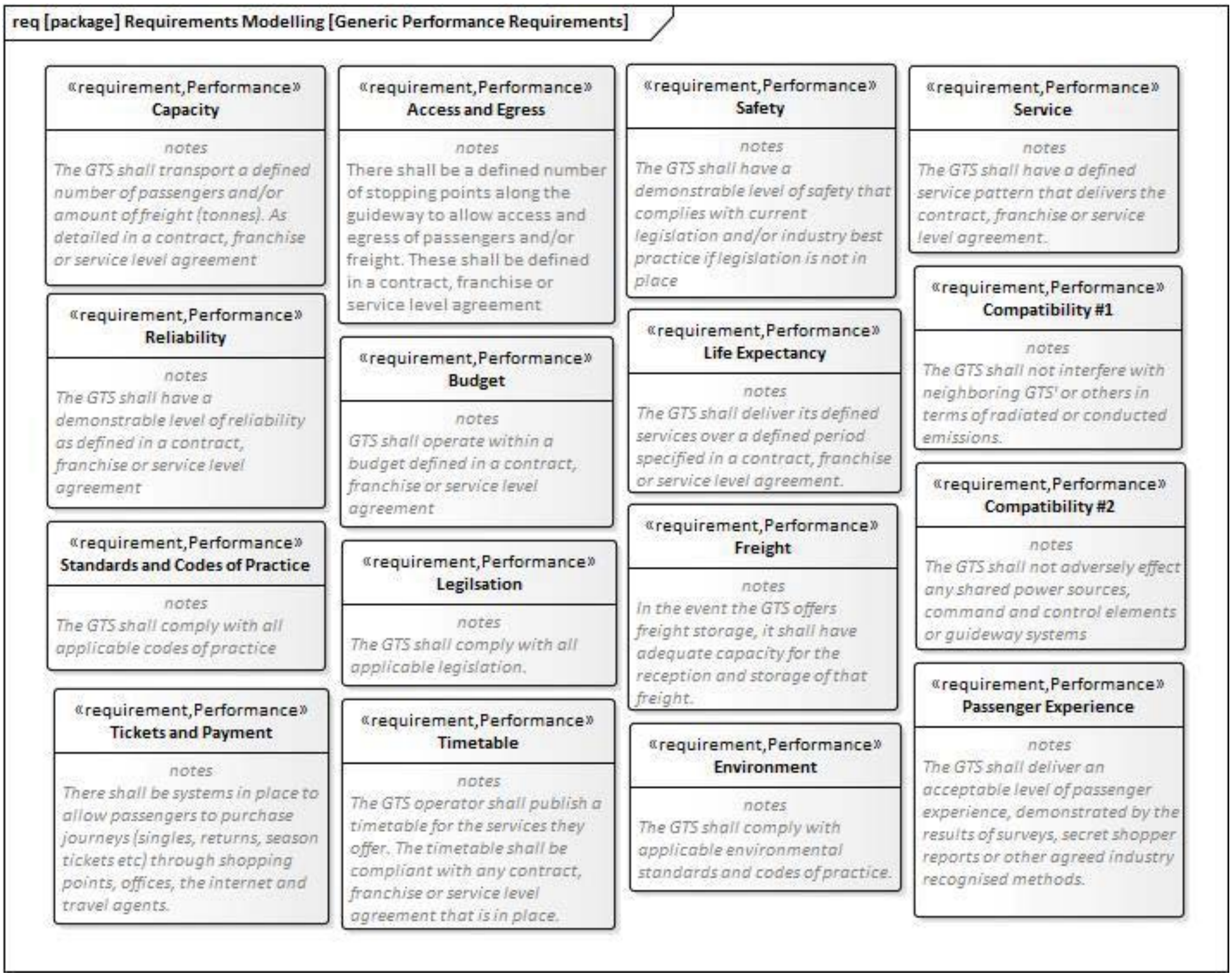

Figure 40 Generic GTS performance requirements (author)

5.5 Critical System Properties

The process of Requirements elicitation as shown in Figure 36, identifies the need to identify critical system properties. These will almost certainly come from documents like Business Requirements Specifications, System Requirements Specifications etc. Even without such documents being in place, it is possible to develop these at a generic level from available literature see Association of Train Operating Companies (2016), Department for Transport (2015), and Department for Transport (2016). These system critical properties are the highest level outputs based on how GTS's are judged by stakeholders and interested parties, e.g. some Key Performance Indicators (KPIs) which clearly express the GTS' structure and performance in the context of the various Stakeholders. These then become critical system properties, who's actual values will be derived from the lower levels of decomposition within the model when it is applied to a real GTS. It should be noted that any set of KPIs at this level are underpinned and supported by a whole hierarchy of lower level KPIs going down through the subsystems right down to structural component level or lowest level of operational rule or management process. It is this relationship between outputs at the various levels of abstraction that the project intends to take advantage of to develop a fully integrated modelling approach that is multi stakeholder friendly.

One approach to railway KPIs was proposed by InteGRail, (2014) they concluded the following for what they say are the four main subsystems of a railway system, Rolling Stock, Operations, Infrastructure and Train Management:

- Rolling Stock:
  - Availability;
  - Reliability;
  - Life Cycle Cost;
- Operations:
  - Customer satisfaction;
  - Number of trains;
  - Punctuality;
  - Transported payloads;
  - Cost of operations;
- Infrastructure:
  - Availability;
  - Reliability;

  - Capability;
  - Life Cycle Cost;
- Traffic Management:
  - Quality of train Plan;
  - Regulation – rescheduling;
  - Information.

The KPIs identified above are at a very high level, and while they are useful in terms of being generic, they do not have the breadth required for this approach. There is nothing explicitly for communications, which has become such an important subsystem over the last few years. Infrastructure has nothing explicit about stations and other access points. Also nothing regarding traction power supplies (although all these could come under infrastructure at its most general). Traffic management doesn't really deal with signalling in terms of its suggested KPIs e.g. reliability statistics. Logistics elements are not included either and they can have an impact on performance.

The National Audit Office (2015), uses:

- Track miles;
- Passenger number forecasts;
- Inward investment;
- Spending; and
- Staff employed.

As one might expect these are very financially biased, however they are still data that are required by an important GTS stakeholder, so should be included, in some form, as properties at the highest level of the GTS model.

Others such as Anderson et al. (2003) have sort to provide much more subsystem (in this case infrastructural view) and operationally biased KPIs:

- Asset utilisation: train km per track km;
- Safety: Accidental equivalent fatalities per train km;
- Financial effectiveness: Harmonised Life Cycle Costs per gross hauled tonne km;
- Efficiency: Planned track possession km hours per track possession km hour;
- Service Quality: train delays due to infrastructure (or any other subsystem for that matter);
- Innovation and growth: Average relative age of assets; and
- Accessibility: Passenger Train km per route km.

Anderson et al. (2003) has gone so far as to suggest how their KPIs should be expressed in terms of quantities and units. These can be viewed as properties that have a type. The numbers that will populate these critical system properties ultimately come from the subsystems at lower levels and are then combined in various fashions to arrive at the top level answer. The essence of the MBSE methodology is keeping these properties all within the same model so they can easily be found and combined as needed. Another advantage is many other questions from other sub-stakeholders can and are answered by the data that comes from the lower levels. Also any changes can be reflected right across the model with less risk of inconsistency.

From the KPIs suggested above a number of generalisations can be made regarding the critical system properties that the generic GTS model needs to represent. Firstly, data about the GTS's physical structure is required, the number and types of rolling stock, track kilometres, number and type of access points (stations, freight yards), number and type of signalling and control assets (including control rooms and other real estate) and topographical information. Also for nearly all the subsystems some form of performance data is required, usually in the form of reliability, availability, maintainability, Safety, punctuality and assets usage. There is also a requirement for financial data such as, Capital Expenditure (CAPEX), Operational Expenditure (OPEX), current, past and future revenue. These are captured in the top level Block as shown below in Figure 41 in chapter 5.7.2.

As mentioned in chapter 2.3 a GTS is composed of a number of main subsystems, three of which are required to make the system a GTS and only a GTS. There are another seven subsystems that can optionally be added to the system. These have no bearing on whether the system is a GTS or not, however all or some of them are needed to produce a realistic GTS representation. As this is a generic model with the ability to act as the basis for modelling any GTS, then all the possible subsystems are included. This first hierarchy of the system is shown in Figure 41.

5.6 GTS Design Constraints

The same can be said for GTS design constraints as has been said for critical system properties, these would usually be defined by where the GTS is to be built, if it is a new GTS, what its duty is likely to be, or by the change taking place. As there is no system, these cannot realistically be identified here. Other elements that would impose design constraints on a GTS are the systems geographical boundaries, the availability of particular power supplies leading to constraints on the use of overhead line equipment for example, or restricting the use of diesel powered vehicles, the height of buildings or the use of tunnels instead of over ground, as is the case of HS2 in the UK around London.

5.7 GTS Structural Modelling

5.7.1 Introduction

The system critical properties identified in chapter 5.5 are captured as structure and behaviour. A Block Definition Diagram (BDD) is used to define Blocks and the relationships between them, such as hierarchical relationships through system decomposition.

One of the key principles of MBSE, that of having a whole system characterised in a single logical and coherent model is demonstrated in this section. Using the different views of the underlying model provided in the SysML language, it is possible to clearly see the model taking on the characteristics defined in the chapters above.

5.7.2 Top Level Structure

The view of the model shown below in Figure 41 displays a number of attributes. It shows at the highest level there is GTSOperationalDomain block. Rather than showing all of the Blocks that it is composed of as Blocks, as it is in the context diagram in Figure 32, they are shown in a compartment of the GTSOperationalDomain block. This allows the focus of this view to then be the GTSEnterprise Block, where the majority of the Requirements identified in the preceding chapters are captured at the highest level. In terms of critical system properties, some of these can be seen in the values compartment of the GTSEnterprise Block. The representation of the remaining critical systems properties related to structural requirements can be seen in four blocks with composition relationships with the GTSEnterprise Block. The full name of each Block is also displayed on the diagram so navigation to that block in the model is made easy.

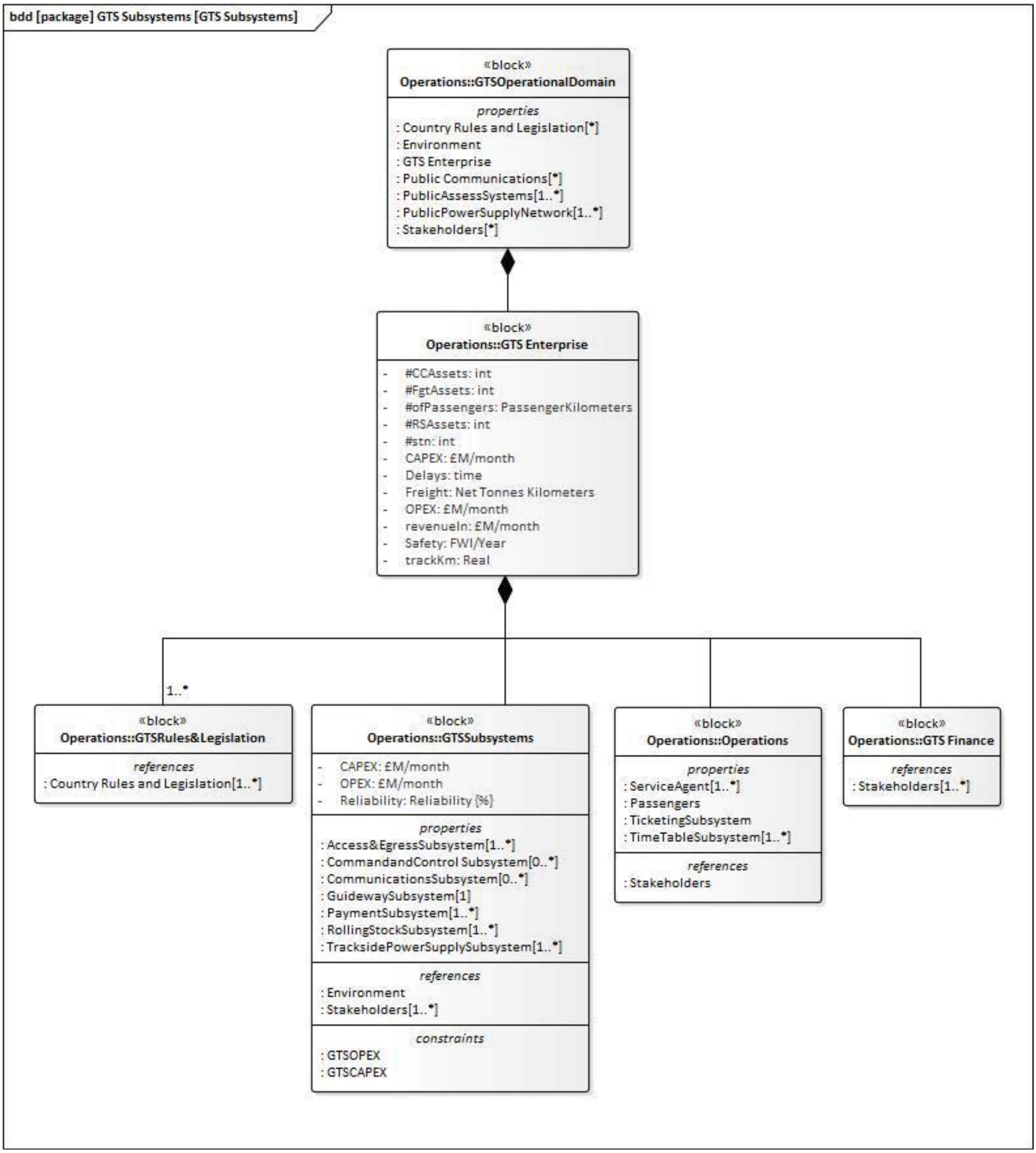

Figure 41 Top Level GTS Model Structural View (author)

The properties captured in value compartment of the GTS Enterprise block are:

- #CCAssets : Integer, will hold the number of command and control subsystem assets that the GTS owns;

- #FgtAssets : Integer, will hold the number of freight assets, in terms of yards etc. that the GTS owns;
- #ofPassengers : PassengerKilometers, will hold the passenger kilometre measure of passenger usage identified in 5.5. PassengerKilometers is a user defined type (see Friedenthal et al (2015) pp 136);
- #RSAssets : Integer, will hold the number of Rolling Stock asset associated with the GTS;
- CAPEX : £M/Month, will hold the figure for capital expenditure associated with the GTS. £M/Month represents millions of pounds per month and is another user defined type;
- Delays : time, will capture recorded delay time;
- Freight : Net Tonne Kilometres, another user defined type designed to capture the measure of freight moved identified in 5.5;
- OPEX : £M/Month, will hold the figure for operational expenditure associated with the GTS. £M/Month represents millions of pounds per month and is a reuse of the user defined type used for CAPEX above;
- RevenueIn : £M/Month, will hold the figure for the revenue generated by the GTS and is another demonstration of the reuse of a centrally stored type;
- #Stn : Integer, will hold the number of Station assets associated with the GTS;
- Safety : FWI/Year, will hold the number of Fatality Weighted Incidents per year, this is the preferred RSSB way of measuring safety and is another use defined type; and
- Trackkm : Real, is a real number that will represent the total amount of track in kilometres of the GTS.

Note, in a practical application these value properties can be changed for some that are more relevant, they are not given as an exhaustive list, merely a guide. The structures above and below the GTSEnterprise Block are also shown in this view, in two ways, designed to maximise readability and the information provided. The composite relationships between the GTSEnterprise Block and the GTSRules&legislation Block, GTS Subsystems Block, the Operations Block and the GTS finance Block are shown as in the context diagram in Figure 32, also in the parts compartments of the GTSOperational Domain, Operations and GTS Subsystem Blocks. The GTSOperationalDomain Block which owns the GTSEnterprise Block shows its parts compartment and this shows all the other systems, including their multiplicities (and if they are required subsystems or not), that compose the GTSOperationDomain Block. This allows the context to be kept in terms of GTSOperationalDomain without cluttering the view of the model. The GTS Subsystems Block also displays a parts compartment this also shows the various subsystems that make up the GTS, along with their multiplicities. As mentioned in section 5.5 all the possible subsystems are shown as this is a generic model. The Blocks also show a compartment called references these show the relationships between entities modelled for the passenger makes a journey Use Case in Figure 42. Note also from the same modelling, the entities defined for the Operations Block to deliver the service are shown in its parts compartment.

5.7.3 Second level model structure

When the behaviour modelling was demonstrated in 5.3 a number of structures were identified that were required to deliver the behaviour described. This is captured on the expansion of the Operations Block shown in Figure 42. It demonstrates that Blocks are used for more than physical things. Looking from the Operations Block, it is composed of three sub Blocks. A timetable Block which is a representation of the main services the GTS will provide. It has two sub Blocks, Route and Journey. Once it is decomposed further, the Route Block would capture the defined Routes that the GTS can take, in terms of stops, distance and timings. The multiplicities demonstrate that one TimeTableSystem can have 1 or more Routes. The other sub Block, Journey, has a composite relationship with the TimeTableSystem Block showing again one TimeTableSystem can have 1 or more Journey's. There is also a dependency between Route and Journey, this captures the fact that a journey is dependent on the Route available and that a Journey is bound by and in some cases is a subset or a multiple of Routes. There is a bi-directional relationship between the TicketingSubsytem which captures that tickets need to represent Journey's; the relationship defines one ticket for one Journey. The TicketingSubsystem also has a relationship with a Block called PermitToTravel which represents the fact the TicketingSubsystem Block will be responsible for producing the Permit to travel (Ticket). The final Block is one that represents Passengers. It is shown in a relationship with the TicketingSubsystem which shows the Passenger interacts with the TicketingSubsystem Block. There is also relationship between the PermitToTravel Block and Passengers which captures a PermitTo Travel being issued to a Passenger.

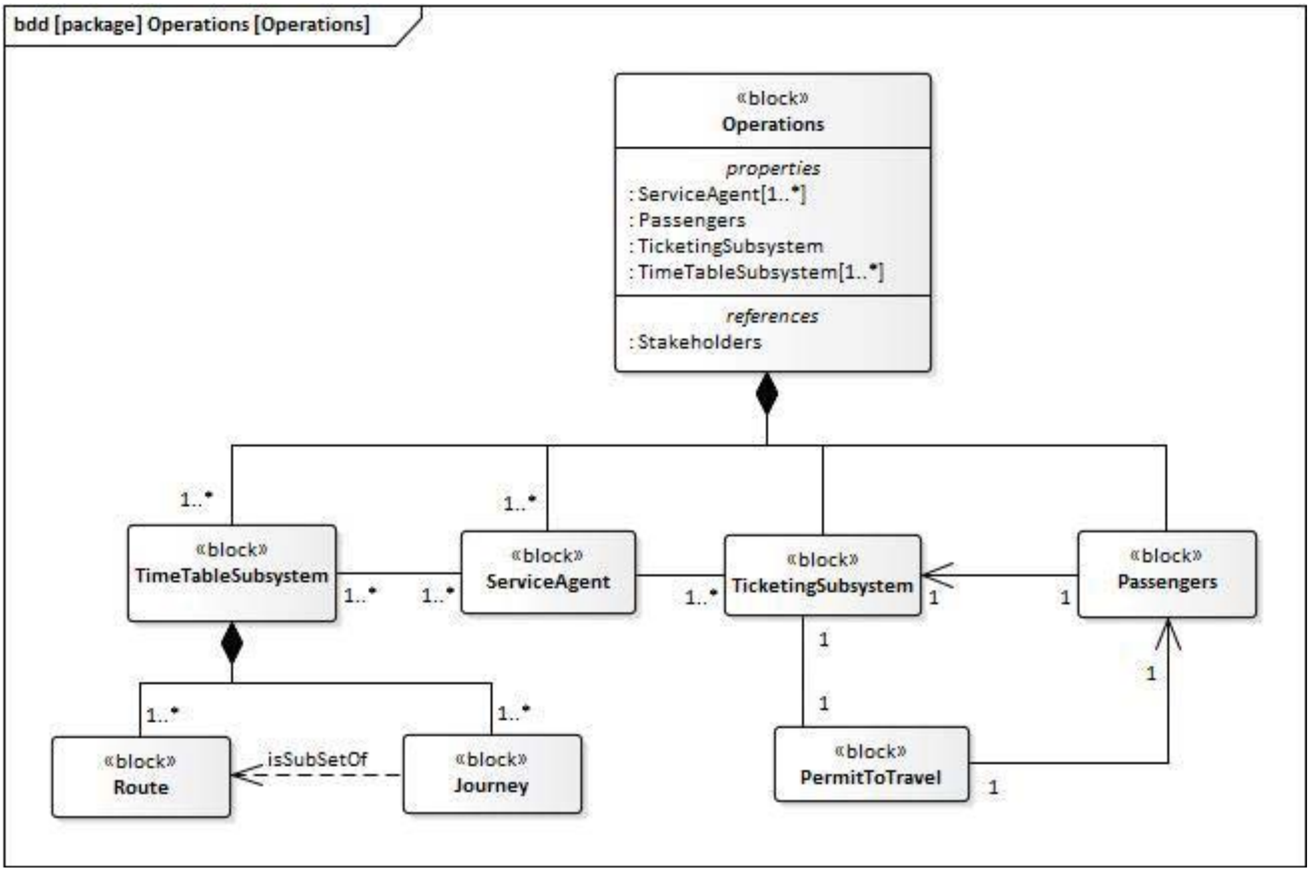

Figure 42 The delivery structure for the passenger wants to make journey Use Case (author)

5.7.4 Definition of subsystem contributions to critical system properties

The GTSSubsystem Block shown in Figure 41 is expanded next to show that some of the Critical System Properties identified in section 5.5 are derived from properties that are attributes of other Blocks. These are numerical values which produce the total GTS CAPEX and OPEX numbers. In SysML variables and their mathematical relationships are defined using Constraint Blocks and Parametric Diagrams. Constraints can be thought of as reusable specifications, that are related to Blocks, that describe relationships between variables (mathematical models), two for CAPEX and OPEX are shown in Figure 43.

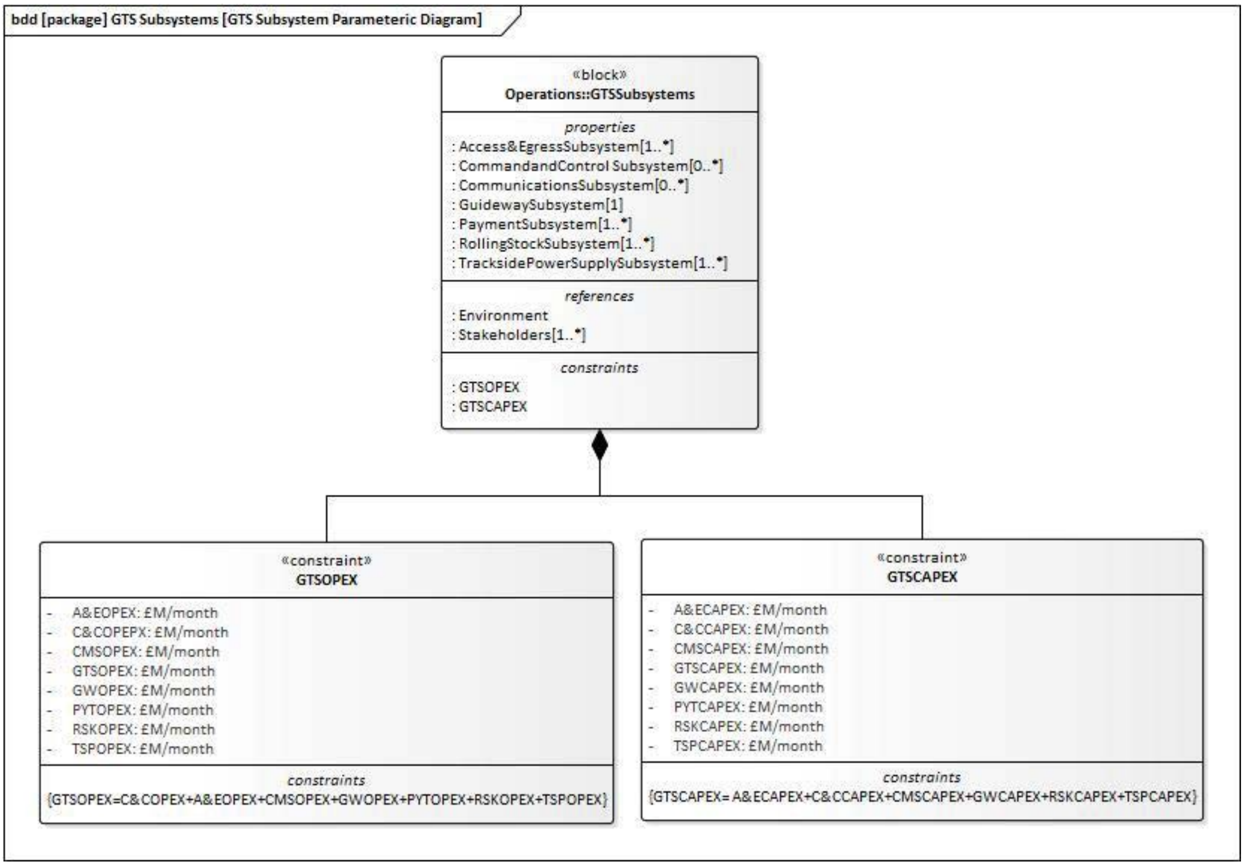

Figure 43 Constraint Blocks for CAPEX and OPEX (author)

Constraints are an extension of a Block as shown by the keyword <<constraint>> in front of the Block name. The constraint parameters compartment below the name compartment shows a list of all the parameters that are the subject of the constraint and their types. The constraint itself is in the constraints compartment which in both cases is a simple addition of the CAPEX and OPEX variables from all of the Blocks that make up the GTS Subsystem Block between braces, for example, for CAPEX:

{GTSCAPEX=RSKCAPEX+C&CCAPEX+A&ECAPEX+CMACAPEX+GWCAPEX+PYTCAPEX+TSPCAPEX}.

A Parametric Diagram is then used to show the relationship from the actual blocks to the constraint, an example of the CAPEX Parametric diagram in Figure 44.

