# A Preparatory Study Towards a Body of Knowledge in the Field of Formal Methods for the Railway Domain

# A PREPARATORY STUDY TOWARDS A BODY OF KNOWLEDGE IN THE FIELD OF FORMAL METHODS FOR THE RAILWAY DOMAIN

BY

APURVA KUMAR, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING & SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2015)
(Computing & Software)

McMaster University
Hamilton, Ontario, Canada

| | |
|---|---|
| TITLE: | A Preparatory Study Towards a Body of Knowledge in the Field of Formal Methods for the Railway Domain |
| AUTHOR: | Apurva Kumar<br>B.Sc., (Computer Science)<br>University of Pretoria, Pretoria, South Africa |
| SUPERVISOR: | Dr. Maibaum, Dr. Gruner (University of Pretoria) |
| NUMBER OF PAGES: | xi, 130 |

# Abstract

Bodies or Books of Knowledge (BoKs) have only been transcribed in mature fields where practices and rules have been well established (settled) and are gathered for any prospective or current practitioner to refer to. As a precursor to creating a BoK, it is first important to know if the domain contains settled knowledge and how this knowledge can be isolated? One approach, as described in this work, is to use Formal Concept Analysis (FCA) to structure the knowledge (or parts of it) and construct a pruned concept lattice to highlight patterns of use and filter out the common and established practices that best suit the solving of a problem within the domain.

In the railway domain, formal methods have been applied for a number of years to solve various modelling and verification problems. Their common use and straightforward application (with some refinement) makes them easy to identify and therefore a prime candidate to test for settled knowledge within the railway domain. They also provide other assurances of settled knowledge along the way.

# Acknowledgements

I am indispensably grateful to the following people without whose support and time, this thesis would not have been completed.

Professor Stefan Gruner, who guided and supported me every step of the way through this thesis. Professor Tom Maibaum for the opportunity of learning from the best.

Anne Haxethausen, Markus Roggenbach, Professor PJ (Hannes) Gräbe and Jackie van der Westhuizen for their insight into the railway domain. René Hosse and Jan Welte for their contribution to the sources of knowledge used in this project.

Professor Sergei Obiedkov for correcting my calculations of stability and adding to my understanding of Formal Concept Analysis. His format of results is preserved in the Appendix of this thesis.

Francesco Flammini, Hans True, Jérôme Lalouette and Stefan Östlund for responding to my Researchgate questions on the subdomains of the railway domain.

# Notation and abbreviations

1. BoK — Body(Books) of Knowledge
2. SWEBOK — Software Engineering Body(Books) of Knowledge
3. FCA — Formal Concept Analysis

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Railway software systems are safety critical. This means that the consequences of errors and bugs in the system can result in injury and even loss of life. It is important to handle the development and running of the system like all other engineering domains and have a strict set of rules to abide by when working on such a system. Usually, the knowledge and rules for a particular domain are collected in a handbook or Body of Knowledge (BoK) to which any practicing expert can refer during the different stages of development, running and maintenance of the system.

This project deals with attempting to define and describe what settled knowledge means with respect to the railway domain and where to look for such knowledge. Settled knowledge is a particular type of knowledge — one that all or most practitioners agree is fundamental to know while working within the domain. The project then attempts to determine whether the domain contains any knowledge that is settled and perhaps identify some of it in this process.

The aim is to discover the state of settled knowledge in the domain of railway software systems. Some secondary aims are to find the characteristics of the domain's knowledge, look out for the location of some of said knowledge and to analyse the data we collect using Formal Concept Analysis (FCA). Here, FCA aids in structuring of the knowledge and identifying patterns of use and importance of methods within the domain.

These aims work towards contributing an initial study to a domain-specific software engineering Body of Knowledge, similar to those that already exist in other fields of software engineering. A Body of Knowledge does not just collect all and any information in the domain. It is a careful selection of well tested methods and rules needed by practitioners to operate within the domain at a given time [30].

In engineering domains, Bodies of Knowledge are extremely valuable in standardising practice within the field. They are also useful in passing on these guidelines and rules that have been tried and tested over generations of previous engineers to new ones. Due to the safety-critical nature of railway systems, it is important for this knowledge to be collected and stored into a Body that is available for all practitioners of the domain.

Collecting this settled knowledge is not a trivial task. In fact, even identifying whether the knowledge in this domain is indeed settled is the first step towards creating a Body of Knowledge. Specifically, in this project, our focus is on a more specific area: studying the formal methods in use within railway systems (Rail-FM-BoK).

Looking at some of the uses of formal methods in railway modelling[1, 14, 7], a good case can be made for the propensity of the domain to contain settled knowledge [30]. Some conferences contain numerous papers on the formalisation of models of railway systems [1, 14]. Therefore, it is worth conducting a study to confirm this view.