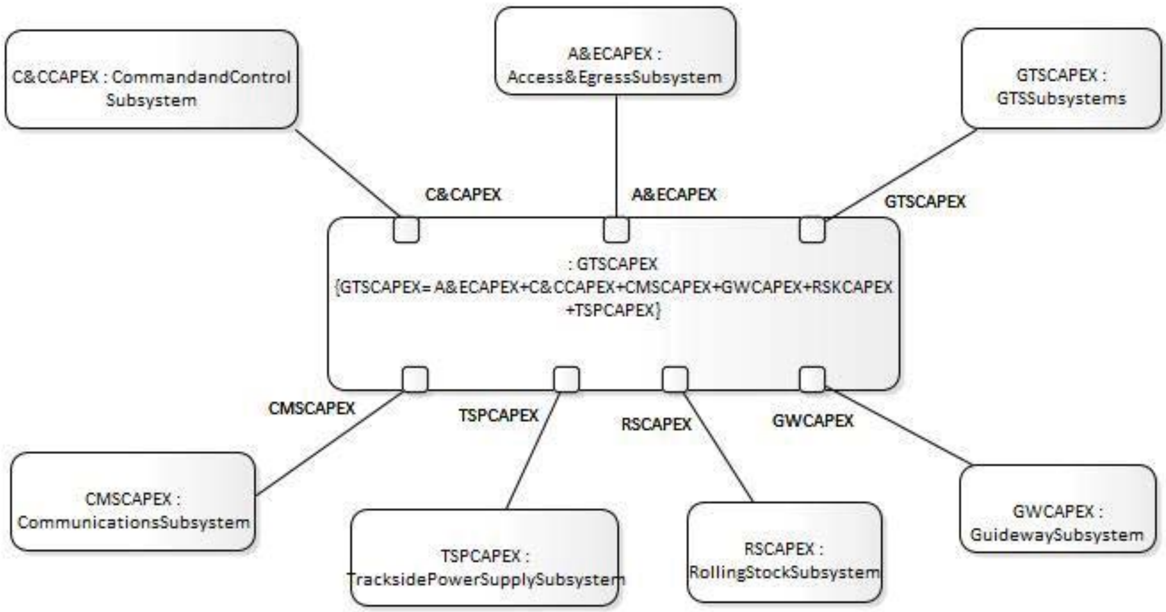

Figure 44 Parametric Diagram for GTS CAPEX (author)

The GTSCAPEX constraint is in the centre of the diagram, the small rectangles represent the Constraint Parameters defined in the Constraint Blocks shown in Figure 43. These are each connected to a Block Property of the associated Blocks, shown around the centre. As stated preiously this is just a specification of the mathematics that needs to take place, what the parameters are, where they come from and how they are related. A parametric diagram does not do the actual maths, however SysML can be linked to other tools that can, such as MatLab SimuLink, that can do this and far more complex mathematics, the results of which can then be placed in the model, either manually or automatically if the tool in question has that facility (this functionality is tool dependent).

5.7.5 Internal subsystem connections, flows and interfaces

There are many more interactions between the GTS Subsystems and these can all be captured in the model to increase understanding of how the GTS subsystems interface and work together. The diagram below in Figure 45 is an Internal Block Diagram (IBD), it is used to model the connections, flows and interactions between the parts of a system. The IBD in

Figure 45 is scoped to the GTSSubsystems Block in Figure 41, it shows the parts of the GTSSubsystems Block, but not the GTSSubsystems Block itself. The diagram does not contain all possible connections, otherwise it would be unreadable. This author has chosen a few of the main connections/interactions to demonstrate how the connections are modelled. Looking at the RollingStock, TracksidePowerSupplySubsystem and GuidewaySubsystem the flow of electricity is modelled from the track side power supply to the rolling stock and back to the track side power supply. A SysML element called a Flow Port is used to achieve this. As this model is generic it is not possible to say if the power supply is overhead, third rail or what voltage or frequency. However, it is still possible to model the flow of electricity in general. In order to model electricity, it has to be defined, this is done using a Block with Block Properties to capture the required attributes, in this case current (measured in Amperes), voltage (measured in Volts), frequency (measured in Hertz) and power (measured in Watts) have been defined along with their respective units, see Figure 46. This definition is placed in a definitions package in the Model Library Package so it can be shared. Flow Ports are then placed on the boarders of the TracksidePowerSupplySubsystem, RollingStock and GuidewaySubsystem Blocks, they are named and all typed to the definition of electricity described above. There is an input and output on each of the Blocks. These are then connected using a standard SysML connector and a itemFlow is added to each connection that is also typed to the definition of electricity. This arrangement states that something called electricity, that is defined by the Electricity Block flows from a Flow Port on the TracksidePowerSupplySubsystem Block along a connector with an itemFlow also typed to the Electricity Block to a Flow Port on the RollingStock Block. It then flows out of the Flow Port on the RollingStock Block along another connector typed to the Electricity Block to the Flow Port on the GuidewaySubsystem Block. It then flows out of the GuidewaySubsystem Block through a Flow Port and along a connector typed to the Electricity Block back into the Flow Port of the TrackSidePowerSubsystem Block. This view of the underlying model is shown below in Figure 45.

So without becoming solution dependent, it has been possible to model that electricity will flow from the TracksidePowerSupplySubsystem Block into the RollingStock Block and returns via the GuidewaySubsystem Block to the TracksidePowerSupply Block. To demonstrate the flexibility of the use of Flow Ports I have also modelled the fact Passengers flow between the Access&EgressSubsystem Block and the RollingStock Block in both directions. The process is the same as described above accept the Flow Ports and itemFlow are typed to the Passengers Block in the Stakeholders Package.

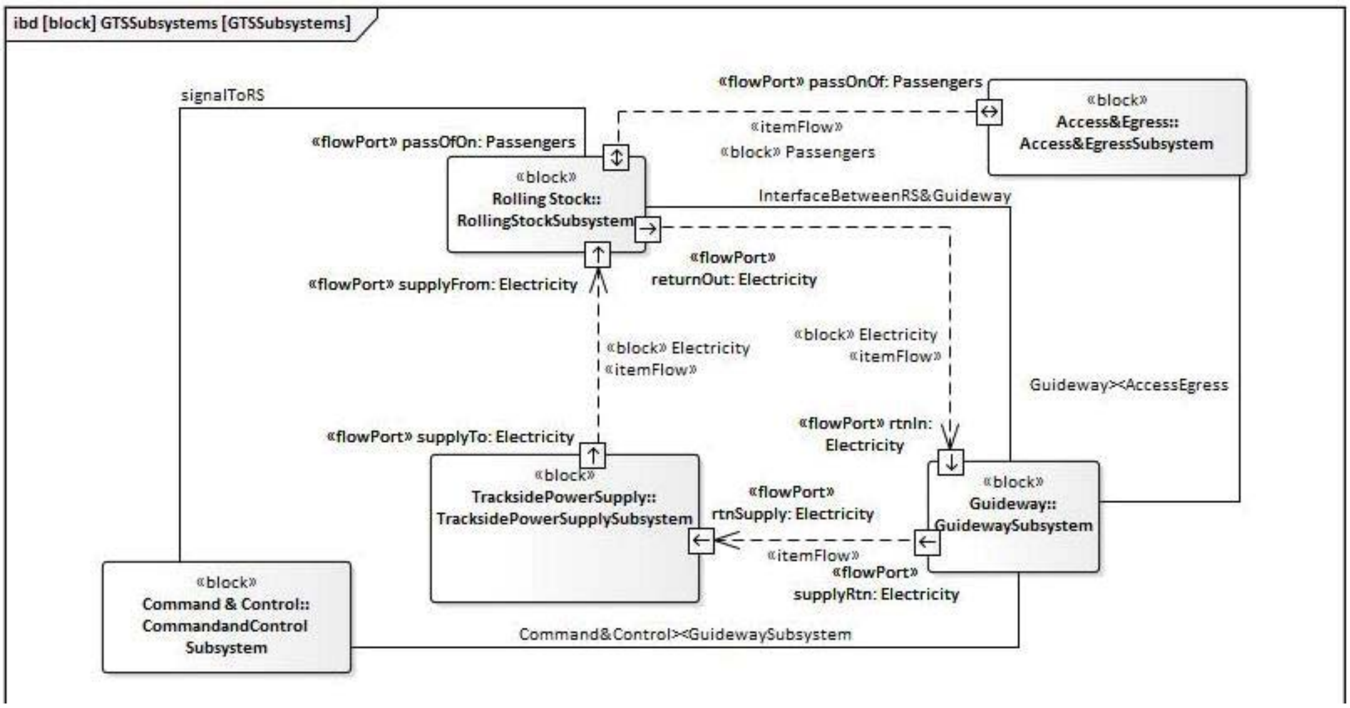

Figure 45 Flow of Electricity through the GTS subsystems (author)

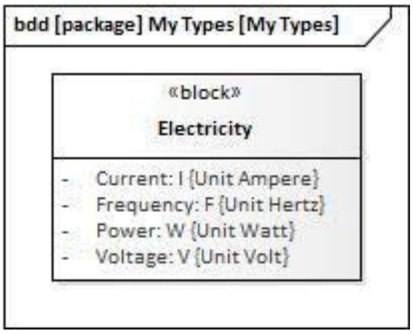

Figure 46 Definition Block for Electricity (author)

There are three other connections modelled in the diagram in Figure 45. The first is between the Guidewaysubsystem Block and the Access&EgressSystem Block, this is to capture the fact there is a relationship between where the guideway system is placed and the edge of the access and egress point, to enable Rolling Stock to pass and Passengers to get on and off. This is not defined in any detail as yet because the Requirements are application specific, so at this stage it is there as a place saver to remind a modeller to characterise the platform train interface when data is available. This is another example of being able to model to levels of abstraction when data is not available. There are also connections modelled from the CommandControlSubsystem Block to the RollingStock and GuidewaySubsystem Blocks. This is to capture the fact that if there is a command and control subsystem, there will be some interaction between it and the Guideway in terms of Route setting, protection and control. Also there is a connection to Rolling Stock either through the operator (if there is one) seeing signals at track side or through some sort of cab signalling system such as ERTMS, CBTC etc. Once detail is available these can then be modelled as interfaces, flows etc., this is another advantage of being able to model to abstraction. There is a further relationship between the Rolling Stock and the GuidewaySubsystem, again the detail of this very solution dependent. How heavy the Rolling Stock is, the profile of the wheels, the profile of the track, the maximum speed and environmental conditions are all part of this connection, which cannot be further defined at this stage but must be once detail is available.

5.7.6 Structural Decomposition

The Block Definition Diagram (BDD) below in Figure 47 shows the RollingStock Block with composite relationships with six subsystems. These are the subsystems that must be present for a system to be Rolling Stock and only Rolling Stock, therefore they are the minimum required for a generic Rolling Stock representation. This representation would need to be specialised in order to represent any actual Rolling Stock (see Figure 48). Note the RollingStockSubsystem Block is the same RollingStockSubsystem Block as used everywhere else in the model, this underlines the fact that each diagram is a view of the model not the model itself. The Flow Ports that were introduced in Figure 45 and values introduced in Figure 41 reinforce the idea.

Examination of Rolling Stock specifications and standards such as the Intercity Express Train Specification (Department of Transport, 2010) or London Underground’s Standard for Rolling Stock (London Underground, 2008) was used to identify a general number of subsystems. An analysis of these subsystems revealed those that are the necessary subsystems for a system to be Rolling Stock and only Rolling Stock.

Clearly one of the most important elements required of something to be GTS Rolling Stock is some method of following the Guideway. This author has chosen the term RailWheelSets to represent this method, as wheels/rollers are used in traditional railway, tramway, metro type applications of GTS’, also they are used on Monorail systems to keep them on and following the rail (Japan Monorail Association, 2016), also where linear motors are concerned (Anon, 2016). The RailWheelSets Block is modelled in a 2 or more composite relationship with the

RollingStock Block, representing that it is a necessary system and there must be a least two sets (the assumption is that wheel sets are made up of two wheels).

The Rolling Stock specifications referenced in this thesis (Department of Transport, 2010 and London Underground, 2008) specify brake systems for Rolling Stock, therefore braking systems are included a necessary subsystem of Rolling Stock. This is represented by a BrakingSystems Block that is modelled as belonging to the RollingStock Block through a 1 to many relationship, showing that at least one braking subsystem must be present and there can be more than one, for example, air brake, electric brake and parking brake.

Rolling Stock will always have some form of body, even if it is a flatbed Permanent way worker's tools and equipment trolley, but it is very unlikely to have more than one, therefore the relationship between the RollingStock Block and Body Block shows that one Rolling Stock element has one body.

It was surprising to this author when the analysis of necessary subsystems revealed that a control system would be a necessary system of Rolling Stock. However, even a simple brake system on a freight wagon needs some form of control from the locomotive hauling it. There could also be a myriad of control and communications systems on board more complex types of Rolling Stock. Therefore, the Control&Communications Block is modelled as having a one or more composite relationship with the RollingStock Block. Meaning that one Rolling Stock element must have one, but can have many control and communications subsystems.

The RollingStock Block has a 1 or more relationship with the ExteriorSystems Block because there is always something on the outside of Rolling Stock, even if it's just a bracket for a lamp, but often there can be some quite complex systems, informing passengers of Routes and destinations, steps to allow access, horns, loud speakers, CCTV and other detection devices.

The RollingStock Block is in a one to many composite relationship with the Fire&EmergencySystems Block. This captures that Rolling Stock always has some form of emergency system, but some Rolling Stock can have numerous emergency systems, for detecting and suppressing fire, warning passengers of problems on a train, passengers warning operators of problems on a train etc.

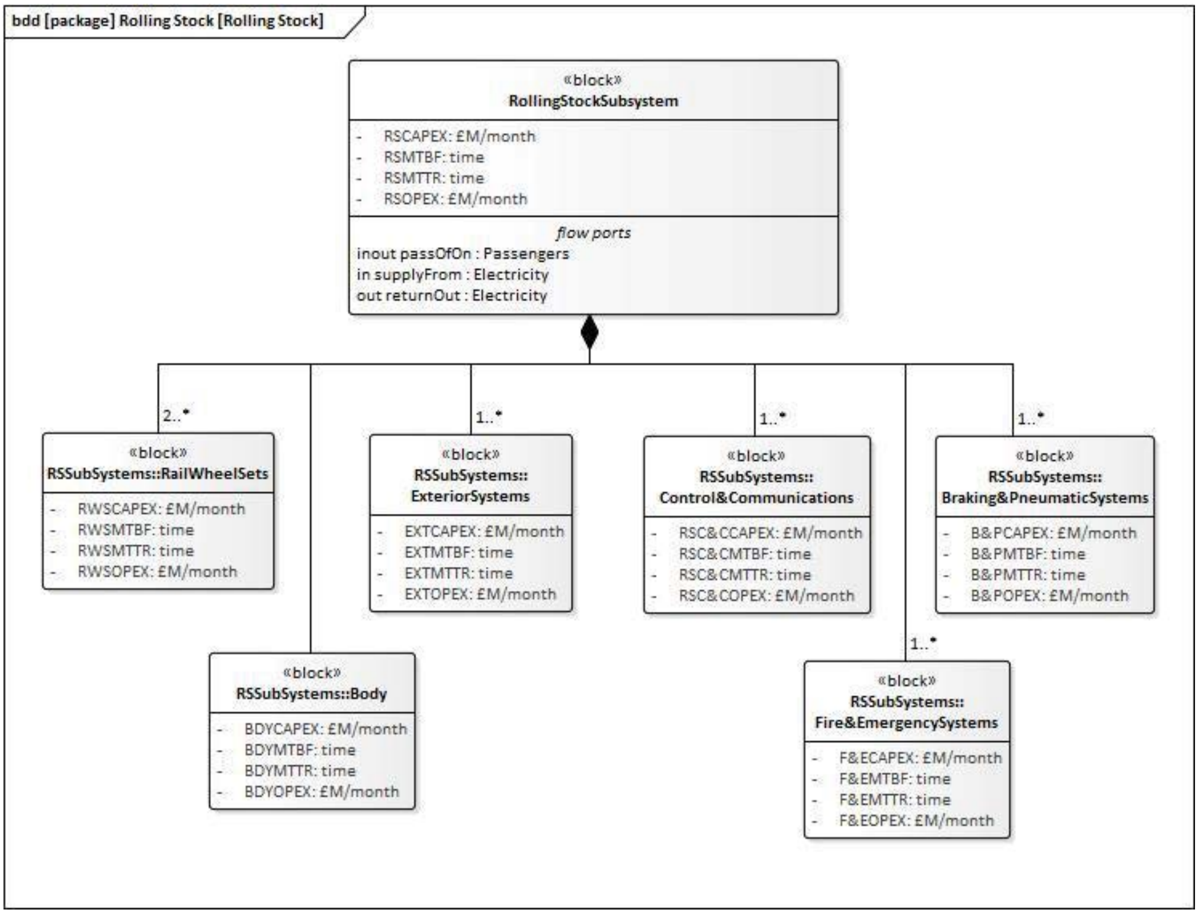

Figure 47 Generic Rolling Stock Block Definition Diagram (author)

This generic model of just the minimum required elements of Rolling Stock on a GTS has its uses in providing a starting place for a modeller, by identifying the absolutely necessary subsystems. This would not produce a model of anything useful on its own. As discussed previously the subsystems that are deployed depend on the type or kind of Rolling Stock required. Rolling Stock can be divided in many ways depending on its application, power supply, size or type of GTS it operates on. The BDD below in Figure 48 shows some of those kinds of Rolling Stock that are part of the underlying model. Others can be added should it be necessary; this demonstrates the ability of this approach to model “more GTS” through extension of the generic model.

The BDD below in Figure 48 shows a Block at the very top of the hierarchy representing a Train, as stated above a Train is a collection of Rolling Stock, so the relationship between the Block representing a Train and the RollingStock Block is a composite one, which shows that one Train is made up of one or more RollingStock Blocks. There are a number of other Blocks under the RollingStock Block representing distinct kinds of, but still generic Rolling Stock.

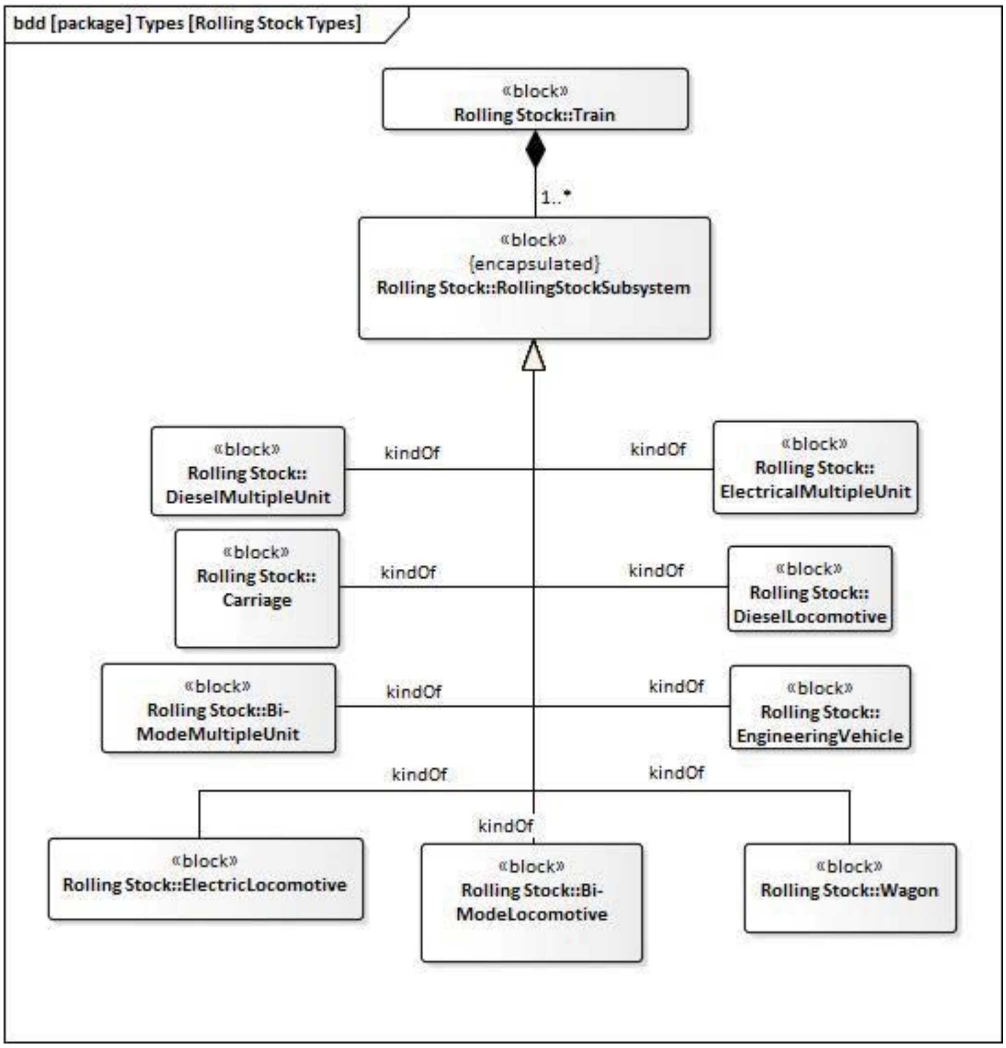

Figure 48 Kinds of Rolling Stock (author)

All the Blocks are related to the Rolling Stock Block via an Generalisation or 'kind of' Relationship. This is represented in SysML by a line between the Blocks with a hollow arrow head at the top of the relationship. This means that all the Blocks connected to the RollingStock Block through this relationship inherit its properties. These can then be added to or removed to specialise the Block to become a particular kind of Rolling Stock. The generic model is now set up to model (arguably) most kinds of Rolling Stock.

The BDD below in Figure 49 shows a more specialised RollingStock Block. As the name of the top Block suggests it's an electric locomotive, which is one of the kinds of RollingStock identified in Figure 48. It has composite relationships with a number of subsystems other than the necessary subsystems defined in Figure 47 that make it more specific. It owns a PowerCollectionSystems Block. It does not specify what sort of power collection system, so it is still solution independent, even though it is an electric locomotive it is a generic electric locomotive. It can also have more than one instance of PowerCollectionSystems, this is to capture that an electric locomotive can have more than one kind of power collection system. The same applies to the PropulsionSystems Block there can be more than one propulsion system.

There can be more than one instance of the InteriorsSystem Block associated with an ElectricLocomotive Block, as there can be more than one interior subsystem, for example the cab lighting, horns, drivers seating etc. can all be considered separate subsystems at different levels.

The ElectricLocomotive Block has a user defined relationship with Couplers of 1..2, this is to capture that there is usually a primary coupling system for normal use and an emergency coupler for degraded modes and rescue.

There can be a number of auxiliary systems on a locomotive of any type, examples being heating/air conditioning, power control system, air system and battery charging. This is modelled by the composite relationship between the ElectricLocomotive Block and AuxiliarySystems Blocks of one or more. There is at least one auxiliary system, even if it's only a compressor to supply air for brakes and control.

The relationship between the DoorsSystems and ElectricLocomotive Blocks shows that there can be more than one door system. This is correct because there will be doors to allow the operator on and off the locomotive, but there can also be doors that allow the operator and maintenance staff into the equipment areas for maintenance and repair purposes.

As this view of the underlying model is of a Locomotive there will be only one heating and ventilation system onboard as there is no separate passenger compartment. In older locomotives there may be no heating, but now days it is very unlikely, so the one to one relationship is deemed to be correct.

The relationship between Bogies and ElectricLocomotive is defined as there are zero or more instances of the Bogie Block in every instance of the ElectricLocomotive Block, this is to capture that some locomotives have more than two bogies, but there are rare cases where there are none.

The six subsystems identified above in Figure 49 are inherited from the generic RollingStock Block. The eight ElectricLocomotive Block specific subsystems are shown as separate Blocks related to the ElectricLocomotive Block. This makes it easier to focus on the extra Blocks that the ElectricLocomotive Blocks owns.

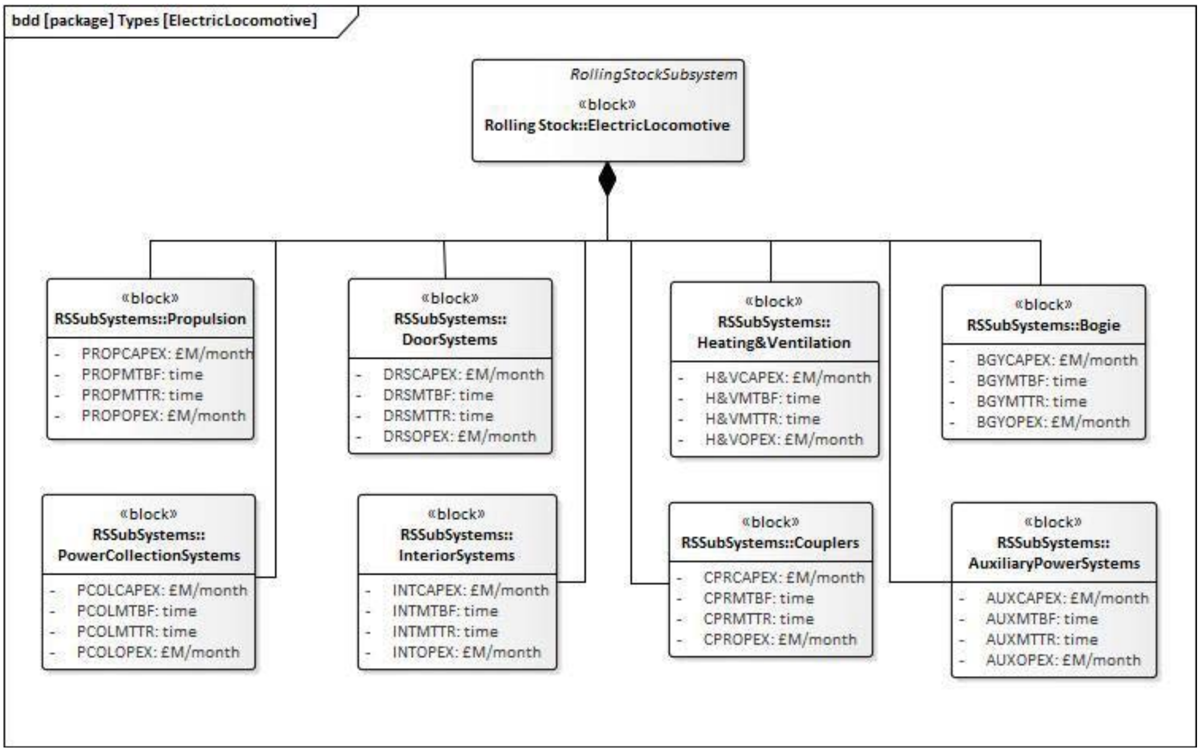

Figure 49 Generic Electric Locomotive BDD (author)

Note from the diagram above each of the Blocks has values for CAPEX, OPEX, MTBF and MTTR, these were inherited from the RollingStockSubsystem block but have been specialised. This is to model the idea described in chapter 5.5, of KPIs at lower levels of model abstraction contributing to the final KPIs expressed at the higher levels. This is done through the same mechanism of Constraint Blocks and Parametric Diagrams as described in chapter 5.7. This is shown in Figure 50 just for MTBF, as a demonstration. The BDD below shows the ElectricLocomotive Block and its Constraint Block for calculating the system MTBF of the electric locomotive from all of its identified subsystems. This is a simplification for demonstration purposes. It is assumed that manufactures supply equipment with its reliability expressed in Mean Time Between Failures (MTBF), rather than some form of failure rate per time period (which for calculations is more useful). Therefore, a simple way to get an MTBF for the complete system (electric locomotive) is to turn all of the MTBF figures for the subsystems into a failure rate, 1/MTBF, these can then all be added together as failure rates and then returned back to an MTBF for the whole system. The full equation is shown in the constraint compartment of the BDD in Figure 49. This process can be applied to each level of abstraction right down to component level in order produce an accurate MTBF for the complete electric locomotive. The MTBF for the whole GTS system can be derived through the application of this pattern, as all the other subsystems of the GTS have the same parameters and ability to nest them.