The project limits are to review between 100 and 350 papers to extract knowledge. The time constraints give the upper limit and after studying attempts at this type of study in the past[40, 23, 31], it is understood that less than 100 papers would not give enough data for reasonable analysis.

Furthermore, the project will only need to construct a concept lattice representation of the collected data and analyse it in order to distill its properties. No custom formal concept identification algorithm will be applied or created. In fact, the Concept Explorer (Conexp) Tool will be used to automatically create the relevant concepts for our purposes. This allows most of the project to focus on the results of the data and its analysis rather than the method of constructing the lattice representation.

Since our aim is to search for settled knowledge within this domain, the project will attempt to find stable formal concepts within the formal context produced by this data. Stability of a formal concept will serve as the primary analysis of the concept lattice. Once the concept lattice has been constructed and the stability values have been calculated, the experimental part of the study can be concluded.

The next part involves the interpretation of the results of the stability calculations. If stability is found in many or all of the concepts, then a reasonable conclusion might be that settled knowledge exists within the domain. However, if most of the concepts are found to be unstable, then the most likely conclusion would be that settled knowledge has not been found within the domain.

A key concept here is what the threshold of stability would be. That is, which stability value is the cutoff for claiming that a particular concept is stable or unstable. This will be investigated as well.

Ideally, from the initial premise of the project, finding sufficient evidence of settled knowledge is expected, after examining papers in this domain. But it is entirely possible that this is not the case or simply that methods used in this project might not be optimal for the task at hand. Any of these solutions are contributions to the initial step towards creating a Body of Knowledge for the railway domain, which was the primary aim of this endeavour.

The next chapter outlines some of the related work to this project. Chapter 3 discovers the differences between science and engineering, in an attempt to differentiate between scientific knowledge and engineering knowledge, ultimately leading towards the need for a Body of Knowledge in specific engineering fields.

Chapter 4 gives background on Formal Concept Analysis and the Concept Explorer tool used in this study. Next, Chapter 5 discusses the sub-domains within the larger domain of Railway Systems and the reasoning for sub-dividing it.

The next few chapters outline the data, method and findings of this project. Lastly, the discussion and conclusion chapters evaluate the findings and give possible rationalisation for the results and their possible consequences.

# Part I

# Background

# Chapter 2

# Related Work

In his article within the same book, Wille gives his motivations for why formal concept analysis can be used to add mathematics to human thought, including fields that do not directly contain mathematics[46]. For this reason, formal concept analysis is also useful in finding implicit structures in a data set. Due to these two characteristics, and because FCA requires fairly simple mathematical knowledge[13], it is easy to witness the widespread use of this method and in particular to areas of software engineering.

Formal Concept Analysis(FCA) is a method of data analysis which describes relationships between a particular set of objects and a set of attributes. The book by Ganter, Stumme and Wille "Formal Concept Analysis: Foundations and Applications"[16], gives a complete description of the theory of formal concept analysis and algorithms around FCA that can be used to build concept lattices and extract information from them.

Calculations done on concept lattices allow the further analysis of the data in them. One of these calculations is called stability and there are different sorts of stability depending on your motivation. FCA also allows one to see some patterns and characteristics of the structure of data when categorised appropriately. This is useful when the structure of data, and in this case knowledge, is not known.

A discussion about knowledge discovery in databases supported by FCA can be found in here[28].

Use of formal concept analysis in software engineering tends to be in the discovery of implicit structures and relations between software modules and the maintenance of these software systems[4]. Especially in the object-oriented movement, FCA is a well-established tool for refining the structure of program code and finding hidden relations between classes. Other domain-specific uses of FCA stretch to formalising business knowledge[43] and helping lexicons in linguistic applications[34, 12].

Explicit formal specification to represent shared knowledge of a particular domain are referred to as ontologies within the field of computer science[13]. These are particularly of interest because software engineering is involved across a variety of domains and formalising the knowledge in these domains is vital to building correct software for any application in the respective domain.

Use of FCA in ontologies for various domains is widespread[13, 42]. The article by Philipp et al.[13] describes a method to merge various ontologies using FCA, and has also been studied at great length in the 1980s[6].

Formalising knowledge, in particular to the field of software engineering, is not easy. Software engineering has been largely criticised for being immature in its approach[41] and this view is backed up by appropriate research[18]. However, there is an attempt to rectify this problem in the form of creating Software Engineering Bodies of Knowledge(SWEBOK)[2].

This effort attempts to emulate other engineering fields that have a similar concept to store their knowledge. Bodies of knowledge are already known to be invaluable as a store of information and as a means to pass on stable knowledge to other practitioners in the domain[3].

A technical report with the intention of building a BoK for the Railway domain can be found here[19]. It is the aim of this thesis to aid the construction of this BoK by providing an initial study of the knowledge found in the railway domain.