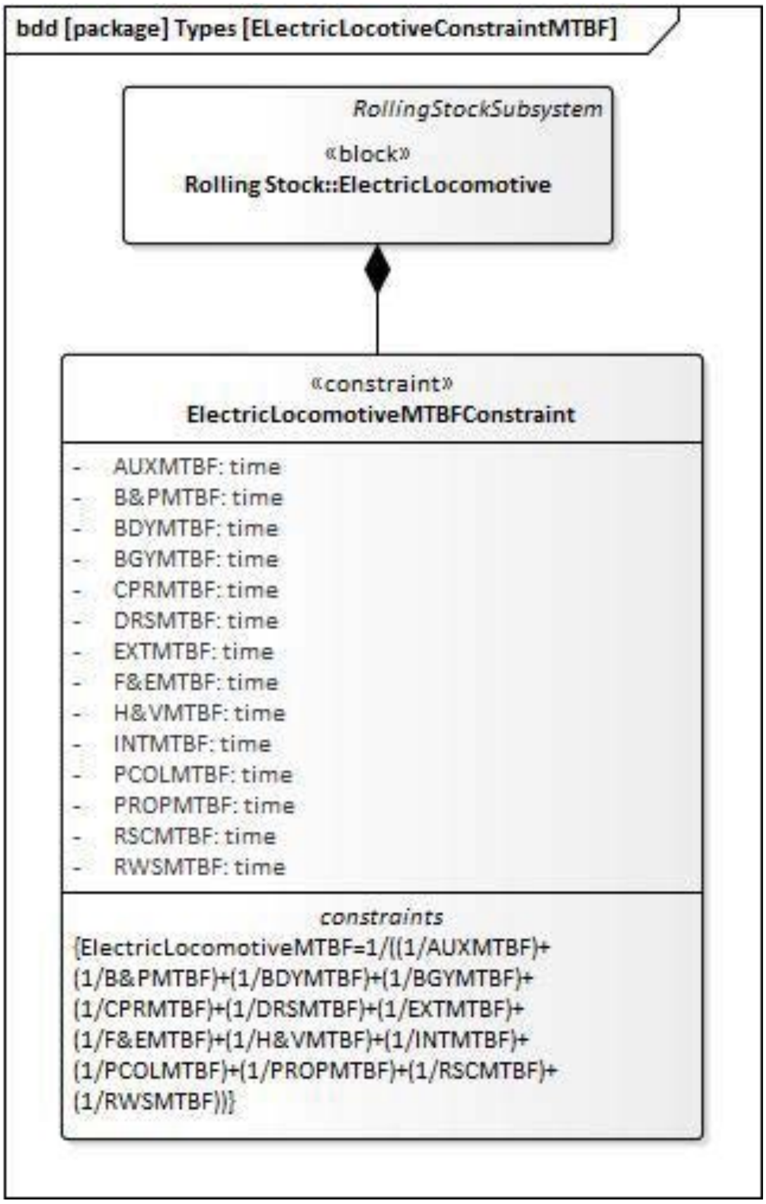

Figure 50 MTBF Constraint Block for ElectricLocomotive Block (author)

The Parametric Diagram for the ElectricLocomotive Block is shown below in Figure 51. It clearly shows where the value of each parameter of the Constraint comes from.

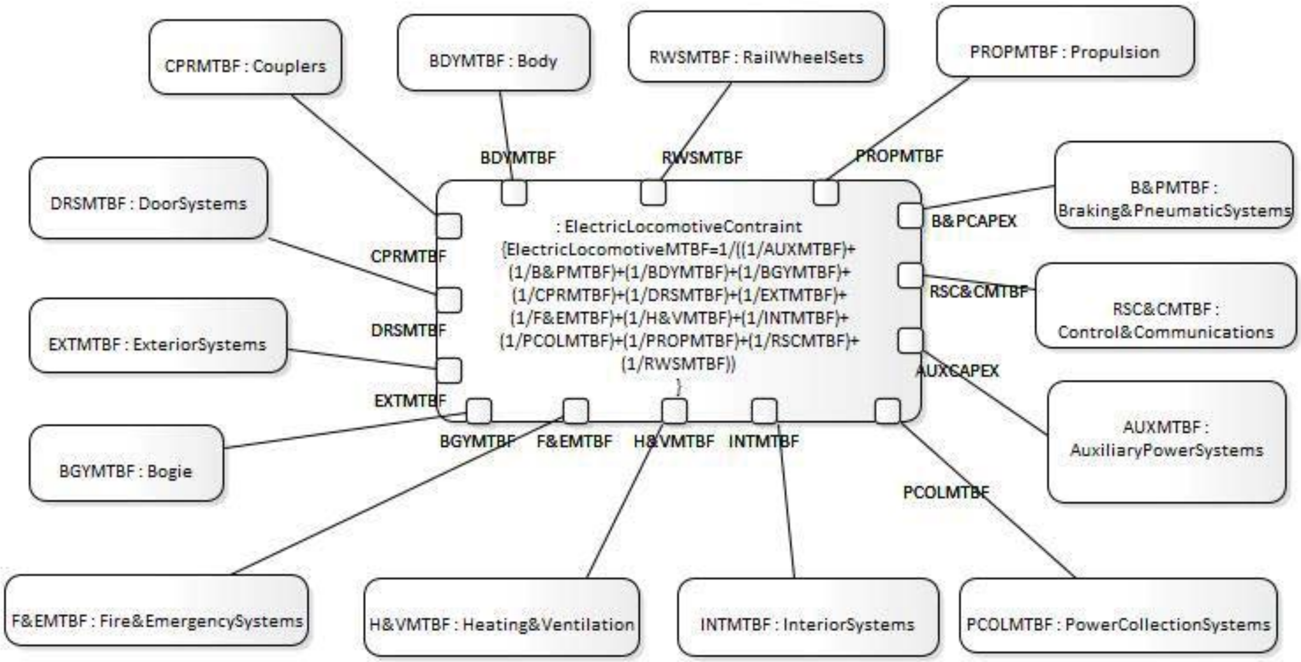

Figure 51 Parametric Diagram for ElectricLocomotive Block MTBF (author)

6 Results

6.1 Introduction

In order to test the assertions made in chapter 2.5 from Elphic, (2011) etc, regarding the difference in understanding of what Systems Engineering is and its current application within the Railway industry, it was considered necessary to get an understanding of the actual application of Systems Engineering and thinking within the Railway industry from a group of practicing engineers within the industry. Also, as a large part of this work is to suggest that a general modelling methodology, used across the industry could facilitate a more universal application of Systems Engineering with all its attendant benefits, it was deemed efficient to get the same group to evaluate (at least part of) the representative methodology developed in chapters 4 and 5. To be useful this sample needed to be as broad as possible.

The evaluation of the representative methodology in particular would be a substantial piece of work to ask a reasonably large number of busy practicing engineers within the industry to undertake on top their daily work. However, such a number of practicing engineers, from (mostly) within the Railway industry, are the students on the University of Birmingham MSc in Railway Systems Engineering & Integration course. This was thought to be a good sample of practitioners because the course is a part time course, which traditionally has a very high proportion of actual practitioners from within the Railway industry and from various career levels and disciplines. Also the students are not all from the UK.

The author was given permission by the university to design a group activity for the Systems Engineering unit of the MSc course. This was to be used as the vehicle to gauge the depth of understanding and application of Systems Engineering within the industry and to evaluate the Model Based Systems Engineering (MBSE) methodology developed in chapters 4 and 5.

The group activity was to be given, along with instruction on the proposed MBSE methodology, over the week the Systems Engineering unit took place at the University. The group activity consisted of 2 surveys, lectures/talks on the application of some aspects of the MBSE approach, the application of those aspects of the approach to a railway systems engineering problem and a final presentation of the results to the class and lecturers.

The purpose of the first survey, carried out before the group activity began, was to provide a baseline. Its aim was to:

- identify the industry in which the students worked;
- identify the students engineering discipline (or other if applicable);
- identify the students work role and level;
- find the level of understanding each student has with respect to the terms System and Systems Engineering;
- Identify each student's application of Systems Engineering in terms of time spent on Systems Engineering type work and use of Systems Engineering tools and processes.

The second questionnaire, carried out after the group activity was completed, was designed to allow the students to evaluate the proposed MBSE approach based on the Systems Engineering problem they were given once the group activity had begun.

6.2 The Survey Sample

Based on the data supplied in the 1st questionnaire, 36 students took part in the group activity. They came from the Railway industry, from academia and other industries, their industries and disciplines are detailed in Table 4 and Table 5 below.

Rail Disciplines	
Communications	1
Infrastructure	11
Power/OLE	1
Railway Planning	1
Rolling Stock	5
Signalling	5
Total	**24**

Table 4 Railway disciplines represented on the UoB RSE MSc course

Non rail Disciplines	
Full time Students	7
Oil and Gas	1
Civil Engineering	2
Automobile	1
Real Estate	1
Total	**12**

Table 5 Non-Railway disciplines represented on the UoB RSE MSc course

The positions within industry the students currently occupy was also diverse, a list of the job titles claimed by the students is given below:

1. Project Manager;
2. Program Manager (capital projects);
3. Project Engineer;
4. Project Director;
5. Manager;
6. Senior Engineering Team Member;
7. Construction Engineering;
8. Systems Engineer;

9. Civil Engineer;
10. Track Maintenance, Track Engineer;
11. Student;
12. Power Engineer;
13. Depot Manager (Rolling Stock);
14. Rolling Stock Maintenance Engineer;
15. Production Engineer;
16. Test Engineer;
17. Signalling Principles Design;
18. Signalling Engineer;
19. Manager Point Maintenance Branch;
20. Team Leader;
21. Asset Manager;
22. Electrical Engineer;
23. Mechanical Engineer; and
24. Assistant Tower Engineer.

6.3 The Group activity

6.3.1 Purpose of the Group activity

The group activity had three objectives:

1. provide useful learning and experience on the subject of MBSE to the students taking part;
2. Provide the author with a broad based data on the application of Systems Engineering and Systems thinking in the Railway industry; and

3. Evaluate the usefulness of the general modelling approach to Railway Systems modelling developed for this project, in terms of facilitating the use of Systems Engineering and Systems thinking.

6.3.2 1st Questionnaire

The 1st questionnaire was designed to establish base data for the students taking part in the activity and to test/evaluate the assumptions made in chapter 2.5 regarding the lack of Systems thinking and Systems Engineering applied across the Railway industry. It was completed before the group activity had begun. It was carried out as early as possible so the course content of the Systems Engineering unit did not influence the opinions the students offered and hence affect the base data. This questionnaire (see Appendix C) asked the students some fundamental questions about what they thought was meant by the terms System and Systems Engineering. This was to establish what, if any, common ground existed within the sample of engineers and what differences existed. The survey then asked the students how much of their time at work was involved in Systems Engineering, given the definitions they had previously given. They were given 4 choices; Majority, Half, Rarely and Never. The reason for asking this question was to establish how much Systems Engineering that the students themselves thought they did, regardless of what they actually did.

The survey went on to ask the students if they applied certain ubiquitous Systems Engineering tools, not just MBSE tools. Asking this was to establish two points:

1. Were the students actually performing Systems Engineering type tasks, but did not really notice because they were not “badged” as Systems Engineering?
2. Did the students think that they were doing Systems Engineering, but were actually not?

These answers were to be compared with the previous answers to build a clear picture of the level of Systems Engineering actually being applied.

The following chapters show and evaluate the answers given in this questionnaire.

6.3.3 The definition of a System

The students were first asked to give a definition for the term System. The answers received were all in plain language, so in order to analyse them the author has grouped together key similar phrases used or attributes identified by the students. To establish commonality they are shown in the graph Figure 52, against the number of students that used them. Unsurprisingly, the majority of students, 31 (79.49%), used a phrase that meant, things working together (People, systems etc). However, after this initial agreement there is significant divergence. Nearly half the students, 17 (43.59%), cite achieving a purpose, this is interesting as a key point of Systems thinking and Systems Engineering (see chapter 2.2) is that a system is designed to achieve a purpose, yet less than half of the Students on the course mentioned it in their definition. Also only 7 (17.95%) of the students mentioned interfaces in their definition. This was unexpected, as interfaces are mentioned frequently in Systems Engineering literature, although this could be explained by the fact that a number of the students had already mentioned things working together in their definition. However, this does hi-light the fact that often interpretation of answers to questions is necessary, but can possibly lead to inaccurate findings. A portion of these students could just not have thought about interfacing.

Only 2 (5.13%) of the students mentioned that a system exists in a given environment, the same number also mentioned that a system has boundary. It is more difficult to reason that these can be included in other answers. It is significant because these are two key elements of Systems Engineering and thinking which if not considered can lead to unexpected arising

System properties once it has been implemented. There was also some confusion between what a System is and Systems Engineering itself, hence some students cited: communications, the way something works and black box thinking as part of their definition of a System as opposed to a definition of Systems Engineering.

Given the answers received it is clear that there is significant divergence of opinion on the definition of a System, particularly after the initial identification that it is various things working together. It underlines the importance of providing a definition of a System for this work as in chapter 2.5, as it helps practitioners start from the same understanding.

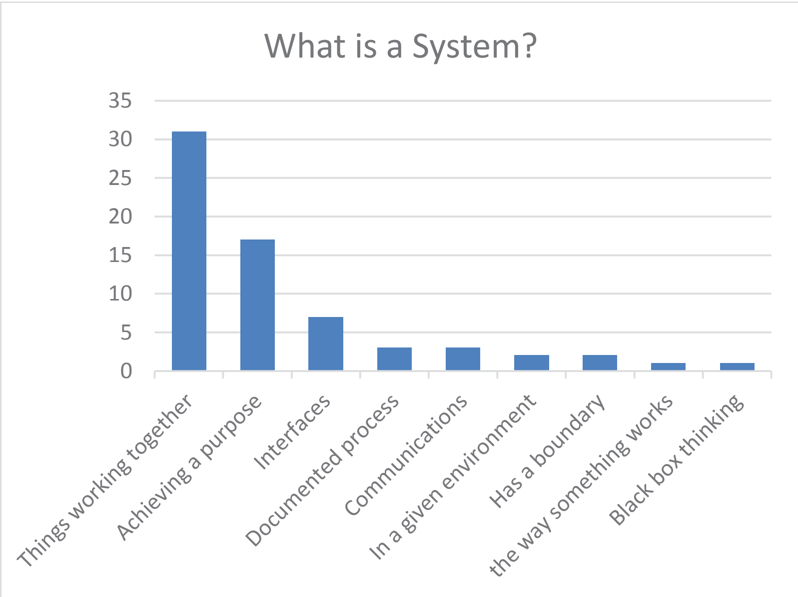

Figure 52 What is meant by the term system (author)

6.3.4 Definition of Systems Engineering

Next the students were asked to define Systems Engineering. This question had two aims:

1. Establish how many of the students understood what Systems Engineering is; and
2. Establish what common understanding existed across the sample group.

As with the previous question, phrases, attributes and key words were grouped together to allow analysis. They are shown below in Figure 53 against the number of students that used them.

With this question there was less agreement than with the previous question. The most common response, given by 12 (33.33%) students, stated that Systems Engineering had an element of methodical management of Engineering Design, which included the technical design and its processes and procedures. One reason for this could be that a significant number, 8 (22.22%), of these students had a design biased background, either actually designing Systems or managing projects that involved new designs. Interestingly 3 of the full time students also identified aspects of design management. Only 10 (25.64%) mentioned Systems objectives or requirements in their response, which is interesting given the profile that this element of Systems Engineering has. Less than a quarter (9) considered Subsystems working or interacting together in their response and only 7 (17.95%) mention interfaces, this collates with question 1, where interfaces were not mentioned to a large degree.

It was interesting after these major themes, which it could be argued would be expected, some of the more technical aspects of system engineering were mentioned by relatively few students. Such as lifecycle (5 or 17.95%). Interdisciplinary processes (3 or 7.39%), although this could be added to multi-disciplinary (2 or 5.13%) to give a total of 5 or 12.52%. Also some of the responses are very discipline specific such as Train Maintenance and improvement. Some seem at cross purposes to the idea of Systems Engineering, such as, separate Systems Engineers for separate disciplines, although this shouldn't be a surprise given that the Railway industry has many job titles that end in systems engineer, such as brake systems engineer, signalling systems engineer etc. The results of this question are in agreement with the assertion that many engineers have a very different idea of what Systems Engineering might be.

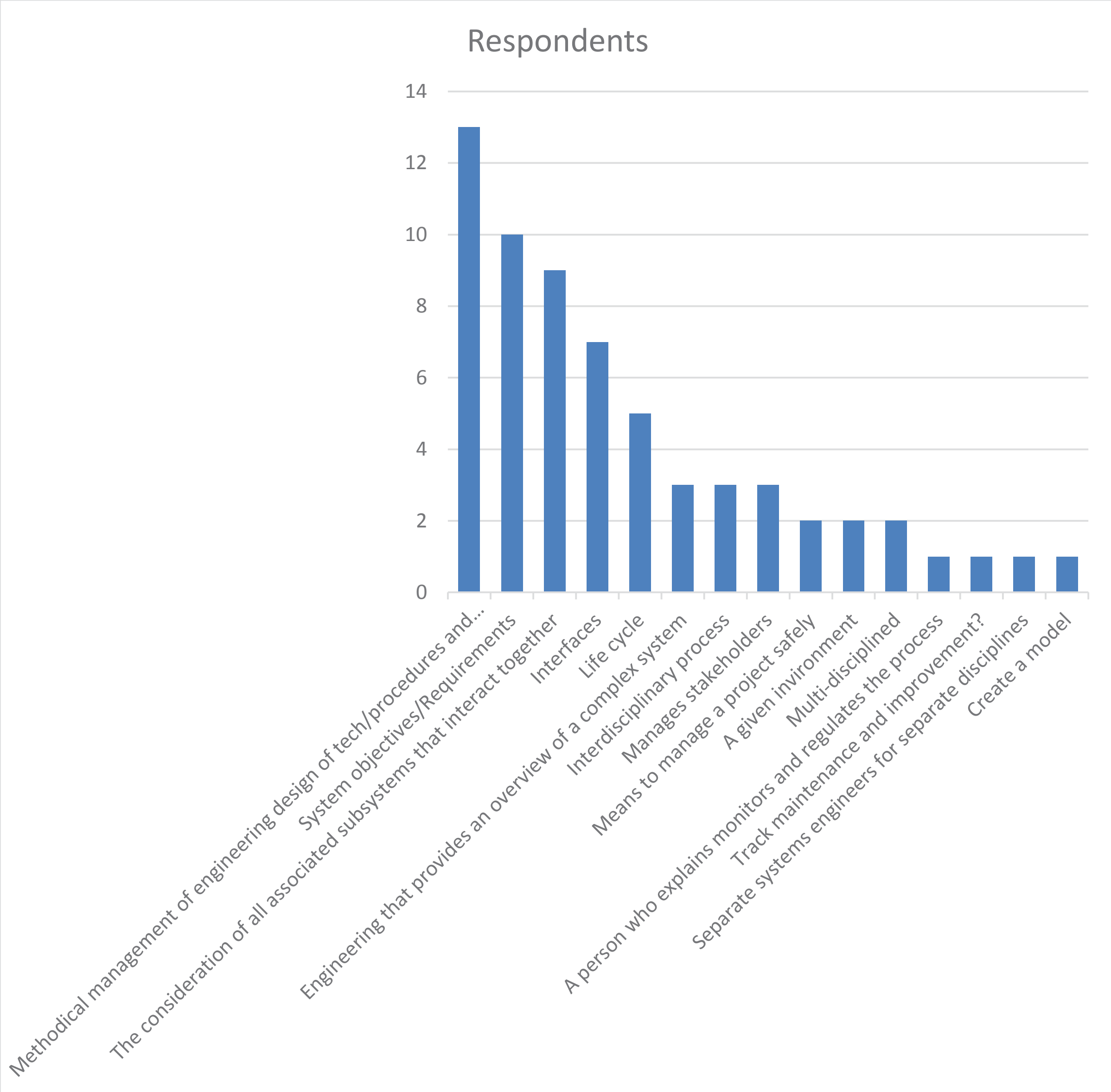

Figure 53 Definition of Systems Engineering (author)

6.3.5 Involvement in Systems Engineering

The next question asked was, given your previous definitions do you consider yourself involved in Systems Engineering in the course of your normal work. The students were given only four choices of answer, majority; half; rarely and never. The reason for giving only four choices was it was not thought necessary to have the students try and give an accurate percentage type answer as it would impede meaningful analysis and grouping of the answers.

The following results were received:

Answer	Number of replies
Majority	8
Half	14
Rarely	5
Never	6
No Answer	3

Table 6 Involvement in Systems Engineering

From these results it is clear that the majority of the students, 22, thought they worked in Systems Engineering for a substantial part of their working life, half or more. Of the 8 students that claimed to work in Systems Engineering the majority of their time, 6 came from the Railway industry (3 from infrastructure, 1 from planning, 1 from rolling stock and 1 from signalling). The remaining 2, from other industries, were 1 from Oil and Gas and 1 from real estate. Of the 14 that stated that half of their time was spent in Systems Engineering, 13 were from the rail industry (5 from rolling stock, 5 from infrastructure, 2 from signalling and 1 from Power/OLE). The other student was from the automotive industry. Of the 5 that stated that they rarely worked in Systems Engineering four were from the Railway industry (1 from communications, 2 from infrastructure, 1 from signalling) the remaining student stated that he wasn't from the Railway industry, but a civil engineer. Of the six students that stated that they never work in Systems Engineering, 3 were full time students 1 was a civil engineer from outside the Railway industry the remaining two were from the Railway industry (1 infrastructure and 1 signalling). Therefore, it was clear that nearly all of the students at least thought they had some experience of Systems Engineering.

6.3.6 Systems Engineering tools and practices

The remaining sections of the questionnaire dealt with the application of Systems Engineering tools and practices. The previous questions dealt with what the students thought they were doing in their workplaces; these next questions were designed to gain an understanding of what the students were actually doing in their workplaces.

The following questions were asked:

1. Is there a documented Systems Engineering process at your place of work?
2. Please add a brief description;
3. Have you used any of the following in your work?
    a. Context diagram;
    b. Systems Engineering Management Plan (SEMP);
    c. Interface Management Plan (IMP);
    d. Systems Engineering Modelling Language;
    e. Systems Engineering Modelling Tool;
    f. Ontology Modelling Tools;
    g. Proprietary Company Modelling Tools/Languages.
4. Is there a formal Change Management Process at your place of work? and
5. Is there a formal Stakeholder Management Process at your place of work?

These questions were designed to cover a number of common Systems Engineering tools and practices that might be expected to be present where Systems Engineering is being applied, they were not intended to be an exhaustive list of all the tools and processes that are available under the title of Systems Engineering.

Question 1, was asked so the author could compare the number of people that thought they used Systems Engineering against those who actually did. It is reasoned that an organisation that did Systems Engineering would have some form of documented process for engineers to follow, to ensure Systems Engineering was applied uniformly across the organisation and its projects, even if it is just adherence to standards on the subject. This is particularly relevant where the answer to the question on how much Systems Engineering a student did in their workplace was the majority or half of their time. It is also reasoned that these people are the most likely to know about a Systems Engineering process being in place rather than someone who only uses it rarely. The answers were as detailed in Table 7 below. Less than a quarter (8) of the students who stated they spent the majority or half of their time doing Systems Engineering stated that there was a documented Systems Engineering process in place. This number was unexpected given that 22 students believed that half or more of their work involved Systems Engineering. This is important because Systems Engineering can be said to be process driven. This being the case it could reasonably be expected that the majority of the 22 students spending half or more of their time in Systems Engineering to have, and indeed know of, a documented process in their place of work. This also raises the question, what was in place in the places of work of the remaining 14 students and how was it applied? It was noted that one student who stated that she/he rarely did any Systems Engineering also stated that there was a documented Systems Engineering process in the workplace.

Answer	Number of replies
Yes	9
No	18
Yes/No	1
No Answer	9

Table 7 Documented Systems Engineering process at place of work

Another observation is that the job titles claimed by those 22 students who spent half or more of their time doing Systems Engineering was spread right across the range from assistant engineer and student to senior manager. However, of those 10 who answered yes or yes/no to the question about a documented Systems Engineering process, 6 came from the rail industry and 3 of those claimed some sort of management role in their job title, the remaining 3 were all engineers and they all had some sort of project related job.

Question 2 required a brief description, for which there were 10 replies, as expected. These replies were significant from the point of view of understanding any practical application of Systems Engineering. The answer yes/no, which at first did not make sense, becomes clearer once the description is taken into consideration. The student's answer was:

> *"We have for instance a safety management system that describes how we must manage safety in the Railway. Similarly, for project management. However, I am not aware of a process for Systems Engineering per se".*

The student correctly identified that, as far as the Railway is concerned, safety engineering is a Systems Engineering function. It needs to consider risk and change in one part of the Railway along with effects to all other parts of the Railway, although it is not necessarily badged as Systems Engineering, or is not part of a wider approach. Although when answering question 3, the student stated that he/she had used both a SEMP and an IMP at work. So although there was not an actual Systems Engineering process in place at that place of work, clearly some Systems Engineering was being carried out. These particular answers hi-light an important point from an organisational point of view, if there is no actual documented Systems Engineering process in place in the company, are the tools and practices identified being applied universally within the organisation, or are they just applied by people who know about them? This student also noted that there was no formal change management or Stakeholder management processes in place. This student is a program manager; the title indicates that this person is in control of multiple individual projects that interact with each other.

A Design Manager who stated the majority of his/her time was spent doing Systems Engineering responded that there was a documented Systems Engineering process at his/her place of work and then went on to describe it as:

> *"This is provided through standards on design and management through internal and client procedures, through design requiring inputs from others and by the fact that a design not working in practice.".*

This student also stated that he/she used a Context Diagram, SEMP and IMP at their place of work. These answers point to an integrated process in place, although again this is not explicit, also if it is reliant on client processes does it change with different clients?