This thesis attempts to achieve this aim by examining a large number of papers that discuss formalising knowledge within the railway domain and creating a concept lattice from the characteristics of these papers.

Use of FCA in representing a meaningful structure of knowledge communities[39] and then refining that structure using pruning by stability has been attempted before[40]. The project aims at following a similar method to refine the lattice produced so that meaningful conclusions can be made from the calculations.

To the best of our knowledge, the use of Formal Concept Analysis in the domain of Railway Software Engineering specifically for use in the classification of its knowledge has not been attempted before.

# Chapter 3

# The Philosophy of Science and Engineering

This chapter is a summary of the conceptual differences between science, engineering and technology, and how these differences have shaped the development of their knowledge. It also discusses how to identify a "mature" domain by studying the knowledge within the field and what constitutes settled knowledge from a number of perspectives.

In general, a body of knowledge contains core concepts, objects, their properties and interrelations in a particular domain. It is an ontology - a collection of knowledge for that domain. It represents this knowledge as a hierarchy of concepts using shared vocabulary to represent types, objects, properties and relationships.

We would need to know what the contents and structure of this knowledge is so that we can firstly identify it and then collect it into the Body of Knowledge. What does this knowledge look like in the domain of software engineering, specifically for the railway domain? Answering this question will help us later in seeking out so-called 'settled' knowledge in particular, which we will define soon.

The category of domain determines what the structure of its knowledge would be.[33, 3, 45] Since this question has not been asked for this domain, we don't know what shape this knowledge can take, but we can make an educated guess as to what it might be by examining the studied knowledge structures of science, engineering and technology[45].

To find the beginnings of the notion of mature knowledge in a field or domain, the theory of finalization is a good place to start. However at the time, no clear definition of the word 'mature' was described. The distinction between science, engineering and technology and their respective knowledge structures is the next step. After which an attempt is made to describe some structure of the knowledge we seek and a simple criteria for its being mature.

## 3.1 Finalization

The finalization idea was originally from a thesis produced in 1976 by Bohme, Van Den Daele, and Krohn[9, 32] and suggested a model of growth for science. In this model there are 3 phases of science growth:

1. Exploratory Phase — empirical and descriptive strategies for the collection of data are predominant.

2. Paradigmatic Phase — elaboration of theories. Extension and increased precision of these theories is an objective along with elimination of inconsistencies and improving the fit between data and theory.

3. Post-Paradigmatic Phase — application of established paradigmatic theories. It is this phase that would allow the influence of external interests on the direction of

the field once it is in this phase and considered "mature". However, measuring this or determining when the transition has occurred is extremely vague and different for the various fields of science.

The Finalization theory came under heavy criticisms and accusations.

Among other reasons including the socio-economic climate of the country and the forum used to discuss the theory, Bohme found that one of the problems encountered was the misunderstanding of the finalization thesis. The parties involved in its criticism superimposed their own ideas onto the meaning of the thesis without trying to appreciate its original argument.

One of the major points mentioned by the Finalization papers was that research needs to be planned so that resources can be efficiently used[9, 32]. This suggests that it is necessary to give science a direction if we are to use its resources efficiently. In other words, a general direction should be given to creative scientists but ultimately scientific institutions should be independent from economic interests. This freedom of research results in an optimum combination of men and team-oriented(planned) research[32].

But the largest problem was that in the presentation of finalization, the word 'science' is undifferentiated from other activities. There are four different activities that Pfetsch touches:

1. basic science

2. applied science

3. experimental development

4. teaching and connected services

The solution to this problem is discussed in the following section.

## 3.2 Science, Engineering and Technology

Pfetsch[32] states that the resources reserved for basic science should be unadulterated but the rest of the above activities are more of a socio-political apparatus and hence are more open for science-policy making. But unlike the Finalization thesis, the other activities such as applied science (or its derivatives[33, 3, 45]) are no longer assumed to be under the umbrella term of 'science'.

What Finalization does not describe is that the involvement of these external human factors causes the knowledge base to grow and shift into a domain based on science, but cannot altogether be called science since its general methodology, the scientific method, changes too. These forces morph science into what we call technology or engineering.

The exact differences and descriptions between the knowledge bases and methods of these three domains is described in various ways. Baber, in his comparison of electrical engineering and software engineering, only alludes to the need for considerable simplification, reformulation and repackaging of theoretical knowledge(science) in order for it to become a foundation for routine, widespread practical work(engineering)[5]. Vincenti and Maibaum try and describe the changes needed more accurately[45, 29].

Baber also mentions that engineering problems require commonly used procedures intended to ensure that human errors do not prevent the potential for error-free designs. Designs for a solution should be independently reviewed against their specifications, the contents of which are entirely written by the person solving the problem.

There is no universal solution to all problems. He describes all this without explicitly referencing science.

Poser and Arageorgis are much more candid[33, 3]. Both try to accurately describe the relationship and differences between science, engineering and technology. While Arageorgis tries to separate them by showing that they both solve different sets of problems, Poser gives a more detailed discussion about the philosophical differences of science and engineering.

## 3.3 Philosophical Differences by Poser

Poser refers to engineering as the science of artifacts and to science as the science of nature. Artifacts can only be relevant in engineering as it requires a real world solution or object as its end product. Science requires no such thing.

Like Baber, Poser assumes that engineering is entirely an applied version of theoretical and scientific work. It is applied science. Engineers make use of theoretical concepts, but there is no need for true laws or theories to be applied, only sufficient ones. An example would using newtonian relativity instead of general relativity in car manufacturing. Applied sciences have their own goals, and consequently their own methods[33].

Poser investigates a number of factors to differentiate science and engineering.

He begins by suggesting that engineering requires a certain level of creativity and science does not, but quickly comes to the conclusion that it is not a distinguishable quality since to find a new hypothesis or a technological solution, the creativity level is about the same.

Then Poser attempts to describe the idea of efficiency. At first, he suggests that for a solution to be appreciated on the engineering level, it needs to be efficient in the practical world. Hence there is no methodological difference between science and engineering and science is simply just an efficiency of hypothesis. Poser also lends his support to the theory of Finalization. He states that all sciences, including technology will have a new and different structure which would depend on nothing but efficiency.

However, later he describes efficiency as not being enough, since we still need rules to travel from one practical state to another. Engineering then needs foundational scientific research bound by classical scientific standards, and only then are rules grounded. Effectiveness then depends on the adoption of results of basic research to a means for an intended end. This view is similar to Baber[5].

Lastly, Poser differentiates Science and Engineering on their intentions and on their methods. Science tries to give a solution which is true for the whole universe and engineers just need a successful rule to apply, without any reference to a universal truth[33]. Engineering cannot avoid a certain teleological view since the problems they are trying to solve have been created by a human being for some intended end.

Poser claims that engineering cannot leave the scientific framework and its boundaries are given by these laws. But whether this is true, is argued by both Vincenti and Arageorgis[3, 45].

## 3.4 A Combined View of Science and Engineering

The paper by Arageorgis on demarcating science and technology starts by defining the difference between the scientific problem and the technological problem. And even though both disciplines rely on each other, they are separate in a number of ways. There is also a difference in the way these two activities evolve. Knowledge in engineering tends to be persistent, whereas in science the knowledge seems to be replaced as research progresses and understanding deepens.

According to Arageorgis, a scientific problem demands further elaboration of some of the elements of a given science and of their interrelations. It can only be called scientific if it is particular to a science. Science tends to leave out details of a problem that it feels are unnecessary or beyond its scope — therefore shrinking the problem to within its domain.

A technological problem demands for the production of a material artifact that will realise a particular desired state of affairs. This falls in line with what Baber describes in his paper as the notion of a solution being 'correct'. It is entirely dependent on the agent requiring the solution and the state of the environment of the problem.

Arageorgis disagrees with Poser that engineering cannot leave the boundaries of science. Engineering has its own knowledge base that science cannot define completely, as Vincenti and Maibaum describe as well[45, 29, 20]. In its effort to explain phenomena, a scientific investigation can wander at will as unforeseen results show new paths to follow. Such investigations never end, as they always throw up new questions. The essence of technological investigation is that they are directed towards serving the process of designing and manufacturing or constructing particular things whose purpose has been clearly defined[29]. So a scientific investigation doesn't end, but an engineering one will, when it reaches its goal.

However, he stresses the intimate relation between science and engineering. There are needs in science that require artifacts from technology in order to complete experiments. Similarly, a technological artifact may need a given science to solve a particular scientific problem. This gives rise to branches of applied science. This differs from Poser's view where engineering is just an applied science.

Both Arageorgis and Poser seem to identify technology as the activity that arises when external(human) interests are satisfied by scientific knowledge. This is the very idea finalization was trying to describe.[9, 32] Eventually, mature science will be used for external purposes. Arageorgis adds that this would not contaminate the pure science activity.

However, Arageorgis claims that both science and engineering can pursue their aims without external needs and demands. This is simply due to the conceptual and methodological framework that both disciplines use. He also alludes to the concept of effectiveness as something that is practically cost-effective to implement. This is what engineering seeks.

The paper has something to say on the development of technology as well - it is cumulative. Once a technological project has been realised, it is proved to be possible forever. It is the social framework which either invalidates or conserves past technological achievements. Progress in science also affects this process.