A Project Engineer who answered that he/she spent the majority of time at work doing Systems Engineering and stated that there was a documented Systems Engineering process in place described it as follows:

> *"We work to Network Rail (NR) processes following GRIP stages and NR/2009 assurance. There is NR guidance on the Systems Engineering process applied to projects (1-209) and we have Engineering Management Plans and Systems*

> *Engineering Management Plans for programs or projects. We follow the V life-cycle which we map to the GRIP process and Transport for London (TFL) pathways governance process".*

This is clearly a comprehensive companywide process, driven by National standards.

Another student who responded that he/she spent the majority of time doing Systems Engineering claimed the job title Project Engineer. She/he also stated there was a documented Systems Engineering process in place and described it as:

> *"I work on rail infrastructure projects and building new infrastructure. This is to Railway standards and all disciplines must be considered e.g. when building a track platform, gauging is a consideration. When building civil assets signalling assets need to be considered e.g. signal sighting."*

From the author's own experience working on British Railway's, this points to using Network Rail Standards, National Notified Technical Rules and Technical Specifications for Interoperability, which although they deal with the Railway System, they are not really a Systems Engineering process on their own. They would be referenced or called up from a Systems Engineering process.

One of the students, who answered yes to the previous question stated

> *"Not aware of the full documented process, however I'm aware of the interdisciplinary check (IDC) process. This is where each subsystem is reviewed together to ensure the overall design meets the requirements."*

The same student also stated that he/she used a SEMP and an IMP in the course of their work, also that there are formal processes for Change and Stakeholder management, Again, this points to a formal Systems Engineering process being in place.

A field signalling engineer stated she/he spent half of their time doing Systems Engineering and also stated that there was a documented Systems Engineering process in place and went on to describe it as:

"In my company, the projects are managed in accordance to EN50126 (British Standards Institute, 2010). Where the basic principle is the V cycle diagram to drive the Engineering actions during the project life cycle, so we have two different teams coordinated and each one is responsible for one side of the V cycle, they are:

- *Design coordination is responsible for the design phase, collecting all functional requirements, interface technical and risk management and project management; and*
- *Implementation coordination is responsible for the build and testing phase, requirement system validation and commissioning of systems."*

Again this clearly points to there actually being a process in place.

Another test engineer stated that there was a documented Systems Engineering process at his/her place of work, the description is as follows:

"I am not involved in this documentation process. The documents that I create respectively handle test specifications, test procedures and test reports for tests that my colleague and or I carry out. Most of these tests are validation tests. These tests have to be carried out by an independent test lab. The test lab delivers the test reports and the System engineers use these for the System documentation".

It is difficult to see how this description shows there is actually a documented process in place.

The remaining two Students who both answered that they spent the majority of their time doing Systems Engineering and that there was a documented Systems Engineering process in place were not from the Railway Industry. The student from the oil and gas industry described the systems engineering process in their place of work as:

> *"We mainly use process flow diagram (PFP) and Process and Instrumentation Drawing (PID) to illustrate the gas pipe system."*

This Student also stated that she/he used a context diagram at work. However, the process described did not seem to be a Systems Engineering process as such, it is more of a design management process. The other Student from the Real Estate Industry described the process in place as:

> *"In the planning process of real estate construction, there is a major procedure that provides effectively each planning process. Normally, my company has adapted the software to create documents to support the working and planning processes".*

The author is unsure what this actually means. However, the student did claim to use both a context diagram and a SEMP.

There was a significant number of students that claimed to work the majority or half their time in Systems Engineering, who stated that there was no documented Systems Engineering process in place at their workplace 11 and 3 respectively. However, 5 of these claimed to use one of or all of context diagram, SEMP or IMP. One of the students also gave the following description:

> *"Systems engineering is embedded in our design process, where we design a product to satisfy a set of requirements. During the design phase different disciplines*

collaborate to ensure that their different designs fit together. At installation and testing designs are verified and evaluated to build a dependable system".

It could be assumed from this description that there is actually a process in place in this student's place of work.

A summary of the number of students that used the various example tools is given below in Table 8.

No Answer	18
Systems Engineering Management Plan (SEMP)	12
Interface Management Plan (IMP)	9
Context diagram	9
Systems Engineering Modelling Tool	1
Systems Engineering Modelling Language	1
Ontology Modelling Tools	0
Proprietary Company Modelling Tools/Languages	0

Table 8 Systems Engineering used by the students

6.4 Representative MBSE Approach

6.4.1 Introduction

The next part of the group activity was to teach the approach developed in chapter 4 to the students, have them apply it and then evaluate it.

It was not possible to explain the whole MBSE approach proposed in this thesis, derive exercises to reinforce the learning and get an evaluation of the approach in the time allowed for this part of the unit (approx. 7.5 hours). It was decided therefore, not to teach a System modelling language, but to give an overview of the approach and then explain some of the most important parts with an exercise to re-enforce the explanation. All the diagrams were to be created using standard drawing tools. This approach did mean an important part of

MBSE would be missing, the central model. As will be seen in the following chapters this was seen as a failing by a number of students.

The areas the author chose, in consultation with the course lecturers, to concentrate on were:

- Context modelling;
- Requirements modelling (including SysML type requirements diagrams and views and viewpoints);
- Activity modelling (including Use case models and SysML type activity diagrams); and
- Functional decomposition.

It was explained to the students that they would be looking at an MBSE approach to Systems Engineering, developed as part of the authors PhD research, they would be asked to apply some of the key parts of the process to a Systems Engineering problem and demonstrate that application with a presentation to the class and lecturers. They were also required to complete two questionnaires, one of which would be marked, which evaluated the parts of the MBSE approach that they had used during their group activity.

The group activity was based on a set of overall hi-level requirements, for introducing a new train onto the Railway, on which they were asked to apply the MBSE approach. The hi-level System requirements were:

"*There is a proposal for new trains to run from London Paddington to Bristol Temple Meads. These trains will provide a semi-fast service on the Great Western Main Line.*

The service will be every 30 minutes in both directions.

The new trains will stop at:

1. *Reading;*
2. *Didcot;*
3. *Swindon; and*
4. *Bath Spa.*

Each train shall have the capacity of carrying 600 passengers, 20% in first class and 80% in standard class.

The Great Western Main Line is being electrified so the trains should be capable of operation on 25kV overhead line and diesel operation. In both modes the top speed shall be 125mph. These are the high-level requirements from the government."

The class was then split into groups and the whole MBSE approach was explained to them. The SysML language was used to demonstrate the various concepts as per the approach developed in chapter 4. The students were not required to use SysML or any other language for the exercise, due to time pressures. This was followed by an explanation of context diagrams, their development and usage.

The groups were then given the first task, which was to develop a context diagram and use this to identify all of the Stakeholders for the hi-level requirements. The aim of this part of the exercise was for the students to use a context diagram similar to that developed in chapter 4.3 which they themselves had developed, so they could:

1. Set their system (a new train) within its environment (the Railway and the wider world where appropriate);
2. Establish their System boundaries; and

3. Identify Stakeholders, people and other Systems, both within the Railway and those from outside the Railway, to ensure all of the Stakeholders have been identified as far as is practicable.

For the remainder of the exercises each group was given a particular set of Stakeholders to think about in detail, this was due to a concern from the course lecturers that all of the groups would have very similar results otherwise. This in itself demonstrated the potential of a process such as the one developed here to drive consistency in the application and management of Systems Engineering across an organisation.

The next exercise was devoted to how the proposed MBSE approach dealt with requirements. The students were given an explanation on requirements, requirements diagrams and model views and viewpoints with reference to the proposed approach. The students were then asked to:

1. develop a viewpoint and a view of the model for a Stakeholder, with a SysML type representation; and
2. develop the next level of requirements from the hi-level requirements they were given at the start and capture them using a SysML type requirements diagram.

The aim of this exercise was for the students to start thinking about decomposing requirements, but most importantly how that decomposition can be carried out, captured and then controlled. A second aim was to understand that each requirement in the decomposition can and should be traced back to a top level requirement using the MBSE approach. This underpins the notion that if a requirement is not in some way helping to deliver the top level requirements then they should question why it is there.

Next the students were given a demonstration of use cases and activity modelling in general and then in line with the approach developed for this thesis. Each group was then asked, with respect to the Stakeholder or Stakeholder group they had been assigned to:

1. Create a Use Case for a particular user function of the System, including a use case specification and a use case diagram; and
2. Create a dynamic model, using a SysML type activity model, of some function a user might require from the System.

The main aim for this part of the exercise was to show the students that there were other ways of generating important System requirements that need to be taken into account when designing a new System or changing an existing System, in order for it to deliver the functionality needed. In this case demonstrating how modelling the way users will use the system once implemented provides a systematic approach to understanding that usage and also how it generates functional requirements that were not identified through the traditional requirements specification from the customer.

The final part of the group activity was functional decomposition. As previously the students were given instruction and a demonstration on functional decomposition, this included why and where it is used and how to apply it, with reference to the SysML and the MBSE approach of this thesis. They were then asked to take one of the higher-level requirements they had identified and decompose it into a number of requirements at an atomic level and if time allowed allocate those atomic requirements to particular parts of the system architecture and demonstrate the requirement was fulfilled.

The purpose of this part of the exercise was to get the students to think about the functions that a System was required to perform and how they might be broken down and allocated to various parts of the System architecture and how an MBSE approach can help in managing the process.

Following this each group gave a presentation of their results and their view of the approach to the rest of class and the lecturers. In general, the feedback from the students was very positive and a large number of them seemed to have grasped the essential elements of the process and were able to apply it, to varying degrees. There were some areas where some students noted they had struggled to understand or see the point, these are detailed below in their evaluations. The course lecturers were also positive and intend to develop the group activity and use it again in the Systems Engineering unit of the course.

6.5 Final questionnaire

6.5.1 Purpose

The marked part of the group activity was to complete a questionnaire on the application of the MBSE approach. This asked the students to give their opinions on its application and usefulness. This questionnaire has two functions, the first was to get industry practitioner evaluation of the MBSE approach for this research and the second was to gauge how well the students had understood the ideas that underpin the use of MBSE as an approach to Systems Engineering. The results of this are detailed in the following sections.

6.5.2 Context Diagram evaluation

Students were asked to rate the use of the Context Diagram introduced in the first part of the exercise. They were asked to rate the following 3 questions from 1 to 5. 1 being not at all, up to 5 being very much:

1. Did using the context diagram give you a clearer understanding of how the system fitted into its environment and the other systems it needed to interface with?

2. Did developing the context diagram help you identify Stakeholders that may not have otherwise been identified?
3. Did using the context diagram guide you to understanding where the system boundaries were?

For question 1, the answers are shown Table 9 below.

Rating	Number of Students
5	10
4	18
3	6
2	0
1	0
No Answer	2

Table 9 Did using the context diagram give you a clearer understanding of how the system fitted into its environment and the other systems it needed to interface with?

Out of the 34 students that gave an answer to question 1, it can be seen that 28 of the students thought that the context diagram helped their understanding of where the System fitted in with its environment and other systems, while 6 rated the statement in the middle i.e. it neither helped or didn't help.

For question 2, 28 of the students gave either the top or second top rating. Again 6 students rated the question with a 3, neither helping or not helping. Interestingly 3 of the same students rated the first question with a 3 also.

Rating	Number of Students
5	13
4	15
3	6
2	0
1	0
No Answer	2

Table 10 Did developing the context diagram helped you identify Stakeholders that may not have otherwise been identified

For the final question of this section, question 3, slightly less than the previous 2 questions, 24 of the students gave a rating of 4 or 5, however in this instance 9 gave the middle rating of, which can be interpreted as neither helping or not helping. Also 1 student, who is a full time student, didn't think it was particularly helpful in this instance.

Rating	Number of Students
5	9
4	15
3	9
2	1
1	0
No Answer	2

Table 11 Did using the context diagram guide you to understanding where the system boundaries were?

From the answers to later questions it is known that 9 of the students had used a context diagram in their place of work, of these 7 rated their use 4 or 5.

From the answers to all three questions it is clear that the majority of the students found this part of the process useful for the Systems Engineering tasks it was designed for. As only 9 of the students had used a context diagram in their place of work, familiarity is not thought to be a major factor. Of the number of students giving one of the top 2 ratings, for each question

(28, 28 and 24) only 7 had previously used context diagrams before, about a quarter. In general, it can be concluded that this part of the approach was considered helpful to most of the students whether they had previous experience of its use or not. This conclusion is also in line with verbal feedback gained from the students during the exercise and presentation.

6.5.3 Evaluation of the approach to requirements

As with the previous section, the students were asked to rate questions about the MBSE approach to requirements elicitation and the development of views and viewpoints from 1 to 5, with five being ‘very much’ and one being ‘not at all’. The Students were asked the following:

1. Did the approach to requirements elicitation provide a structured approach to understanding and then finding requirements?
2. Did you find that using the approach helped you ask concise and meaningful questions of Stakeholders?
3. Did developing views and viewpoints for Stakeholders assist understanding what a Stakeholder needed from the system and how it could be delivered?

The answers to question 1 are given below in Table 12. In this case 29 of the students rated this statement with a 4 (19) or 5 (5). This clearly indicates that the students found this type of structured approach to requirements was beneficial.

Rating	Number of Students
5	10
4	19
3	4
2	2
1	0
No Answer	1

Table 12 Did the approach to requirements elicitation provide a structured approach to understanding and then finding requirements

Of the 2 students that rated the statement 2, one rated the previous set of questions on the context diagram 4, 4 and 3, while the other had rated them with 3s. Clearly the later student did not find any of the approaches used so far particularly helpful. Those that rated the statement with 3 were a more diverse sample, 1 had not provided answers to the previous statement, two rated them with two 4s and a 3 and one had rated them with two 4s and a 2.

The answers to question 2 are given below in Table 13. As with the previous statement It can be clearly be seen that the majority of the students, 25 answered with the top ratings. In all but one case students that had rated the previous statement highly also rated this question highly. However, not all of those 25 students gave the top two ratings to the context diagram questions, as might have been expected, 6 students had rated at least one of the previous statements as 3.

Rating	Number of Students
5	10
4	15
3	7
2	3
1	0
No Answer	1

Table 13 Did you find that using the approach helped you ask concise and meaningful questions of Stakeholders?

Of the two students rated the statement as 2, one who rated the previous statement in this section as 3 and did not give any replies to the previous section on context diagrams. Another also rated the previous statement as 2 but rated the context diagram statements 4, 4 and 3. The remaining student rated the previous statement 2 and also rated the context diagram statements with 3.

The answers to question 3 are given below in Table 14, as with the previous two statements in this section the majority of the students rated the statement 4 (17) or 5 (9). In one case a student that rated both the previous statements in this section 2 rated this statement 4. 3 students that used a rating of 4 or 5 had rated one of the previous statements as 3. In all other cases students that rated this question highly had rated all of the previous statements in this section highly.

Rating	Number of Students
5	9
4	17
3	5
2	4
1	0
No Answer	1

Table 14 Did developing views and viewpoints for Stakeholders assist understanding what a Stakeholder needed from the system and how it could be delivered?

In terms of the approach to requirements elicitation, clearly the majority of the students thought this kind approach was helpful in general. Further the majority felt that the structure enabled them to ask better questions during the requirements elicitation process and to understand their stakeholders better. Also given that there is some divergence in people that gave the top ratings between the context diagram statements and these requirements statements it can be assumed that the students are not just giving high ratings to please the marker but thinking about the statements carefully.

6.5.4 Activity modelling evaluation

This part of the process was focused on understanding how the System was to function and be used and therefore identify further functional requirements based on that understanding. The students were given one statement and one question with a yes/no answer:

1. Did activity modelling help you understand what functions the system needed to supply to deliver its purpose? and
2. Did the activity modelling identify more requirements?

Question 1 asked the students to rate the statement from 1 to 5 as with the previous questions, the number of students for each rating is given below in Table 15.

Rating	Number of Students
5	15
4	13
3	2
2	4
1	0
No Answer	2

Table 15 Did activity modelling help you understand what functions the system needed to supply to deliver its purpose?

The majority of the students thought that using this technique helped them better understand what their System was required to do, 15 rating it 5 and 13 rating it 4.

Question 2 required the students to give a simple yes or no response, because it was designed to understand if the application of activity modelling did actually help identify functional requirements. 28 of the students stated that this was the case and only 7 stated that it was not.

It can be clearly seen from these results that the majority found activity modelling useful for the purposes it is designed for and that in most cases they identified further functional requirements.

6.5.5 Overall approach evaluation

The remainder of the questionnaire focused on future usage of the approach and what the students thought were the least and most useful elements of the approach for them.

As with the previous sections the students were asked to rate the following two questions from 1 to 5:

1. Would you use this approach in a complex project if you had the opportunity?
2. Would you recommend the approach?

These questions were designed to find out if the students thought they might actually use a process, such as the one proposed, in their workplace now or in the future. It was also to understand how useable the students found the approach, clearly if they did not want to use it or recommend it, then it is not likely to be adopted. How the students rated question 1 is given below in Table 16. Here again 25 of the students rated the statement with either a 5 (10) or a 4 (15). This represents nearly 70% of the students would, given the opportunity use this approach in a complex project.

Rating	Number of Students
5	10
4	15
3	10
2	0
1	0
No Answer	1

Table 16 Would you use this approach in a complex project if you had the opportunity

The number of students that used the middle rating (3) is higher for this than any others 10, but no students rated the statement at the more negative end of the rating (1 or 2).

When asked if they would recommend this approach the students answered as detailed below in Table 17. As can be seen the numbers are very similar to those of the previous statement.

Rating	Number of Students
5	12
4	13
3	10
2	0
1	0
No Answer	1

Table 17 Would you recommend the approach?

It is clear from both sets of answers that the majority of students thought that the MBSE approach was useful enough for them to want to apply it to complex projects and that they would recommend it to others. Also none of students used the lower ratings of 1 or 2, which indicates that it was not found to be obstructive or too difficult to use.

In order to get a useful evaluation of the whole approach it was necessary to drill down into the overall ratings. For this reason, the students were then asked to hi-light which of the 6 parts of the process (context diagram, requirements elicitation, views and viewpoints, use cases and activity modelling and functional decomposition) that they used were most useful. The majority of the students answered, identifying two or more parts that they found useful. The number of students that chose each part of the process are given below in Table 18. One student gave no answers.

MBSE Element	Number of Students
Activity Diagram	20
Context Diagram	22
Functional Decomposition	4
Requirements	22
Stakeholder views and viewpoints	7
Use cases	10
Whole approach	12

Table 18 which parts of the process were most useful

The students were asked to give reasons for their choices. As with previous questions involving plain language the author has grouped them into common words and phrases to allow analysis and identify more succinctly what was really useful and why.

A significant number of students (12) identified the whole approach as useful. Although this not particularly useful in terms of analysing the individual elements of the approach some of the reasons for this response are reproduced below for completeness:

- MBSE activity enabled me to correctly identify stakeholders, capture and perfect requirements. It enhanced communication, improved understanding, reduced complexity and saved time;
- MBSE creates an easier knowledge and information sharing platform which can be easily added to;
- Using model based systems design is better than the traditional approaches because model based system design creates form and structure;
- The approach is very useful for identifying all key stakeholders, their requirements, assists with identifying boundaries and breaking down stakeholder decisions. It is a good method of recording the information for future manipulation;

- Each of the processes provided a well-guided approach to understand the requirements of a stakeholder's view and flowed easily between each one;
- This team exercise is very helpful because there are several processes which make students understand model based system engineering of railway operations in terms of stakeholder's requirements;
- In general, all the steps from stakeholder interviews to functional decomposition were found useful as outputs from each step informed the next.

In the case of the context diagram, one of the two most common elements identified by the students as useful (22), the reasons given are summarised as, the context diagram:

- identifies a comprehensive list of Stakeholders;
- can be easily understood by many different Stakeholders;
- defines System boundaries;
- identifies interfaces;
- gave an easy to understand visual view of the System within its environment;
- identified Stakeholders and Systems that would not necessarily have been identified if this step was not used;
- provides a foundation for identifying System requirements;
- inputs to the whole SE process are invaluable.

The same number of students (22) identified the approach taken to requirements as particularly useful. It should be noted that this was not exactly the same set of students that identified the context diagram as above, although some did identify both. The reasons given are summarised below as, the approach to requirements:

- reduces the risk of missing requirements;

- captures and perfects requirements;
- enhances communication with Stakeholders;
- identifies requirements that are vital to the project;
- is applicable to large projects;
- helps manage requirements throughout the System building process;
- drives out issues between differing requirements;
- can clarify the scope of work;
- provides an easy to understand visual description of requirements and their interactions.

The next most popular element identified by 20 students was the activity diagram. The reasons given are summarised below as, the activity diagram:

- shows the user all the possibilities via a loop;
- helps assess deliverability;
- lowers the risk of missing Stakeholders and requirements;
- allows effective fleshing out of requirements;
- allows the developers to walk through every step of a process;
- helps understand the sequence of events;
- supports effective planning and operation.

In terms of use cases 10 students identified these as particularly useful, their reasons for doing so are summarised as, use cases:

- lower the risk of missing Stakeholders and requirements;
- are suitable for large projects;
- clarify requirements;

- are a simple method of looking at the System;
- can assist with identifying/clarifying System boundaries;
- describe the subsystems that feed into activity modelling;
- help clarify related system elements that should be considered;
- assist modelling functions required by Stakeholders;
- can help manage complexity;
- are user friendly;
- are a well-supported and understood tool.

One of the least selected elements is Stakeholder views and viewpoints, this element was selected by 7 students. The reasons for their selection are summarised below as, Stakeholder views and viewpoints:

- are very useful for larger projects with numerous Stakeholders;
- clarify Stakeholder requirements and the scope of work;
- specify Stakeholder concerns related to the System model;
- help refine model requirements.

The element that was selected by the smallest number of students (4) is functional decomposition. The reasons given by the students who did select this element are given as, function decomposition:

- is a useful output from one step to the next;
- ensures that components are able to perform all necessary roles;
- provides an insight into whether a requirement can be delivered;
- shows how a complex system can be broken down into its constituent parts;
- can be helpful for developing testing;

- assists in understanding a complex System;
- is helpful in producing design and functionality specifications for the various components of the system;
- helps with identification of components that are able to deliver more than one function.

The author also wished to understand if there was any discipline or industry bias associated the choices made by the students. In terms of the spread of disciplines and industries in the whole group:

11 students from Railway infrastructure = 30.6%

5 students from Railway rolling Stock = 13.9%

5 students from Railway signalling = 13.9%

1 student from Railway communications = 2.8%

1 student from Railway planning = 2.8%

1 student from Railway power OLE = 2.8%

7 full time students = 19.4%

2 Civil Engineers = 5.6%

1 student from the oil and gas industry = 2.8%

1 student from the automobile industry = 2.8%

1 student from real estate = 2.8%

The distribution of industry and discipline of the 22 students that found context diagrams particular useful is given below in the graph Figure 54. The only non-Railway discipline that appears is students. The Railway discipline that does not appear is Power/OLE. A significant number of the total number from Railway infrastructure 9 from 11, Railway signalling 3 from 5 and Railway rolling stock 3 from 5 made this choice.

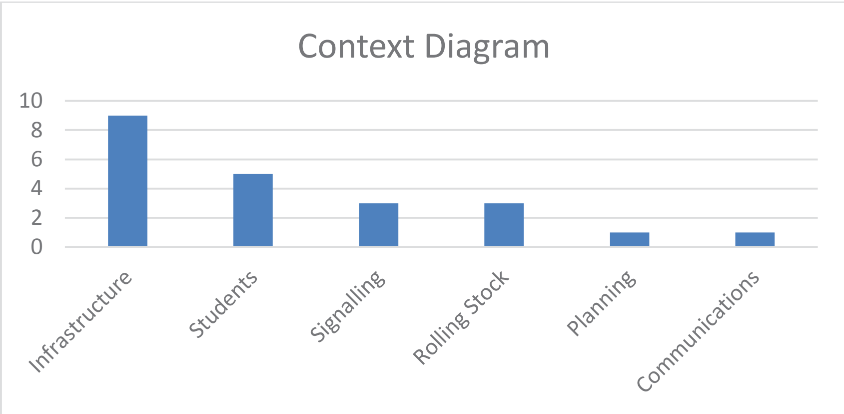

Figure 54 Context diagram (author)

The industry and discipline distribution of the 22 students that chose the approach to gathering and representing requirements through interviewing the identified Stakeholders and capturing those requirements in a requirements diagram is shown below in Figure 55. Again the largest representation is from Railway infrastructure, which as they are the largest body in the sample group is not a surprise. The most striking difference from the distribution for the context diagram is the appearance of the non-Railway disciplines, both civil engineers, the student from the automotive industry and 5 of the students. Communications and power OLE are not represented from the railway disciplines.

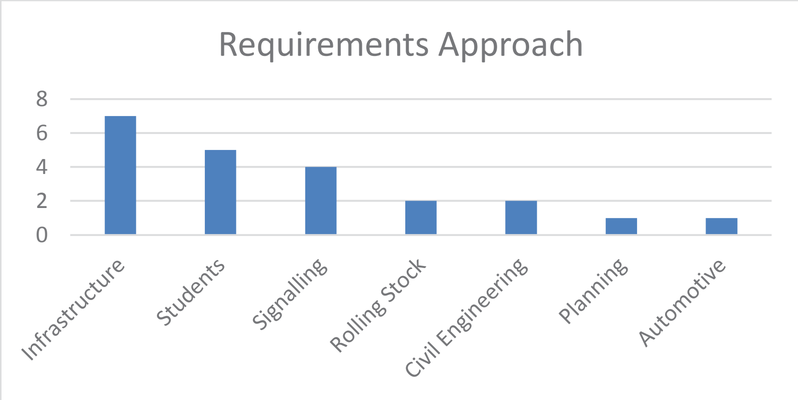

Figure 55 Requirements Approach (author)

Of the 20 students that chose activity diagrams as one of the elements they found most useful, the distribution of disciplines is wider as shown below in Figure 56, 8 out of the total number of 11 disciplines are represented. As with the approach to requirements both non-Railway civil engineers are represented as are the full time students and the student from the Automotive industry. In terms of Railway disciplines Railway infrastructure, signalling and rolling stock are represented. Although the number of infrastructure students is down.