## 3.5 Classification and Identification of Knowledge Generating Activities

Vincenti takes the argument one step further and attempts to describe the exact structure of engineering knowledge. He explicitly states that engineering includes applied science but is not limited to it, and is in fact a knowledge generating field. The engineering method is quite different from the scientific method too.

He agrees with Arageorgis that engineering activity is largely composed of 3 phases: design, construction/production and operation. However, this is a rather simplistic view of the process as each phase is multi-leveled and might even contain some iterations and feedback loops. This multi-level view agrees with what Poser postulated as well[33].

Using case studies he argues the following points:

1. Successful engineering design does not necessarily need a theoretical basis, in contradiction to Baber and Poser[5, 33].

2. Key relationship between human behaviour and engineering requirements can greatly affect the outcomes — similar to the argument in Finalization[32] and Arageorgis[3].

3. Engineering has certain requirements that science cannot fulfill, indicating the need for extra knowledge.

4. Engineers develop methods to account for the absence of required scientific theory.

5. Requirements of production can have a reverse effect on design — not a simplistic model like Arageorgis described. This is also different from the scientific method. From these points it is clear that engineering needs a separate structure of knowledge and methods. He also describes 3 types of knowledge: descriptive, prescriptive and tacit. Descriptive knowledge describes the current state of a system/device. It describes it or gives it properties. Prescriptive knowledge gives a technology the conditions with which it is expected to succeed. It prescribes a state in which the object fulfills its function. Tacit knowledge is underlying knowledge that is hard to describe or define rigidly. It usually comes from practical experience. The combination of each of them in varying degrees produces the broader aims of knowing how and knowing

why. The growth of knowledge also increases the complexity of the subject matter.

In his book, Vincenti goes on to describe the six categories of Engineering knowledge. He describes this list as being not exhaustive or the categories exclusive of each other.

1. *Fundamental Design Concepts*

When designing a device or system, it is important to know the "operational principle" of the technology. What the purpose of the device in its context of problem, that is, how it works. Operational principles also exist for the components of devices. The engineer must be aware that the operational principle is a basis for design, and it is this principle that also provides the criteria by which success or failure is judged. It also defines a device.

This operational principle provides an important difference between technology and science — it originates outside the body of scientific knowledge and serves some technological purpose. The scientific knowledge of a machines tells us nothing about the machine itself, as they have an attached purpose.

The operational principle Vincenti describes in his book relates exactly to the operational theories Arageorgis and Baltas mention in their paper. These theories are practice-oriented and are completely subjected to the social determinations of the engineering activity.

Included in fundamental design concepts is the normal configuration of a device, that the engineer takes for granted in his work. There is a general shape and arrangement that are commonly agreed to best embody the operational principle. This is not definitive. In radical design, this usually changes.

Every object has an operational principle and also a normal configuration, once the device has become an object of normal, everyday design. These two things provide a framework in which normal design can take place. Still, engineers need more knowledge to translate these concepts into a concrete design.

2. *Criteria and Specification*

There have to be specific technical requirements for the technology that satisfy a particular operational principle and normal configuration. Therefore, the general, qualitative goals need to be translated into specific quantitative goals. These criteria themselves constitute an important part of engineering knowledge.

These criteria allow clarity of requirements , that is, no guesswork. These criteria may also apply at different levels of design, such as the overall device, or just a particular part of it. When circumstances become general enough, specifications can become general over the entire technology. These become part of the stored-up body of knowledge in engineering.

Here, another difference surfaces between science and engineering. They differ in purpose. Scientists do not aim at strictly defined goals but engineers have to do so in order to fulfill objectives. This view is shared in all the papers read for this study so far.

Therefore, these criteria and specification can be almost assumed to be uniquely meant for engineering knowledge.

3. *Theoretical Tools*

Most commonly, mathematical tools and models are a large part of the knowledge used to design a system or device. This is because of their quantitative use in the design process. This part of the engineering knowledge is based on scientific knowledge. However, pure mathematical tools have no physical context. Engineers have to adapt the mathematics for their use. This is also mentioned by Baber and Maibaum[5, 29] Next is mathematically structured knowledge, used in scientific studies for explanation. This still needs to be reformulated for use in engineering.

This repackaging and simplification is the process that Baber mentions in his paper on Heaviside's approach to electrical engineering.

After that comes the theories based on scientific principles but are motivated by and limited to a technologically important class of phenomena or even to a specific device. These theories are only useful while the area to which they apply are useful. For example, fluid dynamics. They tend not to need any reformulation.

Next are theories that are not necessarily based in strong science. They have a small base in scientific understanding, but also involve some specific assumption about phenomena crucial to the problem. Their power comes from allowing engineers to carry on with the design process, and modelling a complicated phenomena by some simple method, that may not be entirely correct, but works for the problem at hand.

The last mathematical tool are qualitative assumptions introduced for calculative expedience. These are usually just convenient tools that might even be wrong but provide acceptable results.

Second type of theoretical tools are the intellectual concepts that are used in both qualitative and quantitative assessment. They are a diverse group of knowledge that is sourced sometimes in science or just physical reasoning.

4. *Quantitative Data*

Mathematical tools are not useful without some quantitative data to apply them to a specific situation. This constitutes engineering knowledge because this is the difference between an abstract problem and a concrete one that engineers are normally commissioned to handle.

There are two kinds: descriptive and prescriptive. Descriptive knowledge describes how things are — what the status quo is and how it can be described with numbers. Prescriptive knowledge is knowledge of how things should be to meet a certain end. These requirements are generally what defines the requirements of a product. This obviously doesn't only apply to quantitative data.

5. *Practical Considerations*

These can come up in the form of constraints or requirements of the real world. These are not rigid and hard to define precisely like the mathematical models or the quantitative data.

These are not only reserved for design but production and operation as well.

In this case, experience and feedback from use is the source of most of this knowledge. Sometimes this translates to a design rule of thumb.

6. *Design Instrumentalities*

After having collected the knowledge, engineers need to know how to carry out the design. This is procedural knowledge. They involve the ways of thinking and the judgement skills by which the engineering method is carried out. Again, this is knowledge that is hard to put into a strict definition.

Some examples may include breaking down a large system/problem into smaller, more manageable parts and the act of optimization. But they also depend on the more theoretical tools and intellectual concepts discussed earlier. Ways of thinking can also include analogies to refer to certain phenomena or diagrams that cannot easily be put into words. Visual thinking is a very important skill that engineers need to know and require a certain amount of imagination and intuition.

This is very much tacit knowledge that sometimes can be taught but must also be learned through practical experience of both success and failure.
Although the categories described above are a good guideline for how some of the knowledge can be classified, these categories interact intimately and are not always clear cut. As Vincenti puts it, it is a tightly woven fabric. Of course, this knowledge has to operate within the hierarchically structured design process as described above. This is important for the epistemologically minded engineers - which is what our primary concern in this papers is.

He also addresses the question of where to look in order to find this knowledge. To that end, he describes the following knowledge-generating activities.

1. *Transfer of Science*

It is clear that this activity provides most, if not all the theoretical tools as required by engineers in the design process. It also provides quantitative data that can be transferred directly from scientific knowledge activity. This is not to say that science cannot be influenced by engineering. While engineering is an art, it utilizes the knowledge from developed and developing science[45]. This is still far from saying that engineering is just applied science.

2. *Invention*

Inventive activity appears more prominently with radical design but still has its elements in normal design. Each problem is unique to its environment and social context. This influences what the design goals are, thus in turn requiring the engineer to be slightly creative in applying his knowledge to the problem.

3. *Theoretical Engineering Research*

Engineers take theoretical to be synonymous with mathematical. Much of this occurs at teaching institutions and research laboratories. This is still important as formulae obtained by theoretical research provide precise measurement and new design procedures to designers of products. After all, this is exactly where the knowledge base of engineering came from. It is pursued with a different style and emphasis than scientific research - application rather than illumination.

4. *Experimental Engineering Research*

Its major contribution is to the quantitative data knowledge described above as well as providing new analytical concepts and ways of thinking. It is often hard to separate from experimental research in science since the approach and techniques are quite similar. However, engineers use a variation in parameter selection to supplement design when no applicable scientific theory exists. It also uses techniques like destructive testing which have no scientific standing at all.

Interactive science and engineering experimental research produce the most fruitful outcomes. Even though they are trying to be separated here for epistemological reasons, they function best when minimum distinction is being made.

5. *Design practice*

This activity contributes directly to criteria and specifications, practical considerations and design instrumentalities, rather obviously. Quite a bit of engineering knowledge is found out through actually practicing the art of designing technology. Some knowledge is difficult even to teach or to describe, and therefore must be learned through practical application.

6. *Production*

Provides useful information for quantitative data and practical considerations. Some things can only be determined if a working prototype is created first. Production is universal to all types of engineering, therefore, so is its knowledge contribution.

7. *Direct trial*

The products of engineering effort are always towards some end, therefore the artefacts always tend to be used and operated. This reveals special knowledge about the technology that could not be found in any other activity other than using the device or system on a regular basis. And just like science, testing is a large part of designing a particular technology. Of course in some fields of engineering, this is not possible.

Again, like the categories of knowledge, these activities do not exist in isolation and are not mutually exclusive.

Lastly, Vincenti talks about the phenomena of social agents that exist in engineering knowledge. This element influences the knowledge-gathering activities. Examples include design engineers, research engineers, applied mathematicians, academics, inventors and investors. This diversity of skills and intellect make engineering knowledge quite complicated to construct. It is this social factor that is also described by Baber, Arageorgis and Baltas to influence a large portion of engineering activity[3].