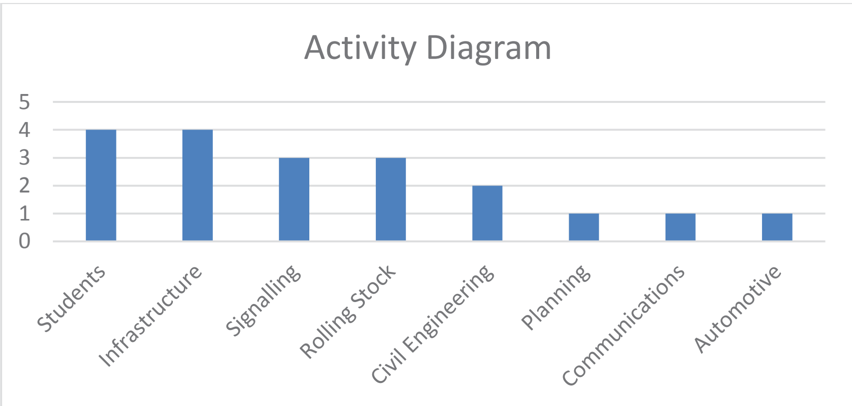

Figure 56 Activity diagram (author)

Use cases were a much less popular choice of particularly useful elements of the approach 10 students chose them. The distribution across the disciplines is below in Figure 57. As there are so few it is difficult to draw any strong conclusions from this distribution other than as an element use cases were not popular with the sample group in general.

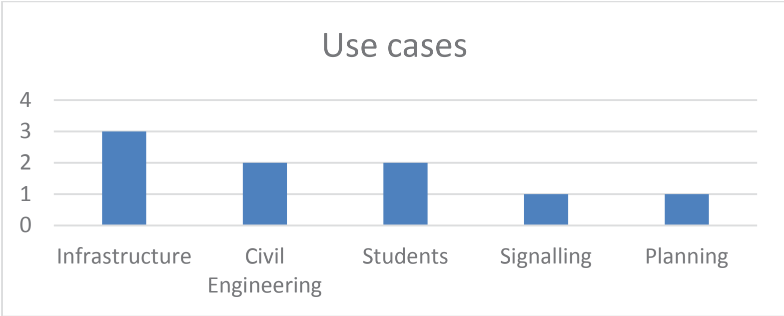

Figure 57 Use cases (author)

Stakeholder views and viewpoints were less popular than use cases, only 7 students thought that they were particularly useful as follows, 1 Student, 1 signalling engineering, 2 rolling stock engineers, 1 infrastructure engineer, 1 communications engineer and 1 civil engineer. This was not a surprise, as this during the group activity this was the element that students had the most trouble understanding. This was mostly to do with the fact it was modelling what a Stakeholder requires from the System model rather than the system itself.

Only 4 students thought that functional decomposition was particularly useful. One each from the disciplines of Railway signalling, rolling stock and infrastructure and one Civil engineer not from the Railway industry. This probably has more to do with the fact this was the very last thing that was dealt with in the group activity and the students didn't really have time to treat it in any depth, so the application wasn't really appreciated.

There was not any hard evidence to show a bias across disciplines within the Railway industry and only very slight evidence to show difference between those in Railway industry and those from other industries. It seems that people from the Railway industry found elements such as the context diagram and activity modelling more useful, but this could easily be explained by the fact the sample group contained more people from the Railway Industry than without.

Two students chose only one element of the process as being particularly useful; from this it could be assumed that they did not really like the approach. One of these is a Civil Engineer in the real estate industry, her/his comments are summarised as:

> *"Regarding system engineering, there are several approaches which could be developed and applied to improve work processes. Different approaches provide lots of advantages effectively based on objectives and scope of work. This is due to system engineering contributing the big picture thinking or considering the whole problem, system and lifecycle of a project. Based on the group assignment, in my opinion, the most useful approach for system engineering is the activity diagram. This approach supports the user to comprehensively and clearly understand the flow of activities and control in a project. The activity diagram illustrates various key processes and functions which can be linked together in series or parallel. A project manager and system engineer can effectively develop this diagram to represent not only particular activities in the system but also the project process. The flow of the activity diagram extensively describes various actions of the main actor and other related Stakeholders in the system. Thus, model users can perceive the required functions and activities which occurred in a concerned aspect step by step. Nonetheless, reading the activity diagram requires some basic knowledge and understanding which could cause a problem for some users".*

On first inspection this looks as if this particular student thought that only the activity diagram was of any use. However, when looking at the answer the same student provided when asked the question which parts of the process did you not find useful and why the answer was:

> *"Several modelling tools provide a large number of advantages to Railway system engineering. However, some of them require the in-depth understanding which can lead to difficulty of operation. Some models do not provide enough information and comprehensive procedure to achieve all project requirements. In my opinion, the use case diagram provided roughly the dynamic or behaviour interactions between various actors and series of activities which can lead to the deficiencies of systematic analysis. The use case diagram and specification can be conducted in the forms of context and chart. Normally, the approach consists of four primary factors namely system boundary, use cases, actors and relationships between an actor and a use case. The actors contribute the main Stakeholders who participate with the system and while the use case is the function of system performed by an actor. Obviously, the diagram only represents system's Stakeholders, activities and the interaction between these two factors. Hence, the diagram only supports users to consider towards the functional requirements, external and internal determinants which can affect the system. System engineering does not acquire other important information such as work process, work sequences, relationships of activities and detail from the use case diagram. This model tends to be less flexible and iterative method due to the diagram only show interaction between actors and use cases. Implementation of the use case diagram for Railway projects may not provide the sufficient important information. Railway developers need to realise model constrain and perform other approaches together with the use case diagram".*

It is clear from these two answers that this student compared the two parts of the dynamic modelling part and didn't not feel that both were needed. This position was reinforced when the student answered:

> *"My previous work relates to the real estate industry. The activity diagram can effectively support the project development for real estate construction since the property development process represents similarly to Railway projects in some aspects such as various Stakeholders, planning and construction process. There are numerous of activities and components that require appropriate management. Likewise, a project manager and system engineer have to deal with project complexity and complication of a property project. Property developers could also understand and picture comprehensive work sequence, function requirements and the explicit interactions between Stakeholders and system activities of the project. This approach can ensure precise requirements of customers and Stakeholders".*

This student's perspective was biased towards project delivery rather than Systems Engineering. The author has come across this blurring of project management and Systems Engineering previously, this is mostly due to the positive effect good Systems Engineering can have on project outcomes, hence Network Rail's call for more and better Systems Engineering and thinking (Network Rail, 2013).

The other student that chose only one Element of the process is a full time student, he/she chose the context diagram and stated:

> *"I found that the context diagram is useful. Firstly, the context diagram is simple to draw and understand. All the blocks representing Stakeholders and environment are linked to the system, showing the interactions between the system and these external forces. It helps me gain a better view on the inputs from external factors to the system and the outputs from the system to external factors. Secondly, using the templet context diagram helped the capture of every external factor including every*

Stakeholder and the environment where the system runs. So that I will not miss any Stakeholders and, therefore, not miss their requirements when doing requirements elicitation. Lastly, the context diagram helps with the understanding of system boundary. It shows what functions are delivered by the system and more importantly what functions are not delivered by the system. In conclusion, I think the context diagram is quite useful at the beginning of system design process, as it helps the designers gain a better understanding on the interaction between external factors and the system and the system boundary as well".

When asked which parts of the process that were not useful this student stated:

"I think that the Stakeholders' view and viewpoint is not so useful to me. Actually I am still confused with the view and viewpoint diagram. I think it would be better if it can be explained in detail with some simple examples".

It seems that both of these students compared the elements of the process that they understood best with the elements they either did not find useful or found least useful, although the view and viewpoint elements and use cases were among the least popular overall. In hindsight the questions could be interpreted as asking for this. However, when answering the questions, would you use this approach in a complex project if you had the opportunity and would you recommend this approach, both the students gave ratings of 4 and 4 and 4 and 5 respectively. So the conclusion that they did not find the process useful cannot really be drawn from these statements.

For a complete evaluation students were also asked which elements of the approach they found least useful and why. The elements that the students considered least useful and the number of students that chose a particular element are shown in the graph Figure 58.

Use cases were identified by 14 students as the least useful element of process, the reasons they gave for this choice are summarised as, use cases:

- are over simplistic;
- summarise some of the relationships between use cases, actors and Systems;
- lack detail in terms of sequence of implementation;
- are confusing;
- are tedious;
- can be overlapped with other modelling in places;
- can become overly complex;
- Do not show implementation only interaction.

The next most unpopular element of the process was Stakeholder views and viewpoints, 13 students identified this element. This is not surprising as this was an area that the students struggled with during the activity and a number also hi-lighted this in their end of unit presentation. The reasons they gave for their selection can be summarised as, Stakeholder views and viewpoints:

- are confusing, unclear;
- are a lot of work for not very gain;
- appeared to overlap with other elements of the approach;
- not fully understood (but just looking for an answer to this question);

The remaining elements that were selected as not particularly useful were selected by relatively few students but are included here for completeness along with a summary of the reasons given.

6 students selected the context diagram, the reasons given were, the context diagram:

- can be complex or get complex quickly;
- is very general due to the sheer number of Stakeholders;

4 students selected the approach to requirements elicitation, their reasons were as follows, the requirements diagram:

- did not seem particularly useful;
- is rather general;

The activity diagram was selected by a single student, the reason the student gave was, the activity diagram is very complex for complicated activities and too complicated for trivial activities.

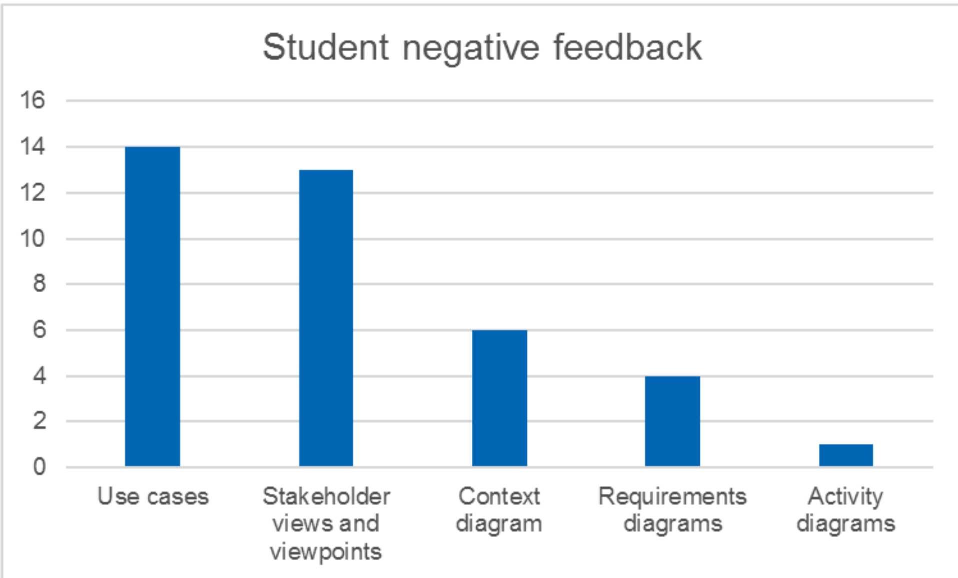

Figure 58 Student negative feedback (author)

In amongst all of the feedback in the open questions there was some negative feedback about the process as whole. Although some of this was directed at the delivery of the activity material and the choices made with regard to what the activity would concentrate on. This feedback is summarised in the bullets below:

- Not sure of how applicable MBSE is in the workplace;
- Did not really understand the process;
- Large overhead in terms of tools and training;
- Would have liked more on UML and SysML;
- There was a lot of confusion in the delivery of the Group activity material which led to some confusion amongst the students;
- Not coherent without the modelling tools such as SysML;
- The exercise was too long to be applied within the time allowed;
- Not enough time;
- Real Stakeholders would have been better than students and staff acting the part.

6.6 Summary of results

When asked to define what a System was nearly 80% of the students used key words and phrases that identified a System as a number of “things” (processes, people and technology) working together. This kind of answer was expected. However, less than half the Students mentioned that a System existed to achieve a purpose. Very few mentioned that a System has a boundary or that it exists in an environment. There was also some confusion between System and Systems Engineering. What was clear, was that after the idea that a System was things working together, the student’s definitions varied considerably, which showed that there was clear divergence of opinion on what a System actually was. This became more apparent when the Students were asked to describe Systems Engineering, a third of the students believed that it was about the management of the design of a System. A little less than quarter of the students mentioned anything to do with interfaces. There was very little agreement after these themes. This underlines the assertions, that there are many ideas on what a System is and even more regarding what Systems Engineering is.

The majority of the students thought they were involved in Systems Engineering half or more of their time. Looking at the application of Systems Engineering tools and processes across the sample of students it was apparent that this was not the case, only 10 of the students could point to a documented/Formal Systems engineering process at their place of work, all 10 were able to supply a description of the process. When asked about specific Systems Engineering tools and practices that they used at work 10 of the students had used a SEMP, 8 an IMP, 8 a Context Diagram and one used company specific tools.

The conclusion drawn from this is that Systems Engineering is applied sporadically across the industry and definitely not uniformly, even though Standards exist to guide practitioners.

In general, the proposed MBSE approach developed in chapter 4 was well received and supported by the students that took part. In all cases where the students were asked to rate statements about the process or particular elements of the process, the majority of the students gave one of the top two ratings (4 or 5). The lowest rating (1) was never used and a rating of 2 was only used by a minority of students, even with respect to the most unpopular elements of the process, Stakeholder views and viewpoints and Use cases. From feedback gathered from the students while they were carrying out the activity, their presentations and their answers to the final 4 questions in questionnaire 2, it was clear that Stakeholder views and viewpoints and to some extent use cases were the elements that were least understood and applied. At least some of the reason for this is due to the delivery of the material (by the author), the short period of time allowed to teach the concepts and the fact that the diagrams were created with standard drawing tools rather within an MBSE modelling language such as SysML/UML which would have hi-lighted the benefits of the central model.

Even with above short comings in the group activity, the majority of students said that they would use the process and recommend it.

Based on the analysis of the data supplied by the students on the course it is clear that this type of approach is new to the majority them and that it would be a welcome technique to be used, particularly on complex projects.

Surprisingly there was very little difference of opinion between those who held more senior positions with those who held more junior positions. For example, when asked the question, "*would you use the approach in a complex project if you had the opportunity?*", 64% of those that held management type roles rated the statement 4 or 5 and 70% of those with more junior roles gave the same rating. There was slightly more of a difference with answers to the question, "*would you recommend the approach*", 62% of managers said they would and 75% of those in more junior roles said they would. In general, it can be concluded that for both sets of people the majority were positive towards the approach.

There were some concerns, quite rightly, around the cost of the modelling tools and training overhead preventing take up of this type of approach.

7 Discussion

7.1 Introduction

This section discusses the authors findings as a result of the research undertaken, modelling work carried out and its evaluation by a representative body of practitioners. It consists of six sections:

1. Introduction, this section, consists of a description of what this chapter contains;
2. Research objectives, provides a discussion on the objects set at the beginning of the project and how they changed as the project progressed;
3. Literature review, contains an overview of the literature review undertaken and the conclusions drawn from it;
4. Suggested modelling approach, contains a discussion on the modelling approach taken, what it was able to cover and lessons learned;
5. The feedback from the evaluation of the modelling methodology from a representative group of practicing engineers; and
6. Results, this sections discusses the level to which the objectives of the project were met and what is left outstanding.

7.2 Research objectives

The focus of this research project and thesis was to look into the possibility of producing a general methodology to provide a general approach to systems engineering and modelling in the GTS industry. The representation/model of a general GTS can then be applied to assist the industry in building general multi-view, multi-purpose, multi-aspect models of GTS's as a whole and therefore aid the application of systems engineering and thinking across the industry. If adopted by a wide stakeholder group, the methodology will drive a consistent systems engineering approach across companies and indeed the industry and thus address some of the barriers to its application hi-lighted within the literature (Beasley, 2012), (Davidz, Rhodes, & Nightingale, 2005) and (Dunford, Yearworth, York, & Godfrey, 2012). This author wanted the methodology developed to be able to use available data from existing industry recognised sources, models and simulators as well as bespoke models and simulators from all areas and bring them together. Also, in order to drive consistency and breakdown some of the logistical and resistance to change barriers hi-lighted by Beasley (2012), Dunford et al (2012) and Pickard (2010), the approach was to be able to represent a GTS from a number of stakeholder viewpoints such as:

- Shareholders (financial predictions, spending, revenues etc.);
- Operators (performance, capacity, revenues, options and energy consumption);
- Maintainers (effects of maintenance strategies, component usage, forward costs predictions); and
- Passengers/Customers.

Parsley et al (2013) demonstrated that guidance, particularly of less experienced engineers significantly affects the successful application of systems engineering and enables the benefits of their application to be more readily realised. Therefore, the approach was also to act as a guide to modellers on how to construct actual models, e.g. have a general view on what a GTS is, what subsystems it should contain and the likely relationships between them, at its simplest level, based on some initial information, which can then be specialised and populated.

To guide this a number of requirements were developed as follows:

1. The modelling methodology and models themselves shall be re-usable, e.g. applicable to different GTS' at different times. This will allow the approach to more

easily be introduced across departments, companies etc and drive the common approach and understanding needed for successful application of systems engineering. Being reusable also reduces application cost and time overheads, another barrier that was hi-lighted in the literature;

2. Both the methodology and the models themselves shall be extendable in length (be able to model more railway) and depth (greater levels of detail) in order to make them relevant as new systems become available and legacy systems expand. This also reinforces the reusability of the models and helps to manage the logistical, cost and time issues identified in the literature;
3. Where possible the methodology shall allow the inclusion of existing quantitative and qualitative models from other sources, to allow complete descriptions to be formed within the models themselves. There are two advantages to this, firstly it promotes reuse of existing models making sure engineers see that their work is still valuable, which helps deal with resistance to change and secondly provides cost and time savings through less repetition;
4. The methodology shall allow and encourage the use of data from existing models to avoid re-inventing already existing and adequate models;
5. The methodology and the models themselves shall be sufficiently open and transparent to allow others to use and add to them to encourage users to “buy in” to the approach and feel ownership through being part of the development process; and

6. The methodology shall encourage and the models themselves shall enable the production of outputs that are readily understandable across disciplinary divides e.g. common representation. Literature review

During the course of the literature survey this author was not able to find a genuine, non-biased whole system GTS modelling approach or methodology. There are a number of approaches that model whole systems for particular purposes, but none that exist to just produce a generic model for general application. Also there are no guides into how one might go about producing such a model. The examples given earlier demonstrate some of the versatility of the UML and SysML modelling languages and the overall object oriented approach to system modelling. They are versatile in terms of how they can be applied, modellers can use all or a subset of the diagrams in the creation of their systems model. Which is useful in terms of managing the perception of how onerous the systems engineering overhead is particularly on smaller projects identified as barrier to the application of Systems Engineering by Beasley (2012), Dunford et al (2012) and others. They have the advantage of being able to capture both structure and behaviour/functionality as demonstrated particularly in Ferrogalini (2015). It also works well with other approaches, as shown in the Digital Railway EA (Umiliacchi, et al, 2018) where UML is used to represent the underlying ontology method/meta-model. SysML is a particularly useful language as it can capture mathematical modelling and requirements. However, although it can model dynamic behaviour in terms of representing it with activity diagrams, state machines and sequence diagrams, it cannot on its own simulate that behaviour. It therefore needs to work with other tools such Matlab Simulink etc. UML particularly and to a certain extent SysML lack rigor particularly in terms of mathematics and logic in their language. This makes them less useful in verification and validation work against standards and specifications on their own.

Although in SysML at least this has been addressed to some degree by parametrics (Peak, et al, 2007 and Ferrogalini, 2015). The above examples show this is overcome in all instances by the use of other formal methods such as the B method (Bousse, et al, 2012), or Petri nets, (Sana, et al, 2010) alongside UML and SysML. Also the examples have demonstrated how SysML in particular can be made to model complete systems, with many differing views. Another advantage of using SysML or UML is the large body of literature, tools and training that are commercially available.

Although there will always be the need for the use of other models, databases and tools SysML seems to provide the central underlying model that can be used within some formal approach to tie all of the others together.

On their own SysML and UML are just modelling tools and to be really useful in modelling large and complex systems they need to be part of a larger approach, such as MBSE or EA, (Delligatti, 2014, p. 5).

Research on the use of ontologies in the railway industry is ongoing at the time of completing this thesis (2019). The main thrust is still towards system and data interoperability and the advantages that can be gained from improvements in the way information systems are used within the industry (Tutcher, et al, 2017). The Railway Domain Ontology is a significant contribution as it provides base ontologies for others to work from (Moris, et al, 2015), although Umiliacchi et al (2018) query its readiness for more general use. There is also ongoing research into how ontologies can tailor information for specific passengers (Gould, et al, 2017).

Research in this area is being driven by continued sponsorship of projects by government or government backed organisations such as the EUs ST4RT project. This research project is on the subject of sematic ontology based automation of transformations between hetrogeneous data formats. It is part of the Shift2Rail interoperability framework (Shift2Rail, 2018).

Undoubtedly all of the major contributions from ontologies are underpinned by some form of system model. Although not an Ontology, RailML is nearest to a whole systems model and is expanding (Nash, et al, 2010) and ontologies such as RaCoON use it. In the majority of cases the modelling has been done to achieve particular purposes, such as data interoperability, ERTMS requirements and test specifications etc. and their expansion has been driven by the need to solve these issues.

So far a complete whole railway system ontology model has not yet been completed. However, there seems to be enough evidence to suggest that using ontology modelling, as part of an integrated approach with other models it could be achieved. The use of other models with it would be particularly useful to give a graphically based easier to understand representation. The Digital Rail project is working on this aspect (Umiliacchi, et al, 2018).

As far as the railway industry is concerned EA is a very new approach. Although it does have a lot in common with UML/SysML type approaches. Its chief advantages are that it is an integrated approach much like MBSE and the demonstrations describe the whole railway from differing viewpoints (particularly as it is business centric). This is of particular interest to this project. At the time of writing this thesis EA is still very much a business/ICT based approach. This can make some of the language and terms used a little difficult for those who do not have that background. This could be a barrier to commercial success, as decision makers could be put off before they understand the advantages. Most of the demonstrations underline this. It is very likely that as the user group gets larger and more diverse this will change. This point is made by the three demonstrations above, TRAK, RFA and Digital Railway. They show how EA is being adapted to the railway industry over time as practitioners become more skilled in its use and the industry requirements become clearer. Using UML or SysML to represent EAs, as in Figure 26, Figure 27 and Figure 28 enlarges the group of people who can easily adapt to its use. This is a clear advantage as lack of skills is identified in the literature as a barrier to the application of systems engineering within companies (Dunford, Yearworth, York, & Godfrey, 2012) and (Davidz, Rhodes, & Nightingale, 2005). None of the EA examples found demonstrate an attempt to actually model a whole railway system, although all state such a model is their aim. All of them are also biased towards process or IT type systems, they seem to be missing a guide to what a railway is or what the railway needs to be. They are developed for specific issues and therefore reflect just the particular railway or part of the railway affected.

Given the above the author believes that the methodology developed is novel to the railway industry at this moment in time.

7.3 Approach taken

Given the complexity of GTS’ and the results of the literature survey detailed above it was decided to develop a GTS model that meets the project requirements by using an implementation approach based on MBSE (to handle the complexity issues), realised in SysML, taking in some ideas from enterprise modelling (such as looking at the GTS as an enterprise and views and viewpoints), leading to interaction with other approaches for more specialised modelling including bespoke mathematical models, Ontology representations etc. These were not however developed in this thesis for reasons of space and time.

The MBSE approach decided on is shown in Figure 29. It is a 5 stage process:

- Stage 1, is the model set up stage, this is where the 'what, when and who' are identified at the highest level, for example who the primary Stakeholders are, what sort of GTS will it be, for example will it be a mixed traffic railway, a tramway, a metro etc. Also where will the GTS be situated. The key output from this stage is high level Requirements and a list of primary Stakeholders.
- Stage 2, model organisation, this stage, from a purely modelling point of view, is vital. It details the construction of the model, how easy/difficult it will be to understand, navigate around, control (configuration, relationships, behaviours etc.) and reuse. Getting this right is important if wide spread use of the methodology is to be encouraged. The output from this stage is a SysML artefact called a Package Diagram which represents the model structure see Figure 30.
- Stage 3, Context Model, the context within which the GTS that is being modelled is defined, it identifies the systems and "things" that will interact with the GTS system. Also the GTS is identified as an Enterprise, see section 4.3.3. The Stakeholder list is also revisited to allow for new Stakeholders to be identified from this more detailed view of the system and its operational environment. The system model boundaries are identified at this stage as well. The outputs from this stage are a Block Definition Diagram called the Context diagram see Figure 32 and the views and viewpoints for the Stakeholders see chapter 4.3.4.
- Stage 4, Requirements, the new Stakeholders are identified and their Requirements recorded. Functional Requirements are derived from operational scenarios and critical system properties, design constraints and black box structure are driven from the identified Requirements. The Requirements process also starts the system decomposition process. The outputs from this stage are Use Cases, see Figure 37,

Operational representations in the form of behavioural diagrams see Figure 38 and Figure 39 and Requirements diagrams see Figure 40.

- Stage 5, system decomposition and validation. The system is decomposed layer by layer until the correct level of abstraction is achieved. By the correct levels it is meant that each of the Requirements has been met through a single atomic argument, which acts as validation. At each level of abstraction, the critical system properties are analysed to understand if a further level of abstraction is required to deliver the level information needed, see Figure 50 and Figure 51, for example looking at system reliability, there are choices, reliability can be obtained at a system, subsystem or component level depending on the data available and the requirements of the particular Stakeholder.

Although time and space have not allowed the development of a complete GTS model (also this author is not expert enough in all the many areas that a GTS encompasses), a representative set of SysML artefacts diverse enough to demonstrate the power of the MBSE approach has been developed.

7.4 Feedback from Evaluation

The methodology developed in this thesis was introduced to and used by a representative group of practicing rail biased engineers who were attending the University of Birmingham Railway Systems Engineering and Integration MSc course. The aims of the exercise were to have practicing rail biased engineers evaluate the methodology developed and give an opinion on its applicability and usability. It was also to understand what the engineers thought about systems and systems engineering, to gain an understanding of the application of systems engineering and thinking across industry. It was clear, that there was a large divergence in opinion across the group over what a system actually was and also the definition of systems engineering. This underlined earlier assertions, that there are many ideas on what a System is and even more regarding what Systems Engineering is. This leads to confusion and inconsistent application. These have been identified in the literature as major barriers to the application of systems engineering in any industry (Beasley, 2012), (Davidz, Rhodes, & Nightingale, 2005), (Dunford, Yearworth, York, & Godfrey, 2012) and (Elliott, 2014).

The majority of the students thought they were involved in Systems Engineering half or more of their time. Analysis of their feedback did not back this up. When asked about specific Systems Engineering tools and practices that they used at work 10 of the students had used a SEMP, 8 an IMP, 8 a Context Diagram and one used company specific tools.