Vincenti describes a number of differences between science and engineering, showing that while both of them rely on each other[3, 45], the knowledge base for each is as different as the methods used to get there. It is this knowledge we seek for our Body of Knowledge - the structure of which he lays out as above. This work will form the basis for the identification and recognition of this knowledge.

## 3.6 Summary

Poser, Vincenti, Arageorgis and Baltas describe engineering as a multi-level activity. In order to design a device or a system, we need to first understand its operational principle - what it does. This is usually influenced by more than just pure scientific requirements. These devices and systems being designed are made for a specific purpose. This purpose is not given by a scientific requirement, but by a social context in which the technology is being intended for use. That overall goal drives the smaller aspects of design, as the goal is broken up into smaller design tasks that have their own requirements constrained by the overall goal as well as scientific and practical expertise. This builds a hierarchy of engineering design.

All the authors mentioned here agree on the influence of social contexts on the success of products of engineering. Without a particular goal in mind, engineering activity loses its meaning. This social aspect of engineering tends to be more focused on the higher levels of the hierarchy - the fundamental concepts, criteria and specifications. This social context then becomes built into the cognitive structure and content of engineering. To determine the structure of this engineering knowledge, it is important to look at the complex relationship between the context, the categories of knowledge and hierarchy of design.

Jackson said, an engineering handbook is not a compendium of fundamental principles, but it does contain a corpus of rules and procedures by which it has been found that these principles can be most easily and effectively applied to the particular design tasks established in the field. The outline design is already given. The methods of value are micro-methods, closely tailored to the tasks of developing particular well-understood parts of particular well-understood products[29].

From these insights, naturally settled knowledge would be the knowledge present in a Body of Knowledge(BoK). This would be consistent throughout the domain and appear in some form for all similar problems being observed in the domain. The knowledge would be found in the knowledge-generating activities outlined by Vincenti and would have the structure and categories as described above[45].

Therefore settled knowledge may contain fundamental design concepts that consist of operational theories and normal configuration, criteria and specifications, theoretical tools that consist of mathematical models and "mature" scientific theories. It also may consist of quantitative data and practical considerations common throughout the design process. One of the most important but harder to find knowledge is this so-called tacit knowledge in the form of design instrumentalities that give us the basis of how an engineer should think in order to be successful in his design task. It is this sort of knowledge that needs to be recognised as settled engineering knowledge as well, which is so often missed out in software engineering. The aim of the future BoK should be to discover these operational theories of software engineering for railway software systems.

The question of "how long" is still a bit vague. A starting point would be to find any consistencies over most or all of railway software specification. If this is viable, it would be a promising avenue with which to start gathering this knowledge.

In the case of software engineering, logic is the tool engineers use as a replacement of mathematics in other engineering branches. It serves the purpose of providing models in terms of what artefacts must be understood. These artefacts are conceptual. However, the laws that govern computation are man made and can change. It is this freedom of choice that sets software engineering apart. It is not bound by physical laws like other branches of engineering. We also have choice of formalisms and language[29]. This gives us greater freedom in descriptions but also misunderstandings in notations if they are not standardized. A future BoK would change that.

Formal methods are almost always used in the same manner, with slight adjustments depending on the overall aim. They are also abundant in the railway domain[15]. Focusing on formal methods would be the first step towards creating this BoK because they are very easily identified and filtered out from a large amount of knowledge already in the domain.

# Chapter 4

# Formal Concept Analysis using the Concept Explorer Tool

Formal Concept Analysis is a branch of lattice theory that was developed in Darmstadt in the 1980s. It can be used for analysis of simple attribute object tables (contexts) and exploration of different dependencies that exist between attributes and objects[47]. For example, a database can be seen as a many-valued context.

Formal Concept Analysis is useful because it is a theory of data analysis, knowledge representation and information management which identifies conceptual structures among data sets[10]. This project aims to look at the domain of railway software systems and identify conceptual structures within its knowledge sets, that is, a selection of the vast amount of technical papers in the domain. Formal Concept Analysis has been used successfully in many fields such as medicine, information science, software re-engineering, civil engineering and others[35].

The most relevant feature of Formal Concept Analysis is the production of graphical representations of inherent structures in data in the form of mathematical concept lattices. These mathematical lattices can sometimes be interpreted as classification systems in information science. The interpretation this project uses is quantitative analysis of the desired attributes within the lattice. Further explanation can be found in the Method chapter of this report.

## 4.1 Mathematical Description

The following description is taken from the book Formal Concept Analysis by Ganter and Wille[17] and the articles by Uta Priss[35] and Peter Burmeister[10].

A concept consists of an intension and extension. The extension of a concept are all formal objects which belong to the concept and an intension of a concept are all formal attributes that apply to all formal objects of the concept.