The conclusion drawn from this is that Systems Engineering is applied sporadically across the industry and definitely not uniformly, even though Standards exist to guide practitioners. This evidence underlines that obtained from other industries as demonstrated by Beasley (2012), Dunsford et at (2012) and others.

In general, the proposed MBSE approach developed in chapter 4 was well received and supported by the students that took part. In all cases where the students were asked to rate statements about the process or particular elements of the process, the majority of the students gave one of the top two rating (4 or 5). The lowest rating (1) was never used and a rating of 2 was only used by a minority of students, even with respect to the most unpopular elements of the process, Stakeholder views and viewpoints and Use cases. From feedback gathered from the students while they were carrying out the activity, their presentations and their answers to the final 4 questions in questionnaire 2, it was clear that Stakeholder views and viewpoints and to some extent use cases were the elements that were least understood and applied.

The majority of students said that they would use the process and recommend it.

7.5 Level of Success

In terms of the requirements set out in chapter 3 of this thesis it is possible to say they have been met as follows:

- Requirement 1: The modelling approach and the models themselves shall be re-usable, e.g. are applicable to different GTS' at different times.
    - The MBSE modelling approach developed encourages reuse, the demonstration artefacts are of a generic nature, allowing them to be specialised and used by others to fit the Requirements of any particular GTS, an example of this is the specialisation of the RollingStock Block to an ElectricLocomotive Block. See Figure 47 and Figure 49.
    - In the evaluation of the approach by the sample of practicing engineers on the University of Birmingham MSc course, students demonstrated that they were able to specialise and adapt (effectively reuse) the generic model fragments supplied.
- Requirement 2: Both the approach and the models themselves shall be extendable both in length (be able to model more railway) and depth (greater levels of detail) in order to make them relevant as new systems become available and legacy systems expand.
    - SysML is designed to be extendable it gives the ability to design for abstraction, so structures and behaviours can be added to down through the

lower levels of abstraction as more information becomes available, or as new subsystems or components are added.

- Requirement 3: Where possible the approach shall allow the inclusion of existing quantitative and qualitative models from other sources, to allow complete descriptions to be formed within the models themselves.
  - The MBSE approach developed allows the use of data and inputs from other modelling, an example being that the particular SysML tool the author employed for this project can be linked to MatLab, Simulink, Telelogic DOORS and others. However, all these "live" outside the actual modelling environment. So technically the approach does not allow the inclusion of other models only the results thereof. This is demonstrated in chapter 4.3.4 where the Viewpoint specifies that one of the outputs should be in Microsoft Excel.
- Requirement 4: The approach shall allow and encourage the use of data from existing models to avoid re-inventing already existing and adequate models.
  - This Requirement is satisfied as described above.
- Requirement 5: The approach and the models themselves shall be sufficiently open and transparent to allow others to use and add to them.
  - SysML was chosen as the modelling language part of the MBSE approach precisely because it is an openly available language used and supported by major corporations and societies (OMG, INCOSE, Lockheed Martin, NASA to mention a few). Also SysML is developing a large body of instructional literature, industry applied examples and training, see literature survey.

- Requirement 6: The approach shall encourage and the models themselves shall enable the production of outputs that are readily understandable across disciplinary divides e.g. common representation.
    - The ability of SysML to use Views and Viewpoints enables the contents of the model to be output in a number of formats and styles etc., so information can be delivered to Stakeholders in a way they can understand, also it can be delivered in formats for other modellers to use, see section 4.3.4.
    - The whole process was evaluated by a representative body of practicing engineers from within the railway industry. In general, their feedback was positive, with most believing that the process was usable and valuable, see chapter 6. This is likely to go some way to addressing the problem of resistance to using systems engineering techniques and tools identified in a number of industries within the literature survey (Beasley, 2012), (Davidz, Rhodes, & Nightingale, 2005) and (Dunford, Yearworth, York, & Godfrey, 2012).

One of the chief advantages of the SysML approach is the amount of information that is available to researchers, in terms of text books, published papers and tutorials, something that was lacking with the ontological approach. Training and guidance was hi-lighted in the research as one of the barriers to implementing systems engineering and systems thinking (Beasley, 2012), (Parsley, York, Dunford, & Yearworth, 2013) and (Dunford, Yearworth, York, & Godfrey, 2012), therefore using a tool set that is in common usage across a number of industries goes some way to breaking down this particular barrier. The graphical nature of SysML in general makes it quite intuitive, although there are quite a number of diagram types that can go into developing a single model. The true advantage of this approach and its implementation in SysML is that once an initial model is developed it is reasonably easy to maintain and manage, therefore the initial costs of setting the model up, training etc. could recouped through particular applications of a generic model. This also reduces the effort going forward, which is another of the barriers to the application of systems engineering and systems thinking identified in the literature. SysML's ability to hold a representation of a whole GTS within a single model is an advantage in terms of understanding how changes to one part of the system can have an effect on other parts of the system, which again in terms of costs, can prove an advantage in terms of maintenance costs and for further applications of the same model. To make the approach successful in a commercial sense a concerted up front effort in tool selection and training would prove very useful.

Other barriers to the application of systems engineering identified in the literature such as logistical barriers are also reduced by the use of a common approach and industry recognised tool set.

However, it must be stated that this was a proof of concept project and there needs to be more specialised work done to demonstrate with complete confidence that this is a practical and usable way to go about whole railway system modelling going forward.

Although this work has gone some way to addressing some of the practical barriers to the application of systems engineering and its associated tools and techniques, there are still some significant issues to overcome. There is plenty of evidence that shows the positive benefits that can be expected from the application of systems engineering (see chapter Introduction1). However, it is usually associated with problems not occurring, for example projects not exceeding their timescales and budgets or delivering what was originally expected. These are very hard to quantify on new projects as they only become evident was things have gone wrong. So there is still work to do to convince key stakeholders, project directors and managers that any upfront costs are worthwhile.

8 Conclusions

The main contribution of this work is the development of a methodology for the development of whole systems railway models that provide practitioners with a common approach and tool for developing and specialising models for particular applications.

The research shows that although there is a considerable body of research and a number working projects within the railway industry and academia with respect to creating models which represent the railway system, a complete general railway/GTS model has not yet been attempted, see chapter 2.

This research has suggested a methodology that, if adopted, can facilitate the development of a problem and solution independent model of a complete GTS, see chapters 4 and 5. Using this approach, the general model can be specialised for particular applications. Once the model is developed it is reusable and extendable and can be applied to all or parts of a GTS.

As the methodology is also a guide , see chapter 4, it can drive a consistent approach to the application of systems engineering and promote systems thinking as it guides engineers through the process by:

- Providing a common organisation for the model;
- A structured process to identify stakeholders by modelling the complete context of a proposed new GTS or a change to an existing GTS, see chapters 4.3 and 5.2;
- Providing a structured and integrated approach to requirements identification, see chapters 4.4, 5.2, 5.3, 5.4, 5.5 and 5.6;
- Guiding a practitioner through requirements decomposition, see chapter 4.4;
- Guiding a practitioner through the process of system decomposition and allocation of requirements to the various parts of the system, see chapter 5.7; and

- Iterative validation as the model is developed against requirements.

The methodology also aids systems thinking. This is demonstrated by the results of the evaluation carried out by the representative body of engineers on the University of Birmingham's MSc course on Railway Systems Engineering and Integration, see chapter 6.

The research and evaluation carried out also confirms assertions made in the literature that there are widely differing opinions across practicing engineers within the rail industry regarding what is actually meant by the terms system and systems engineering. These leads to some engineers thinking that they are applying systems engineering when they are not and thinking that they do not apply systems engineering when in actual fact they do, what they do is just not "badged" as such. There being no clear view on the subject has led to the application of systems engineering and systems thinking across the industry to be sporadic at best and worst rarely applied.

Also some of the practical barriers identified in the literature to the application of systems engineering can be overcome by a common industry or project wide view of the system of interest and how to model it.

9 Further Work

The previous section states that this project has demonstrated that the modelling approach proposed can be used to model complex systems like a GTS. Also the feedback from the user group who evaluated the process was positive. Although these results are promising the process should be applied and evaluated in the real world on a real project to provide a proper test of its industrial applicability, if that is possible.

The demonstrations within this thesis are examples of what can be achieved, however it is necessary to apply this initial work in detail to a complete GTS subsystem or a complete GTS. This can only be done through a project that combines discipline expertise from all the areas identified during the project, including technical knowledge, operational knowledge and financial knowledge. A possible approach is to have MSc students take on a particular GTS subsystem for a dissertation and model it using the proposed methodology down to component level, this system model could then be put to a particular use and it outputs compared with other approaches and the real world. This would provide proper validation. Another project could bring them all together to form a complete GTS representation that can then be set to modelling real world GTS issues, such as the effects of perturbation on real routes and decision support for re-routing.

It has not been possible to evaluate completely how this work deals with the barriers that exist in Rail and other industries to the application of systems engineering, its thinking and tools. The feedback from the evaluation group was positive, but it needs to be further tested in an industrial environment.

10 References

Abed, S. (2010). European Rail Trafic Management System - An Overview. *2010 1st International Conference on Energy, Power and Control.* Basrah: IEEE.

Ackoff, R. L. (2010). *Systems Thinking for Curious Managers.* Axminster: Trinity Publishing Limited.

Ackoff, R. L., & Rovin, S. (2003). *Redesigning Society.* Stanford: Stanford University Press.

American National Standards Institute. (2003). *Processes for Engineeting a System.* Philadelphia: American National Standards Institute.

American National Standards Institute. (2015). *ISO 9000:2015.* Retrieved from webstore.ansi.org: https://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9000%3a2015

Anderson, A., Hirsch, R. T., & Adeney, W. (2003). *Developing Benchmarking methodologies for Railway Infrastructure Management Companies.* Retrieved from Imperial colleage: http://www3.imperial.ac.uk/pls/portallive/docs/1/8597698.PDF

ANON. (2005). Chapter 2. In ANON, *Introduction to Object-Oriented Systems Analysis and Design with the Unified Modelling Language, Version 2.0* (pp. 23-55). USA: John Wiley and Sons.

Anon. (2014). *TRAK*. Retrieved from en.wikipedia.org: http://en.wikipedia.org/wiki/TRAK

Anon. (2016). *Electric Traction Drives.* Retrieved from www.railway-technical.com: www.railway-technical.com/drives.shtml

Anon. (2016). *Rolling Stock Technical Pages.* Retrieved from Railway Technical Pages: http://www.railway-technical.com/trains/

ANON. (2018, May 20). *B-Method*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/B-Method

ANON. (2019, April 14). *OSI Seven Segment Model*. Retrieved from Webopedia: https://www.webopedia.com/quick_ref/OSI_Layers.asp

Artisan Architect. (2008). *SysML tutorial version 7.0-a.* USA: Artisan Software.

Association of Train Operating Companies (ATOC). (2016). *Key Train Requirements.* London: ATOC Limited.

Association of Train Operating Componies. (2016). *Key Train Requirements, issue 4.* London: ATOC Limited.

Bahill, T. A., & Szidarovszky, F. (2009). Comparison of Dynamic System Modeling Methods. *Systems Engineering Vol 12, No3*, 183-201.

Balmelli, L. (2007). An Overview of the System Modelling Language for Product and System Development. *Journal of Object Technology Vol 6, number 6*, 149-177.

Balmelli, L., Brown, D., Cantor, M., & Mott, M. (2006). Model-Driven systems development. *IBM System Journal, Vol 45, No6*, 569-585.

Barnes, P., & Gallop, M. C. (2015). Systems Modelling of Network Rail's Western Route. *Asset Management Conference.* London: IET.

Bayati, H., Iwnicki, S. D., & Stow, J. M. ((circa), 2004). *Rail Technology Unit Website.* Retrieved from Core: https://core.ac.uk/download/pdf/

Beasley, R. (2012, July). *Barriers to Systems Thinking*. Retrieved from ResearchGate: https://www.researchgate.net/publication/285405766

Begg, D., Fischer, S., & Dornbusch, R. (1994). *Economics.* Maidenhead: McGraw-Hill.

Berger, L., Lawrence, A., Roggenbach, M., & Seisenberger, M. (2017). Verification of the European Rail Traffic Management System in Real-Time Maude. *Science of Computer Programming*, 61 - 88.

Best, N., & Hyland, B. (2012). *UK Railway Systems Reliability - Modelling the Future - a case Study.* UK: Network Rail.

Boehm, B. (1988). A spiral model for software development and enhansement. *IEEE Computer, IEEE 21(5)* (pp. 61-72). IEEE.

Booton jr, R. C., & Ramo, S. (1984). The Development of Systems Engineering. *IEEE Transactions on Aerospace and Electronic Systems Vol AES-20 No 4* (pp. 306-310). Redondo Beach: IEEE.

Borst, P., & and Akkermans, H. (1997). Engineering Ontologies. *International Journal of Human Computer Studies*, 356-406.

Bousse, E., Mentre, D., Benoit, C., B., B., & Takaya, K. (2012). Aligning SysML with the B Method to Provide V&V for Systems Engineering. *Model-Driven Engineering, Verification and Validation 2012.* Insbruck: HAL.

British Broadcasting Corporation. (2011, November 24). *New York train crash: Metro-North derailment in Bronx*. Retrieved from BBC News: http://www.bbc.co.uk/news/world-us-canada-25176496

British Standards Institute. (1999). *PD R009-003 Railway Applications - Guide to the Specification of a Guided Transport System.* London: British Standards Institute.

British Standards Institute. (2002). *BS ISO/IEC 15288:2002 Systems Engineering - Systems Life Cycle Processes.* London: British Standards Institute.

British Standards Institute. (2010). *BS EN50126-1:1999 Railway Applications - Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic Requirements and Generic Process.* London: British Standards Institute.

British Standards Institute. (2010). *BS EN50126-2:2007 Railway Applications - Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 2: Guide to the Application of EN50126-1 for Safety.* London: British Standards Institute.

British Standards Institute. (2010). *BS EN50129-2:2003 Railway Applications - Communitication, Signalling and Processing Systems - Safety Related Electronic Systems for Signalling.* London: British Standards Institute.

British Standards Institute. (2011). BS EN50128 Railway applications. Communication, signalling and processing systems. Software for railway control and protection systems. *British Standard*. London: British Standards Institute.

British Standards Institute. (2011). *BS ISO/IEC/IEEE 42010:2011, Systems and Software Engineering - Archetecture Description.* London: British Standards Institute.

British Standards Institute. (2016). BS EN 61508-3:2010. Functional safety of electrical/electronic/ programmable electronic safety-related systems. Software requirements. London: British Standards Institute.

Chymera, M., & Goodman, C. J. (2016). Calculation of Train Performance. *IET Professional Development Course on Electric Traction Systems* (pp. 28 - 40). Stevenage: Institute of Engineering Technology.

Cockburn, A. (2000). *Writing Effective Use Cases.* US: Addison-Wesley.

Consell, D., & Mery, D. (2003). Foundations of the B Method. *Computing and informatics Vol 22, 2003-01-31*.

Corazzan, R. (2010, July 10th). *Theory and History of Ontology, a resource for Philosophers.* Retrieved from http://www.foramalontology.it.

Cox, J. (2014, January 24). *What is Enterprise Architecture*. Retrieved from Youtube: https://www.youtube.com/watch?v=V5Xisag6aN8

Davidz, H. L., Rhodes, D. H., & Nightingale, D. (2005, January). Enablers and barriers to systems thinking development: Results of a qualitative. *Conference of Systems Engineering Research.* Hoboken, NJ, USA: Researchgate.

De Neufville, R. (May 1995). Designing Airport Passenger Buildings for the 21st Century. *Transport Journal, UK Institution of Civil Engineers, Vol 111*, 83-96.

Delligatti, L. (2014). *SysML Distilled - A Brief Guide to the Systems Modelling Language.* London: Addison Wesley.

Demboski, M. (2006, December 13th). V/T SIC: The Vehicle/Track Interface System Model (VTISM). London.

Department for Transport. (2013). *http://trak.courceforge.net.* Retrieved February 5th, 2013, from http://trak,courceforge.net.

Department of Defence. (1974). *MIL-STD 499A Engineering Management.* USA: Department of Defence.

Department of Defence. (2001). *Systems Engineering Fundamentals.* USA: US Department of Defence.

Department of Defence. (2008). *Systems Engineering Guide for Systems of Systems.* US: Department of Defence.

Department of Defence. (2015). *Department of Defence Dictionary of Military and Associated Terms.* Retrieved from www.dtic.mil: http://www.dtic.mil/doctrime/dod dictionary

Department of Defence. (2018). *DoDAF Home Page .* Retrieved from Department of Defence Web site: https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_background/

Department of Transport. (2010). *Schedule 1 Appendix A, Train Technical Specification IEP-TECH-REQ-35, issue 5.* London: Department of Transport.

Department of Transport. (2015). *Northern Rail Franchise - Train Service Requirements.* London: Department of Transport.

Department of Transport. (2016). *Northern Rail Franchise - TSR1- Service Level Commitment.* London: Department of Transport.

Dghaym, D., Poppleton, M., & Snook, C. (2018). Diagram-led Formal Modelling using iUML for Hybrid ERTMS level 3. *6th International Conference on ABZ on ASM, Alloy, B, TLA, VDM and Z* (pp. 338- 352). Southampton, UK: Springer Verlag.

Dunford, N. C., Yearworth, M., York, M. D., & Godfrey, G. (2012, October 19). *Wileyonlinelibrary.com*. Retrieved from Wiley Online Library: https://onlinelibrary.wiley.com

Easton, J., Davies, J., & Roberts, C. (2011). *Railway Modelling: The Case for Ontologies in the Rail Industry.* Birmingham: University of Birmingham.

Electrical A2Z. (2018). *7 layers of OSI Model and their functions*. Retrieved from ElectricalA2Z Web Site: http://electricala2z.com/cloud-computing/osi-model-layers-7-layers-osi-model/

Elliott, B. (2014, April). Benefits of Adopting Systems Engineering Approaches in Rail Projects. *PhD Thesis*. Birmingham, UK.

Elphic, J. (2010). Why is Water Wet. *Conference: Railway Signalling and Control Systems (RSCS 2010), IET Professional Development* (pp. 250 -270). Birmingham, UK: IEEE.

Erd, A. (2007). Elements of the UML Model of the Rail Vehicle Maintenance System. *Journal o Polish CIMAC,* .

Estefan, A. (2008). *Survey of Model Based Systems Engineering (MBSE) Methodologies.* California USA: California Institute of Technology.

Ferrari, A., Fentechi, A., Gnesi, S., Magnani, G., & and Felleca, A. (circa 2014). *Adoption of SysML by a Railway Signalling Manufacturer.* Florance: University of Florance.

Ferrogalini, M. (2015). *MBSE in the Railway Industry Sector: Aslton ASAP Methodology.* Retrieved from OMG Wiki: www.omgwiki/mbse/lib/

Forsberg, K., & Mooz, H. (1998). *Systems engineering, faster, cheaper, better.* San Jose: Center of Systems Management.

Freidenthal, S., & Wolform, J. (2010). *Modelling with SysML.* Retrieved from John Hopkins Applied Physics Laboratory: www.jhuapl.edu/ott/technologies/doc/

Freidenthal, S., Moore, A., & Steiner, R. (2009). OMG System Modelling Language (OMG SysML TM) . *Tutorial*. Object Management Group (OMG).

Freidenthal, S., Moore, A., & Steiner, R. (2015). *A Practical Guide to SysML: Systems Modelling Language 3rd edition.* San Francisco: Morgan Kaufmann Publishers.

Goodman, C. J. (2004). Overview of Electric Railway Systems and the Calculation of Train performance. *Eighth IEE Residential Course on Electric Traction Systems* (pp. 1 -24). Stevenage: Institution of Electrical Engineers.

Gorski, E., Park, C., & Southworth, F. (2004). A Systems Engineering Framework for Design, COnstruction and Operion of the Next Generation Nucliar Plant. *INCOSE International Symposium* (pp. 236-245). Toulouse, France: INCOSE.

Gould, N., Mackaness, W., Bechhofer, S., Stevens, R., & Cooper, L. (2017). *Personalised Information Systems in Multi-Model Transportation Decision Making.* Edinburgh: Heriot-Watt University.

Greatbanks, J. (2014). *Crossrail Programme - Modelling and Electrification.* Retrieved from IRSE: www.irse.org/Knowledge/publicdocument/7 James Greatbanks Presentation

Gruber, T. (1993). *A transitional approach to portable ontology specification, Knowledge systems laboratory, Test report KSL 92 -71.* Stanford: Stanford University.

Guarino, N., Oberle, D., & Staab, S. (2009). *What is an Ontology.* Berlin: Springer-Verlag.

Haskins, C. (2006). *Systems Engineering Handbook: A Guide for System Life cycle Processes and Activities, International Council on Systems Engineering (INCOSE), version 3.* Retrieved from ingenieria: http://www.ingenieria.unal.edu.co/ASCI/sistemasycomputacion/doc/systemsEng/SEHandbookv3_2006.pdf

Haxthausen, A. E., Peleska, J., & Kinder, S. (2011). A Formal Approach to the Construction and Verification of Railway Contol Systems. *Formal Aspects of Computing 23*, 191 - 219.

Hinkelmann, K., Gerber, K., Karagiannis, D., & et al. (2016). A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. *Computers In Industry, 79*, 77-86.

Hoinaru, O., Mariano, G., & Gransart, C. (circa 2013). *Ontology for Complex Railway Systems Application to ERTMS/ETCS System, hal-00912733 version 1.* France: French Institute of Science and Technology for Transport, Development and Network (IFSTTAR).

Honour, E., Axelband, E., & Rhodes, D. H. (2004, October). Technical Report: The value of systems engineering. *Lean Aerospace Initiative*. Virginia, USA: LIA Consortium.

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2007). *Practical Guide to Building OWL Ontologies Using PRotege 4 and CO-ODE Tools, Edition 1.1.* Manchester: University of Manchester.

INCOSE. (2007). *INCOSE-TP-2004-004-02 Systems Engineering Vision 2020 Version 2.03, Technical Operations International Council On Systems Engineering (INCOSE).* Retrieved from INCOSE.ORG: http://www.oldsite.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf

INCOSE. (circa 2010). *What is a System: A Consensus of INCOSE Fellows.* Retrieved from INCOSE: http://www.incose.org/AboutSE/WhatIsSE

InteGRail. (2014, feb). *Key Performance Indicator Trees - How to define and measure the performance of Raulway Systems.* Retrieved from InteGRail ewb site: http://www.integrail.eu/documents/fs25.pdf

International Organisation for Standardisation. (2011, December 01). ISO/IEC/IEEE $2010 Systems Engineering and Software Engineering - Architecture Description. *International Standard*. Geneva, Switzerland: ISO.

International Organisation for Standardisation. (2015, May 15). ISO/IEC 15288 Systems Engineering - System Life Cycle Processes. *International Standard*. Geneva, Switzerland: ISO.

Japan Monorail Association. (2016). *Types of Monorail Guide.* Retrieved from nihon-monorail: http://www.nihon-monorail.or.jp/pamphlet/312

kais, Z., Sallak, M., Schon, W., & S, R. (2014). A UML approach for modelling and verification of railway signalling systems specificaitons. *10th International Conference on Modelling Optimisation an Simulation - MOSIM14.* Nancy: HAL.

Kamours, S., & Boulet, P. (2007). Model Based Testing for the ERTMS system with SysML and MARTE. *MoDELs 07, MoDeVVa workshop.* Nashville: Springer.

Kotusev, S. (2017). Enterprise Architecture: What Did We Study? *International Journal of Cooperative Information Systems*.

Langer, G., Lewis, R., & Roberts, C. (2008). Ontology Driven Railway RCM Integration. *INTElligent inteGration of RAILway systems.* Derby: InteGRail.

Lewis, R., Elphick, J., & Roberts, C. (2008). *RCM Data Integration Ontologies and the Third Tier.* London: Institution of Engineering Technology.

Linhares, V. L., da Silva, A. J., & de Oliveira, R. (2006). Empirical Evaluation of SysML through the Modeling of an Industrial. *ETFA'2006 - 11th IEEE International Conference on Emerging Technologies and Factory Automation* (p. 1 to 8). Prague: IEEE.

Lodemann, M., & Luttenberger, N. (2010). *Ontology Based railway Infrastructure Verification - Planning Benefits.* Kiel: Department pf Computer Science, Communication Systems Research Group.

London Underground. (2008). Standard for Rolling Stock 1-180 Issue A2. London.

Long, D., & Scott, Z. (2011). *A Prima for Model Based Systems Engineering 2nd Edition.* Blacksburg: Vitech Corporation. Retrieved from wordpress: https://tomwolf.files.wordpress/2013;12/mbseprimer2ndedition ful.pdf

Mantzos, A. (2016). *Cost Allocation Pilot Study, Modelling and Results - A Report for Network Rail.* Milton Keynes: Network Rail.

Mars Climate Orbiter Team Finds Likely Cause of Loss. (1999, Sept 30). Retrieved from NASA Web Site: https://mars.nasa.gov/msp98/news/mco990930.html

Martin, J. (1996). *Systems Engineering Guidbook: A Process for Developing Systems and Products.* Boca Raton: CRC Press, Inc.

McNulty, R. (2011). *Realising the Potential of Rail in Great Britian.* London: Department of Transport.

Meadows, D. H. (2009). *Thinking in Systems, a Prima.* London: Earthscan.

Ministry of Defence. (2018). *MOD Architecture Framework.* Retrieved from Ministry of Defence Web site: https://www.gov.uk/guidance/mod-architecture-framework

Mohan, R., & Arumugam, G. (circa 2012). Constructing Railway Ontology using Web Ontology Language and Semantic Web Rule Language. *International Journal of Computing Technical Applications Vol 2 (2)*, 314 - 321.

Mori, M., Ceccarelli, A., Lollini, P., Fromel, B., Brancati, F., & Bondavalli, A. (2018). System of Systems Modelling using a Comprehensive viewpoint-based SysML. *Journal of Software: Evolution and Process*, 1- 20.

Moris, C., Easton, J., & Roberts, C. (2015). *Position paper: Ontology in the Rail Domain, The Railway Core Ontologies.* Birmingham: University of Birmingham.

Murphy, N., & Roberts, S. (2012). *Systems Engineering - Railway Operators Get It Too.* Retrieved from Incodeonline: http://www.incodeonline.org.uk/Documents/Group/Railway

NASA. (2009). *Systems Engineering Handbook.* Retrieved from ocw.mit.edu: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-enineering-fall-2009/readings/MIT16_842F09_handbook.pdf

Nash, A., Huerlimann, J., & Krauss, V. P. (2010). *RailML - a Standard data interface for railroad applications.* Switzerland: Institute of Transportation Planning and Systems, Swiss Federal Institute of Technology.

National Defence Industry Association - Systems Engineering Division. (2010). *Top Systems Engineering Issues in teh US Defence Industry.* USA: National Defence Industry Association.

Neaga, E. I., Harding, J. A., & Lin, H.-K. (2006). Towards a meaningful manufacturing enterprise metamodel: a semantic driven framwork. *IMechE Journal Engineering Manufacture Vol 221 Part B.*, 499 - 511.

Network Rail. (2013). *Technical Strategy June 2013 - A future Driven by Innovation.* Retrieved from Networkrail.co.uk: http://www.networkrail.co.uk/publications/technical-strategy

Noy, N., & McGuiness, D. (2001). *Ontology Development 101.* Stanford: Standford University.

Object Management Group (OMG). (2006). *OMG SysML specification.* Retrieved from omg.org: https://www.omg.org/spec/SysML/1.4/About-SysML/

Office of Rail and Road. (2013). *2006 no. 599 - The Railways and Other Guided Transport Systems (safety).* London: Office of Rail and Road.

Office of Rail and Road. (2015). *Freight Rail Usage 2015-16 Quarter 1 Statistical Release.* Retrieved from orr.gov.uk: http://orr.gov.uk/__data/assets/pdf_file/0020/19307/freight-rail-usage-2015-16-q1.pdf

Office of Rail and Road. (2015). *Passenger Rail Service Satisfaction: Quality and Methodology.* Retrieved from orr.gov.uk: http://orr.gov.uk/__data/assets/pdf_file/0018/4374/passenger-satisfaction-quality-report.pdf

Office of Rail and Road. (2015). *Passenger Rail Usage 2014-15 Quarter 4 Statistical Release.* Retrieved from orr.gov.yk: http://orr.gov.uk/ data/assets/pdf file/0005/18095/passenger-rail-usage-2014-15-q4.pdf

Office of Rail and Road. (2015). *The Railways and Other Guided Transport Systems (Safety Regulations)) 2006 (as amended) - A Guide to ROGS.* London: Office of Rail and Road.

Office, N. A. (2015, Sept). *National Audit Office Website.* Retrieved from Planning and Delivery of the 2014 - 2019 rail investment progreamme: https://www.nao.org.uk/wp-content/uploads/2015/10/Planning-and-delivery-of-the-2014%E2%80%932019-rail-investment-programme1.pdf

Ogunsola, A., Reggiani, U., & Sandrolini, L. (2009). *Modelling Electromagnetic Fields Propergated from an AC Electrified Railway using TLM.* Kyoto: EMC09.

Paredis, C. (2011). *System Analysis using SysML Parametrics : Current Tools, Best Practice.* Retrieved from open.modelica.org: https://open.modelica.org/images/docs

Parsley, A., York, D. M., Dunford, C. N., & Yearworth, M. (2013). The use of a systems engineering process guide to accelerate improvement in systems engineering application and expertise. *Systems Conference (SysCon).* Institution of Electrical and Electronic Engineers (IEEE).

Peak, R., Burkhart, R., Feidenthal, S., Wilson, M., Bajaj, M., & Injoong, K. (2007). Simulation Based Design Using SysML Part 2, Celebrating Diversity by Example. *INCOSE International Symposium*.

Peak, R., Burkhart, R., Freidenthal, S., Wilson, M., Bajaj, M., & Injoong, K. (2007). Simulation Based Design Using SysML Part 1, A Parameter Primer. *INCOSE International Symposium*.

Peak, R., Friedenthal, S., Moore, A., Burkhart, R., Waterbury, S., Bajaj, M., & Injoong, K. (2005). *Experiences using SysML Parametrics to Represent Constrained Object Based on Analysis Templates* . USA: Georgia Tech.

Pearce, P., & Friedenthal, S. (2013). A Practical Approach to Modelling Submarine Subsystem Architecture in SysML. *Submarine Institute of Australia Technology and Engineering Conference*, (pp. 347 - 360).

Pickard, A. C., Nolan, A. J., & Beasley, R. (2010). Certainty, risk and gambling in the development of complex systmes. *20th INCOSE international Symposium*.

Plum, N. (2013). *TRAK Enterprise Architecture Framework Viewpoints.* London: Department for Transport.

Plum, N. (2016, January 14). TRAK Enterprise Architecture Framework. London: Department of Transport.

Plum, N. (2018, January 31). TRAK Enterprise Architecture Framework Metamodel. London: Department for Transport.

Pomona Colleage. (2017). *History of UML*. Retrieved from Pomona Colleage Web site: http://www.cs.pomona.edu/classes/cs121/supp/UML%20tutorial/history_of_uml.htm

Rail Safety and Standards Board. (2011). *The Railway Functional Architecture - Summary Report.* London: Rail Safety and Standards Board.

Rail Safety and Standards Board. (2012). *The Future Railway.* Retrieved from www.Futurerailway.org/: http://futurerailway.org/Documents/RTS 2012 The Future Railway.pdf

Rail Safety and Standards Board. (2014). *Annual Safety Performance Report 2013/14 Key Facts and Figures.* Retrieved from www.rssb.co.uk: http://www.rssb.co.uk/Library/risk-analysis-and-safety-reporting/2014-07-aspr-2013-14-key-facts-and-figures.pdf

Reichl, K., Fischer, T., & Tummeltshammer, P. (2016). Using Formal Methods for Verification and Validation in Railway. *10th International Conference on Tests and Proofs* (pp. 3 - 12). Vienna : Springer Verlag.

Richards, D., Stuart, A., & Hause, M. (2009). *Testing Solutions Through SysML/UML.* UK: INCOSE.

Rosenberg, D., & Mancarella, S. (2010). *Embedded Systems Development Using SysML.* USA: Sparx Systems Pty Ltd and ICONIX.

Royce, W. (1970). Managing the development of large software systems. *IEEE Procedures WESCON.* IEEE.

Sana, J., El Miloudi, K., Bourdeaud'huy, T., & Lemaire, E. (2010). European Traffic Management System Validation using UML/Petri Nets modelling Strategy. *European Transport Review Vol 2.*, pp. 113 - 128.

Scott, B., & Grasso, J. (2009). A Need for Formalisation and Auditing in Enterprise Architecture Approaches and Programs. *Journal of Enterprise Architecture*, 18-30.

SEBok. (2015). *System Modelling Concepts.* Retrieved from sebokwiki.org: www.sebokwiki.org/

Shift2Rail. (2018, May 26). *ST4RT*. Retrieved from Shift2Rail: https://shift2rail.org

Thanh, V. (2004). *Review of Systems Engineering Process and its Application to Railways.* Birmingham: Birmingham and Manchester Metropolitan Universities.

Thompson, D. (1995). *Concise Oxford Dictionary.* Oxford: Clarendon.

Transport for New South Wales. (2015). *Business Requirements Specification TMU AM 06010 GU.* Sydney: New South Wales Government.

Tutcher, J., Easton, J., & Roberts, C. (2017). Enabling Data Integration in the Rail Industry Using RDF and OWL - the RaCoOn Ontology. *ASCE-ASME Journal of Risk and*

Uncertainty in Engineering Systems, Part A: Civil Engineering, vol 3, no. 2. DOI: 10.1061/AJRUA6.0000859.

Umiliacchi, P., Henning, G., Langer, G., & Shingler, R. (2008). *Standardization Data Interchange Between Railway Systems: Integrated Railway Information System.* Derby: Bombardier.

Umiliacchi, P., Lane, D., & Roman, F. (2011). Predictive Maintenance of Railway Subsystems using Ontology based Modelling approach. *9th World Confereence on Railway Research.* Lille.

Umiliacchi, S., Bhatia, D., Brownlee, A., & Brown, C. (2018). *Enterprise Architecture with Railway Systems Engineering.* London: Network Rail.

UNISIG ERTMS User Group. (1999). *FRS V4, 29, Functional Requirements Specification.* UNISIG.

UNISIG ERTMS User Group. (2008). *Sub Set 26 (all) System Requirements Specifications.* UNISIG.

University of Birmingham. (2018). *Enterprise Architecture Framework.* Retrieved from University of Birmingham Intranet: https://intranet.birmingham.ac.uk/it/documents/public/architecture/Enterprise-Architecture-Framework.pdf

Weilkens, T. (2007). *Systems Engineering with SysML/UML - Modelling, Analysis, Design.* New York: Morgan Kaufmann.

Weisman, R. (2011). *An Overview of TOGAF version 9.1.* Retrieved from Open Group web site: http://www.opengroup.org/public/member/proceedings/q312/togaf_intro_weisman.pdf

White, R. D. (2014). Electrification and Protection. *IET Electric Traction Summer School.* London: Institution of Engineering Technology.

Wikipedia. (2018, May). *TRAK*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/TRAK

11 Appendix A – Previous Ontology Work

To demonstrate the applicability or otherwise of an ontology approach to developing a whole GTS model, an OWL Protégé model of the simplified DC traction system shown below in Figure 59 was developed.

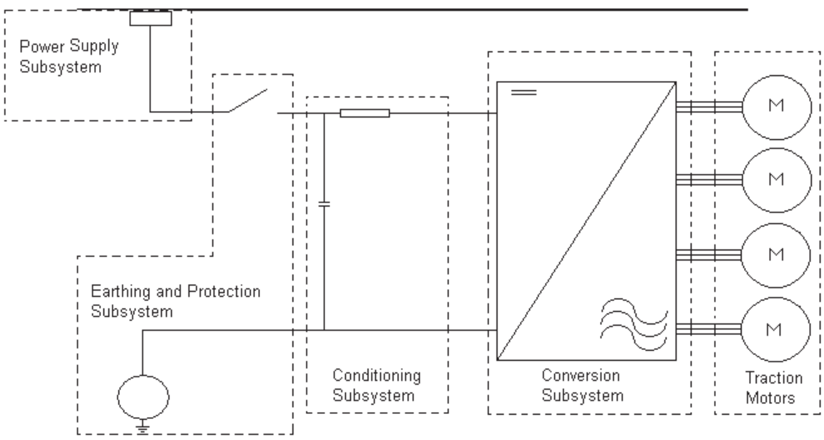

Figure 59 Demonstration Rolling Stock Traction System (author)

The first step toward developing the ontology was to frame the domain of discourse. In the case of this demonstration, this was the rolling stock traction system described in Figure 59. This was achieved by developing the superclass subclass hierarchy or taxonomy, given in Figure 60. Some thought was required at the start as to what the ontology was needed for, as this dictated how the system is decomposed (Noy and McGuiness (2001); Gruber (1991 and1993) and Gruber and Olsen (1994)). For demonstration purposes it was decided to model the simplified Rolling Stock traction system with respect to MTBF and cost, so an architecture centric view of the system was required. The taxonomy was translated in Protégé OWL as shown below in Figure 60.

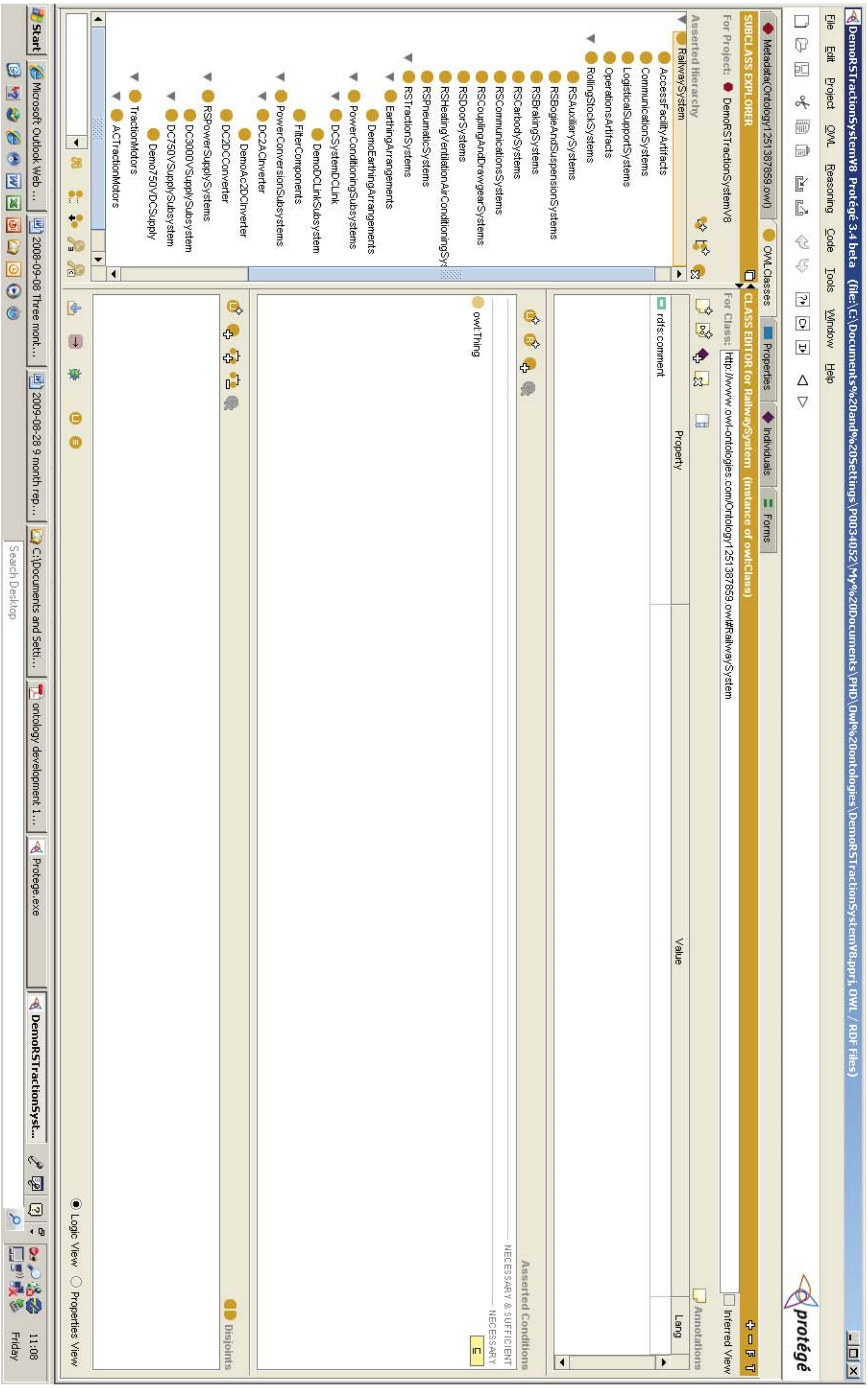

Figure 60 Class hierarchy for a Railway System (author)

It was also necessary to think about inheritance (Noy, McGuiness (2001)), for example, are there attributes that the Rolling System will inherit from the Railway system or that the DC750V subsystem will inherit from PowerSupplySystem. In the case of this demonstration there is inheritance. The system is being modelled in terms of MTBF and cost, in order to able represent MTBF and cost at the system/subsystem and component level, an individual that is member of any of the classes/subclasses will need to have a property that allows the input of cost and MTBF data, from appropriate inputs from outside the model. In this case the properties are defined at the highest level and are then inherited by each subclass as shown below in Figure 61 and Figure 62.

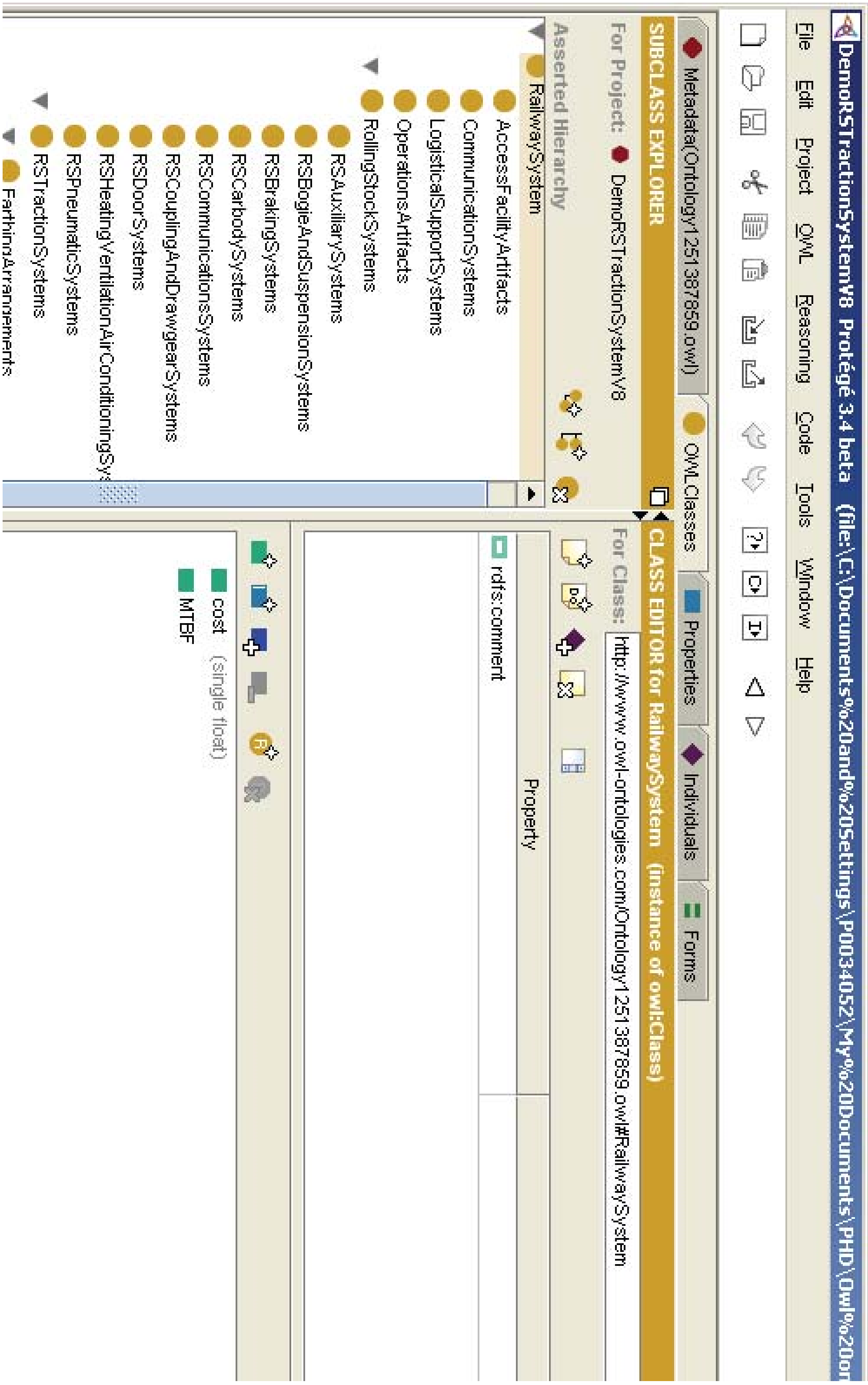

Figure 61 MTBF and Cost properties at the system level (author)

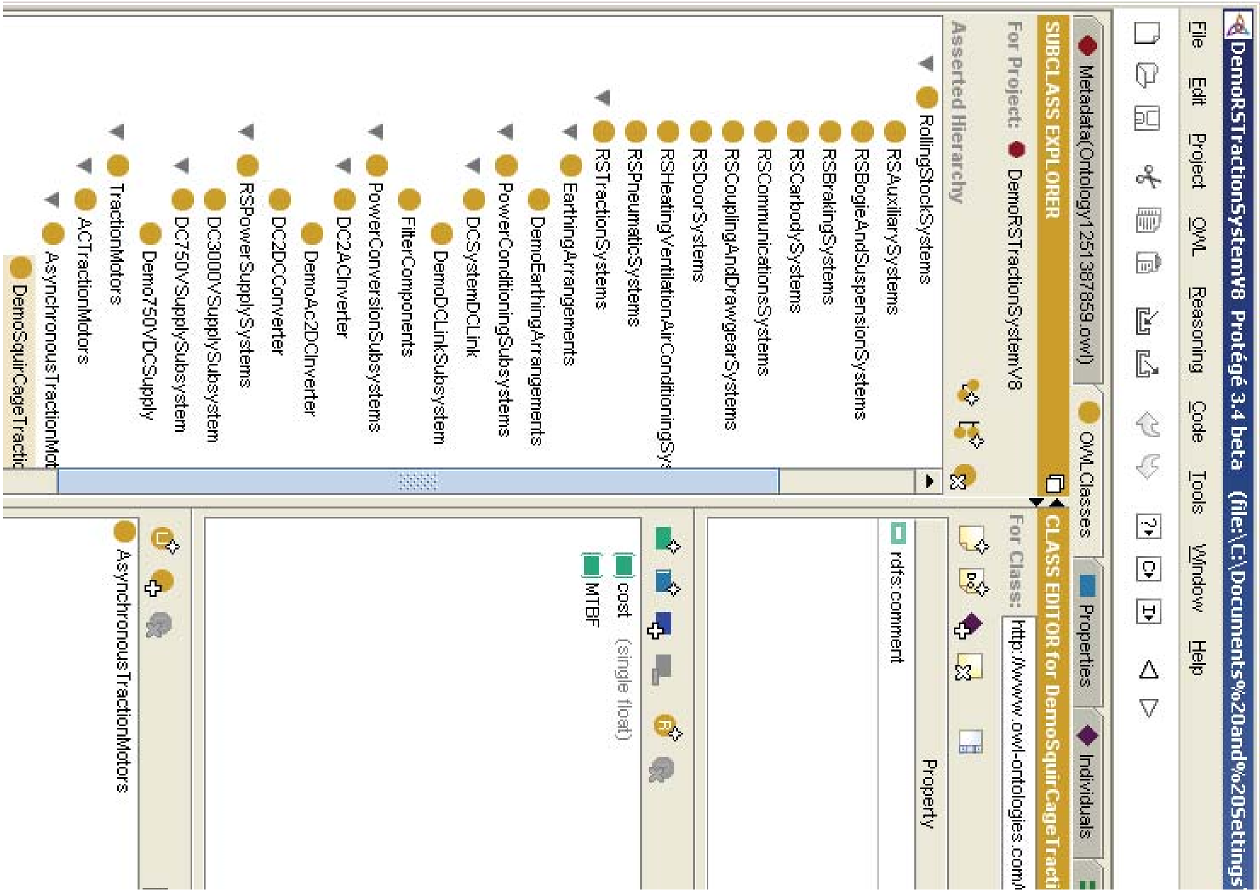

Figure 62 MTBF and Cost properties at the subsystem level (author)

The next step was to develop a decomposed representation of the general Rolling Stock traction system, in terms of this demonstration with the cost and MTBF parameters that needed to be modelled, but also in the context of the whole railway system.

Ontologies tend to be designed to be used by both human and computer agents, in order for a computer to ‘understand’ the system, the system and its subsystems needed to be defined exactly and logically (in the mathematical sense). It was therefore necessary to define what the relationships were between the various classes and subclasses. In ontology terms, to explain exactly what is meant by a Rolling Stock traction system it needed to be defined logically by its relations to other classes and subclasses. As can be seen from Figure 59 the definition of a Rolling Stock traction system is made up of five other subsystems:

- Earthing arrangements;
- Power Condition subsystems;
- Power Conversion subsystems;
- Power supply subsystems; and
- Traction motors.

It should be stated that all traction systems need to have at least one (two in the case of traction motors) of these components and indeed if a system has at least one of all these components then it must be a Rolling Stock traction system.

The highest level these relationships should be specified at is the general Rolling Stock traction system level. This gives the possibility of specifying a train at a higher level than component level. This is done by specifying restrictions to properties. The properties are created first. In order make sure that there is as little possible confusion later on, it is possible to have hierarchies of properties in Protégé OWL. It can be stated that classes are made up of various subclasses indeed subclasses have a relationship with classes in which they are part of. The hasPart property is created and a number of sub properties are created below it to identify particular systems, such as the hasEarthingArrangementPart, this is shown below in Figure 63.

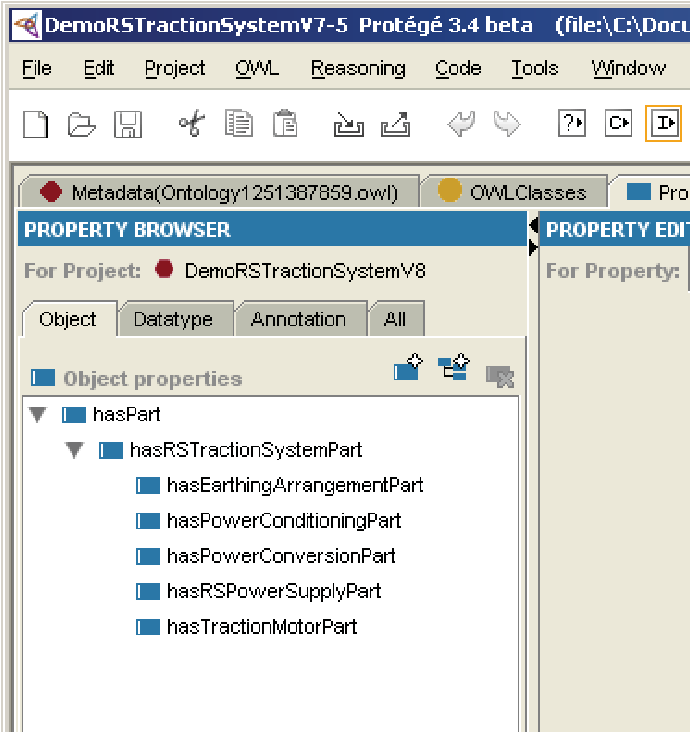

Figure 63 partOf properties (author)

These properties are then used to create relations between classes and further they are restricted to specify exactly what it means for an individual to be a member of a particular class.

For the RSTractionSystems this is a three stage process. It must be stated that some hasPart relations exist between RSTractionSystems and the other subsystems stated above. For this an existential restriction is used:

∃ hasPowerConversionPart some PowerConversionSubsystems;

In order to define the above even further it is necessary to state that an individual that is a member of the class RSTractionSystems can only have relationships along the hasPowerConversionPart property with individuals that are members of the class PowerConversionSubsystems, this is achieved through a universal restriction

∀ hasPowerConversionPart only PowerConversionSubsystems;

In plain English the above 2 restrictions are saying that a Rolling Traction System must have at least one PowerConversionSubsystem.

The same restrictions need to be set up of the remaining 4 RSTractionSubSystems. It can be stated that any individual that has the relationships described can be a member of the class RSTractionSystem. What it does not say is that if an individual has the relationships mentioned then it must be a member of RSTractionSystems. In Protégé OWL this is easily achieved by moving the restriction to the necessary and sufficient part of the asserted conditions widget as shown below in Figure 64.

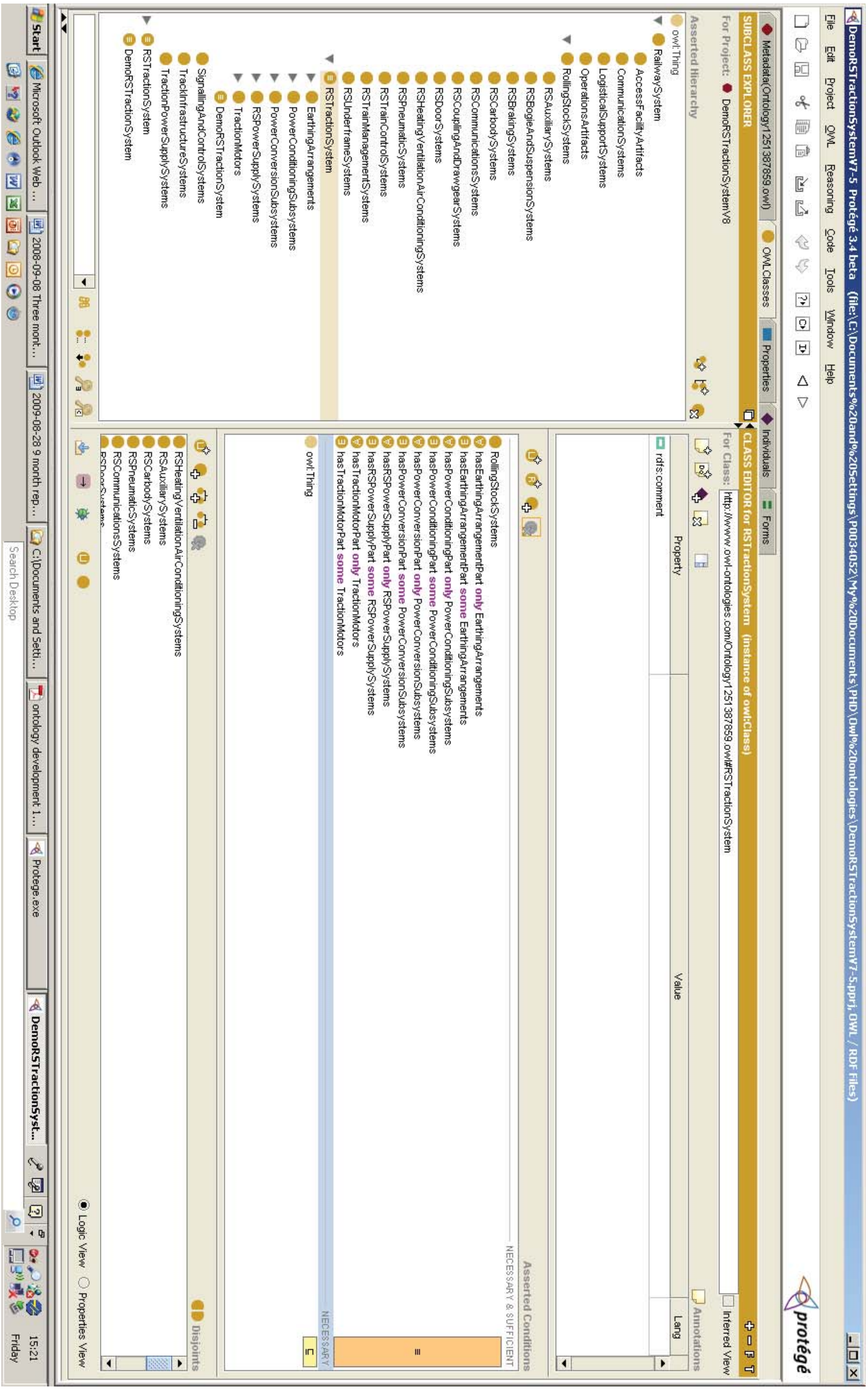

Figure 64 Conditions required to be a member of the class RSTractionSystems (author)

The conditions for an individual to be a member of the class RSTractionSystems are thus defined. However, what was required for this part of the project was a particular traction system made of members of the classes that have prefix demo in Figure 60. To achieve this a subclass of RSTractionSystems was created DemoRSTractionSystem. The same sort of restrictions as described earlier for the RSTractionSystem were applied by the relationships to the members of the classes that are prefixed with demo. Also another restriction is added one of cardinality. As this is a particular system it is possible to say how many individuals of the classes should be related to the individual from DemoRSTractionSystem. They will all require just one, except for traction motors, of which 4 are required. A full set of these restrictions is shown below in Figure 65

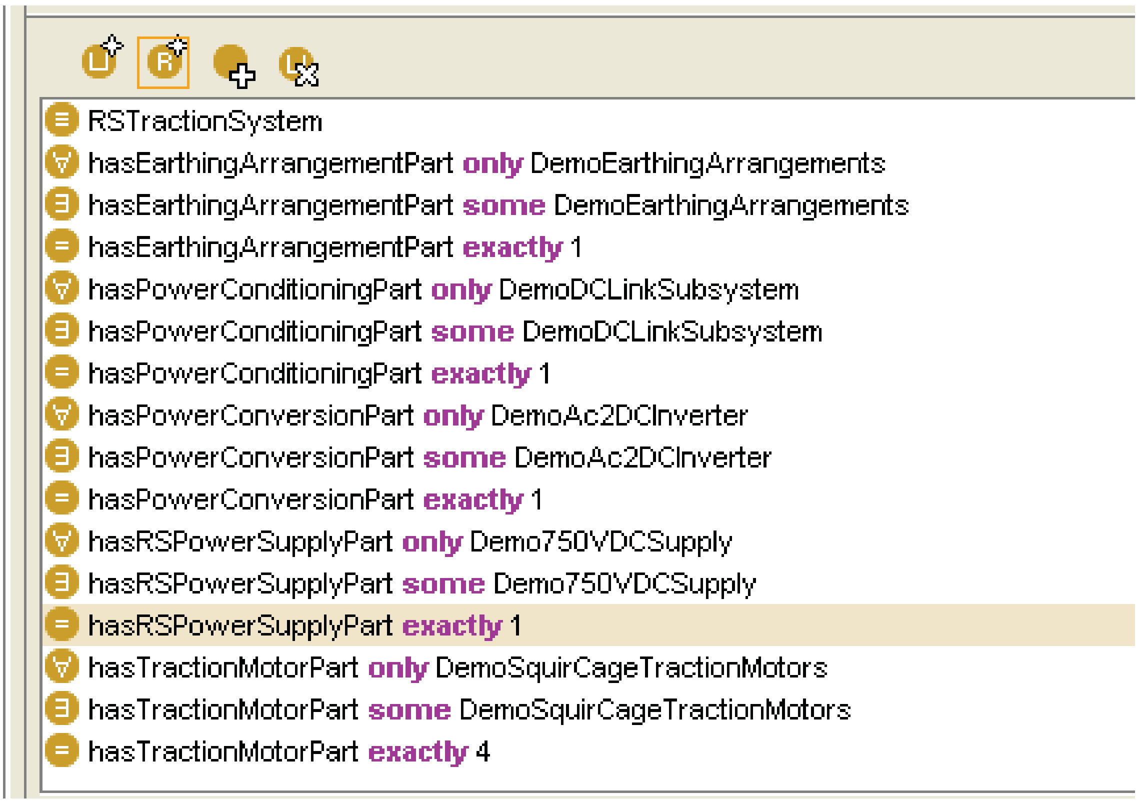

Figure 65 Definition of DemoRSTractionSystem (author)

Once the class hierarchy is developed a model can be instantiated to represent a particular use. In this case it is done from the bottom up in line the modelling strategy given in Figure

1. An individual of each demoClass is instantiated. When the individuals are created the user can see that data for MTBF and cost must be added. An example of this approach in terms of the traction motors is shown below in Figure 66.

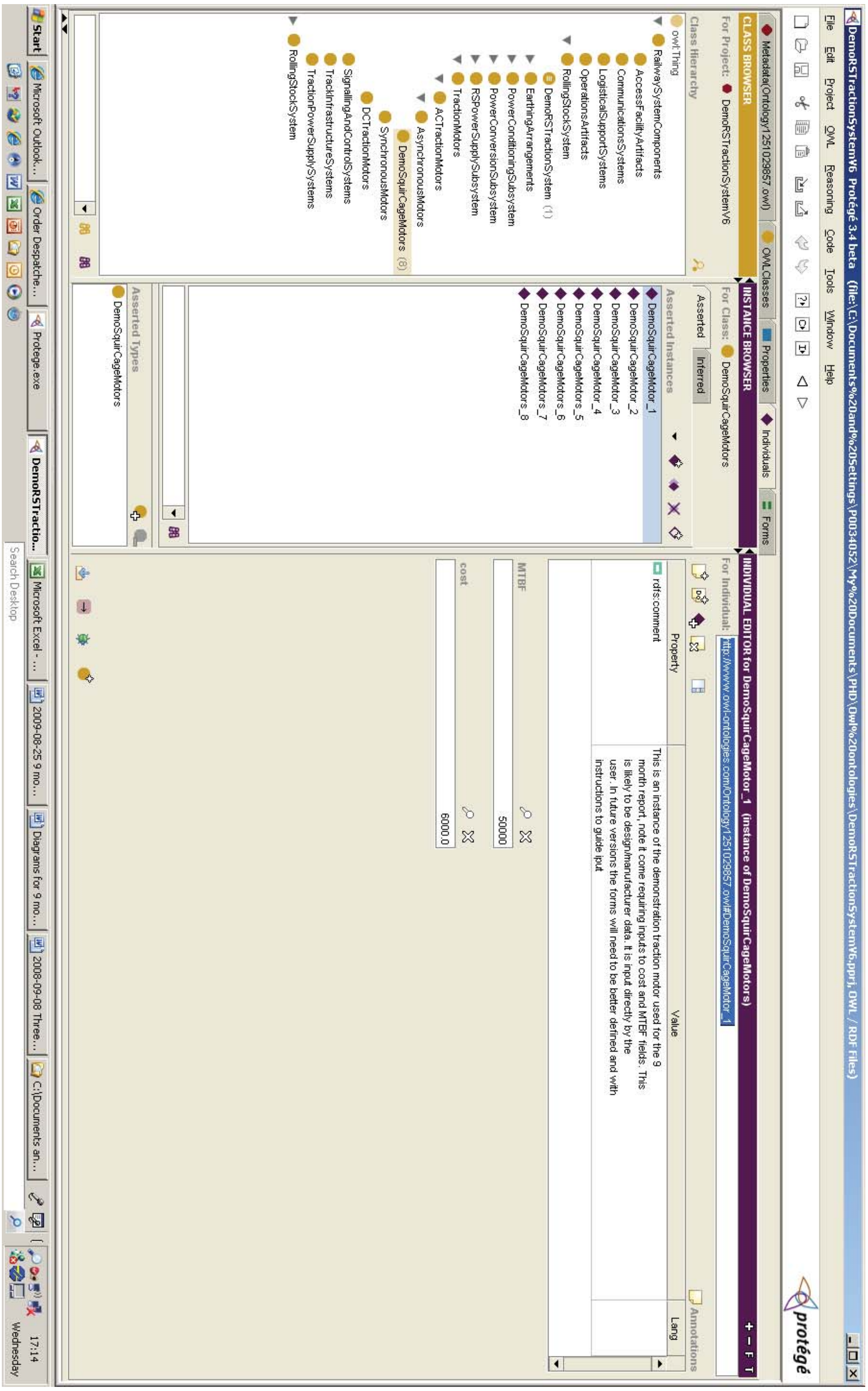

Figure 66 Instance of a traction motor (author)

Due to the inheritance within the structure, the same properties can be seen in the whole system instantiation as shown in Figure 67. It also shows the necessary components of the system, again this guides the modeller to the minimum requirements of the model.

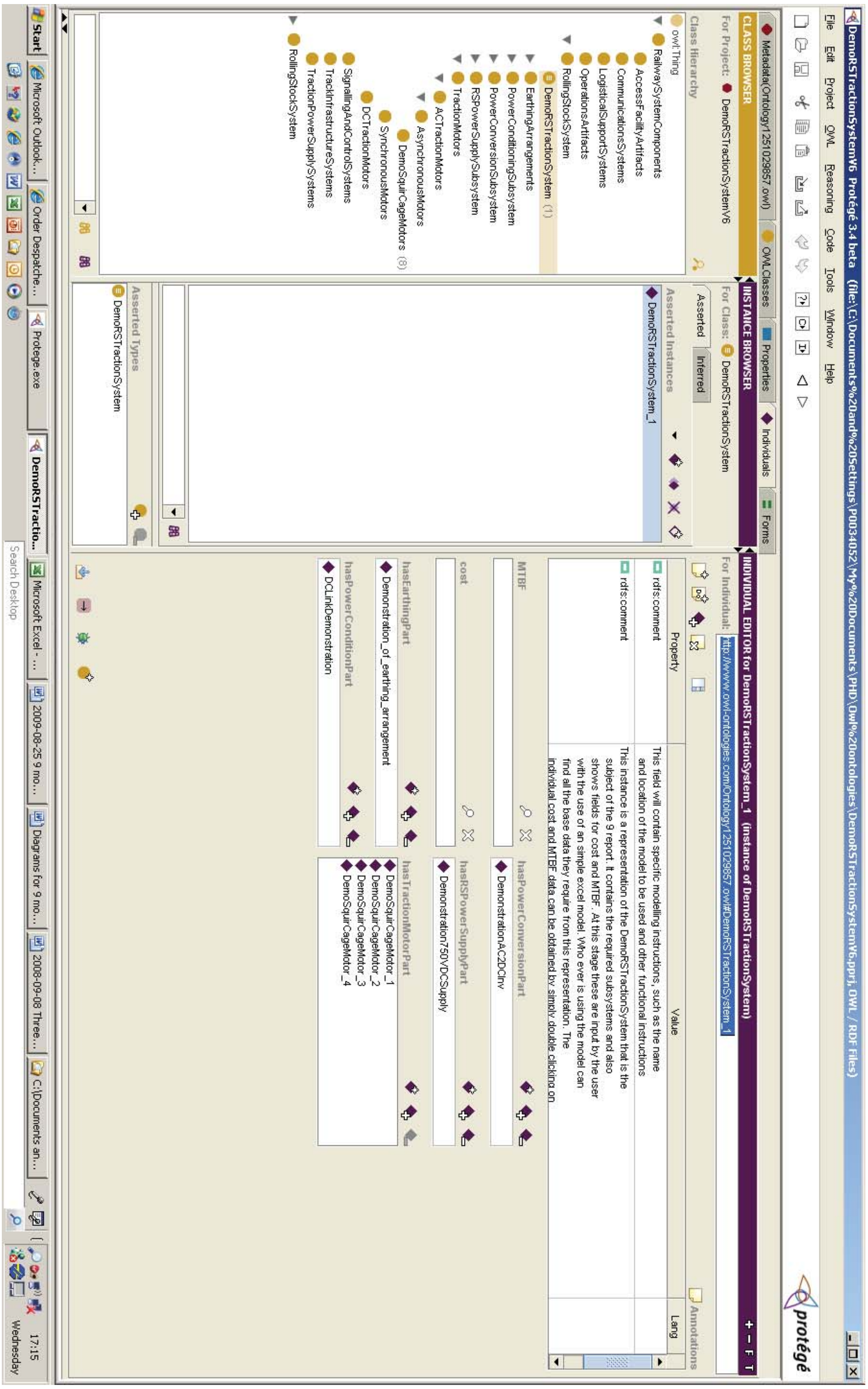

Figure 67 Instance of DemoRSTractionSystem_1 (author)

The model described above was not intended to provide an in depth understanding on how to build ontologies, it was intended to provide a comparison between the ontological approach and the object oriented (SysML/UML) approach, when trying to model a complex GTS, for the early stages of the research.

It has been possible with the demonstration model above to demonstrate that a general model of a GTS can be developed, then a particular model of railway system can be created within it and it can store information about that system. This information can then be fed in and out of the model at the appropriate places to furnish the correct data at the correct places within the model. This process is guided by information stored within the model about the system, what it should be made up of, the data that it should contain and its format.

This opinion is endorsed by the literature survey in general but particularly by, Borst, Akkermans (1997); Cheng-Leong, Pheng and Keng Leng (1998), Gruber, Olsen (1994), Izza, Burlat (2006), Kitawmwra (2006) and Noy, McGuiness (2001).

The main drawback with modelling a whole GTS with an ontological modelling tool, such as OWL Protégé, is that although it capable of achieving the aims of this project, it is really not very intuitive. Also once the model starts to grow it becomes very unwieldy, difficult to understand, control and maintain. It requires very specialist knowledge. It is very difficult to imagine that people who have a “day job” working with particular systems are going to be persuaded that it’s a good use of time to gain enough knowledge to make its implementation practicable, if it ever would be. For these reasons it was decided to carry on the project using the SysML/UML approach.

12 Appendix B - Notes on the application of SysML

12.1 Composition

Composition is captured in SysML by connecting a parent block with one or more sub blocks with a composition relationship represented by a line with a black diamond at the system end of the relationship and an open arrow head at the composition end, as shown below Figure 68. These relationships have a multiplicity at each end of the relationship which captures how many of the subsystem can be present or are required by the system. In the example below the composite end shows a multiplicity of 0..2 which means there can be zero, one or two subsystems. This is useful for a general model, when defining a subsystem that may or may not be present in the actual system being modelled but that could be present in other systems being modelled using the same basic model, the zero meaning that the system does not require that subsystem. At the system end of the relationship the multiplicity is one, however by convention and to make models more readable, the 1 is normally left out, so where no multiplicity is shown, the multiplicity is assumed to be one.

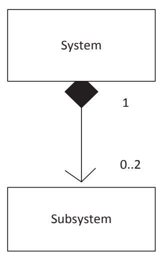

Figure 68 Composite relationship between a system and a subsystem (author)

Other relationships are also captured on the context diagram, these are associations which show the various blocks that depend on each other or need to act with each other, these can be uni-directional, shown as a line with an open arrow head in the direction of the relationship between two blocks, or bi-directional shown as a line with no arrow heads between two blocks, these also have a multiplicity as explained above. At this context modelling stage of the process, no further detail is shown on the diagram in terms of the relationships between the various system blocks. This is to ensure readability; greater detail is given in other SysML artefacts (diagrams) that are developed as the modelling progresses.

12.2 Activity diagrams

The philosophy of Activity Diagrams is based on (imaginary) Tokens moving between Actions. Actions can accept them as inputs and produce them as outputs. There are two types of Token, Object and Control. Object Tokens flow around the system and Control Tokens control how Actions consume and generate them. Object Tokens can correspond to anything that flows such as data or physical things like water, a part flowing through a production line or even a Journey.

It is modelled with a number of SysML Activity Diagram model elements:

- Initial Node; the notation is a solid black circle. This is a control node and is associated with an action through a Control Flow (Sometimes called a Control Edge). The Activity starts here;
- Activity Final Node; the notation is a bull’s eye with its centre as a solid black circle. This ends the Activity;
- Flow Final Node; the notation for this is a circle with a diagonal cross in it. This ends a flow, but does not end the Activity;
- Actions; the notation for an Action is a rectangle with rounded corners. There are two types of Action used on the diagram (although SysML has many others), the first is an Opaque Action (the light blue rectangle). An Opaque Action is an Action that can be decomposed no further, it is said to be atomic. The second type of

Action used is the Call Behaviour Action (darker blue rectangle), this action invokes another behaviour (one of three kinds, Activity, State Machine or Interactions). It is a way of decomposing behaviours from a high level to lower levels, for example we know that an Action like “passenger GTS Journey” is likely to have a number of sub Activities, as will be demonstrated below. The naming notation is slightly different from an Opaque Action, the name string is in the form of <action name> : <Behaviour Name>. The action name is modeller defined, the Behaviour Name must match the name of an Activity, State Machine or Interaction that is defined in the model, this is the part of the name that appears by default on the diagram. Also when the Call Behaviour Action invokes another diagram it has a rake symbol in the bottom right hand corner;

- Input and Output Pins; the notation for these is a small rectangle on the edge of an Action with a directional arrow inside. Pins accept Tokens that may represent units of information, matter or energy;
- Object Flows, the notation for these is a solid line with an arrow head at the destination. Object Flows connect the pins on the Actions together and Object Tokens flow along them;

Control Flows, the notation for these is a dotted line with an arrow head at the destination. Control Flows connect into actions directly (no Pins) and control when they are invoked and in what order.

The next stage in this process is to capture a generic structure based on the Requirements identified in the preceding chapters to represent a generic GTS at least as far as possible without becoming solution specific.

In SysML the Block is the principle unit of structure. It is used to define a type of system, subsystem, component, component interconnection, items that flow through the system, external entities, conceptual entities or other logical abstractions. A Block is an item of definition, it defines a type not an instance of a type (Delligatti 2014 pp 26), therefore it is ideal for a generic model that can guide a user/modeller, who can then choose instances of the types required, given certain rules and Requirements. A Block has structural features and behavioural features.

A Flow Port is the element used to model types of matter, energy or data that flow in or out of a Block (Delligatti 2014). The notation in the Artisan © SysML tool used in this thesis is a small rectangle on the boarder of a Block. It has an arrow inside it pointing in the direction of the flow, or two arrow heads pointing in opposite directions for bi-directional flow.

13 Appendix C - User Group Feedback questionnaires

13.1 Systems Engineering Questionnaire #1

13.1.1 Introduction

This questionnaire is designed to understand the current use of Systems Engineering and Systems thinking with the Railway industry at this moment in time. It is to support research carried out for a PhD thesis by G Greenland at the University of Birmingham. It is not the intention to hi-light any individual or organisation as good, bad or indifferent in respect of systems engineering capability, only to gain a broad view of Systems Engineering practice within the industry. The only identification you are requested to give is a student number to aid analysis, it will not be used to identify any individual in the final thesis. The questionnaires themselves may be attached as appendices to the thesis.

13.1.2 About the respondent

Student number: ___________________________________

Date: _ _/_ _/_ _ _ _

Which discipline area of the railway industry do you mostly work in (please tick only one)?

Rolling Stock general ☐

Rolling Stock Mechanical ☐

Rolling Stock Electrical ☐

Signalling ☐

Power Supply (OLE, 3rd Rail) ☐

Operations ☐

Infrastructure (track, structures, stations etc) ☐

Communications ☐

Describe in not more than 30 words what you think the word system means?

Describe in up to 30 words what is meant by the term Systems Engineering?

Given your definition above do you consider yourself involved in systems engineering in the course of your normal work (please tick only one)?

Majority of my time ☐

Half of my time ☐

Rarely ☐

Never ☐

13.1.3 Workplace Practice and Process

Is there a documented process for systems engineering at your place of work? ☐ Yes ☐ No

If yes, please add a brief description below (up to 100 words only please)

Have you used any of the following in your work, tick all appropriate:

Context diagram (any graphical representation of a project in the context of its environment) ☐

Systems engineering management plan (any kind of written plan detailing how the systems aspects of a project are to be dealt with) ☐

Interface management plan (a method of documenting interfaces between the various tangible non-tangible and human entities that your project effects) ☐

Systems Engineering Modelling Language (e.g. RailML, SysML, UML, TRAX) ☐

Systems Engineering modelling tool (a, usually software, environment in which a modelling language is implemented e.g. Artisan Studio©) ☐

Ontology modelling tools (DAML, OWL) ☐

Company propiratory system modelling/management tool ☐

13.2 Questionnaire 2

13.2.1 Introduction

This questionnaire is designed to elicit your opinions on the process applied to this assignment. It is ***NOT*** to capture your opinion on the actual assignment. Please answer the questions from that point of view only.

Answer all questions.

As with the 1st questionnaire the identification required is your student number for marking and evaluation purposes. No individual will be identified the subsequent thesis or any published papers.

13.2.2 About the respondent

Student number: ______________________________

Date: __/__/2017

Do you work in the rail industry?

If no skip section 3 and go to section 4.

13.2.3 Rail industry

Which discipline area of the railway industry do you mostly work in (please tick only one)?

Rolling Stock general ☐

Rolling Stock Mechanical ☐

Rolling Stock Electrical ☐

Signalling ☐

Power Supply (OLE, 3rd Rail) ☐

Operations ☐

Infrastructure (track, structures, stations etc) ☐

Communications ☐

Other, please state

____

Now please go to section 5.

13.2.4 Other industry

If non-railway, which industry do you work in?

____

Please state your core discipline:

Electrical Engineer ☐

Mechanical Engineer ☐

Civil Engineer ☐

Systems Engineer ☐

Project Manager ☐

13.2.5 Other, please state ____Task 1 System Context Diagram and Stakeholder Identification

With five being very much and one being not at all. Please rate the use of the context diagram

1. Did using the context diagram give you a clearer understanding of how the system fitted into its environment and the other systems it needed to interface with?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

2. Did developing the context diagram helped you identify stakeholders that not have otherwise been identified?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

3. Did using the context diagram guide you to understanding where the system boundaries were?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

13.2.6 Task 2 Requirements Elicitation and Stakeholder views

With five being very much and one being not at all. Please rate the approach to requirements elicitation.

Did the approach to requirements elicitation provide a structured approach to understanding and then finding requirements?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

Did you find that using the approach helped you ask concise and meaningful questions of Stakeholders?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

Did developing views and viewpoints for Stakeholders assist understanding what a Stakeholder needed from the system and how it could be delivered?

1	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐

13.2.7 Task 3 Activity Modelling

With five being very much and one being not at all. Please rate the use of Activity modelling.

Did activity modelling help you understand what functions the system needed to supply to deliver its purpose?

1	2	3	4	5
☐	☐	☐	☐	☐

Did the activity modelling identify more requirements?

Yes ☐

No ☐

13.2.8 Task 4 Functional Decomposition

With five being very much and one being not at all. Please rate the use of Functional Decomposition part of the approach.

Did following the approach make system and functional decomposition simpler?

1	2	3	4	5
☐	☐	☐	☐	☐

Did the approach help you to understand what functionality was delivered by what parts of the system?

1	2	3	4	5
☐	☐	☐	☐	☐

Did the approach make requirements decomposition and traceability understandable?

1	2	3	4	5
☐	☐	☐	☐	☐

13.2.9 Overall rating of the Model Based Systems Engineering approach

With five being very much and one being not at all. Please rate the Model Based System Engineering approach overall.

Would you use this approach in a complex project if you had the opportunity?

1	2	3	4	5
☐	☐	☐	☐	☐

Would you recommend the approach?

1	2	3	4	5
☐	☐	☐	☐	☐

What parts of the approach did you like and why? (use up to 30 words only)

What parts of the approach did you not find useful and why? (use up to 30 words only)

The end

Thank you very much for your participation