In this case, the objects will be the sources of knowledge. Published papers within the railway domain from various journals, conference proceedings or books that may contain settled knowledge.

The attributes are a collection of a number of characteristics of these papers. For example, when they were published, which sub-domain the papers deal with, what formal methods are used, and even some key techniques for software engineering and development within the field of railway software systems.

Concepts have relationships with other concepts within the context which is generally some form of hierarchical relationship. This subconcept-superconcept relationship is essential to finding structures within the data. The relationship also means that the extension of a subconcept is contained within the extension of the superconcept. Also, the intension of a superconcept is contained within the intension of the subconcept.

| | needs water to live | lives in water | lives on land | needs chlorophyll | dicotyledon | monocotyledon | can move | has limbs | breast feeds |
|---|---|---|---|---|---|---|---|---|---|
| fish leech | × | × | | | | | × | | |
| bream | × | × | | | | | × | × | |
| frog | × | × | × | | | | × | × | |
| dog | × | | × | | | | × | × | × |
| water weeds | × | × | | × | | × | | | |
| reed | × | × | × | × | | × | | | |
| bean | × | | × | × | × | | | | |
| corn | × | | × | × | | × | | | |

Figure 4.1: Cross Table Example[36]

So the mathematical model of a formal context contains formal objects, formal attributes and relationships between them. It can also be represented by a cross table(see figure 8.13). The elements on the left side are formal objects and the elements at the top are formal attributes. The relation between them is represented by the crosses. The resulting concept lattice can be seen in figure 4.2.

These images allow us to visualise the extents, intents and the formal context. From a particular concept (little circles or nodes in the lattice), the extent can be seen by following all paths that descend from that concept. The intent can be seen by all paths that ascend from the concept in question.

### 4.1.1 Formal Context

A context $\mathbb{K}$ has a structure $\mathbb{K} := (G, M, I)$ where G and M are sets representing objects and attributes respectively. I is a binary relation between sets G and M where $I \subseteq G$ x $M$ and $gIm$ means that the object $g$ has the attribute $m$.

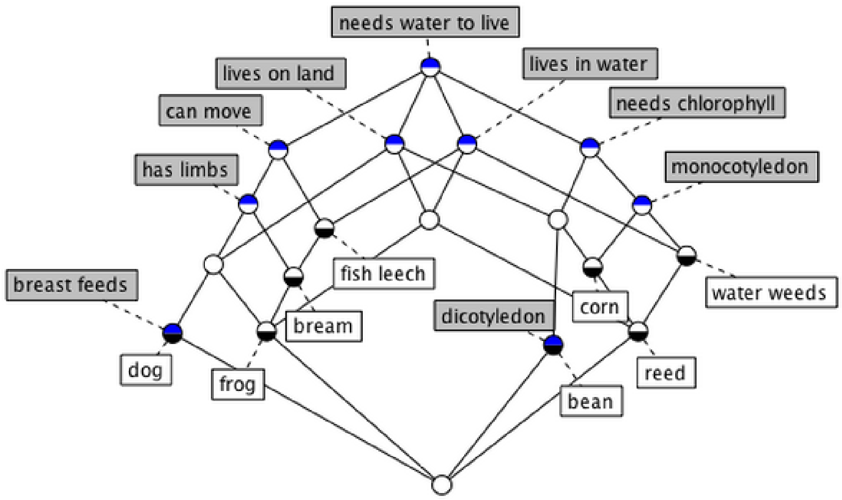



Figure 4.2: Concept Lattice Example[36]

We define two operators for arbitrary $X \subseteq G$ and $Y \subseteq M$ such that:

$$X \mapsto X^I := m \in M | \forall g \in XgIm$$

$$Y \mapsto Y^I := g \in G | \forall m \in YgIm$$

These operators have the following properties:

(i) $Z_1 \subseteq Z_2 \implies {Z_1}^I \supseteq {Z_2}^I$,

(ii) $Z \subseteq Z^{II}$,

(iii) $Z^{III} = Z^I$.

Within this context $\mathbb{K}$, we can define a concept as a pair (A,B) with $A \subseteq G$, $B \subseteq M$, $A = B^I$ and $B = A^I$. A and B are called the extent and intent of the formal concept (A,B) respectively. The mathematical meaning of the relationship between the subconcept and superconcept is as follows:

$$(A_1, B_1) \leq (A_2, B_2) \Longleftrightarrow A_1 \subseteq A_2 (\Longleftrightarrow B_1 \supseteq B_2)$$

The set of all formal concepts of context $\mathbb{K}$ together with their defined order relation is denoted by $\mathfrak{B}(\mathbb{K})$. Then, concepts can be constructed using the derivation operators to obtain, for $X \subseteq G$ and $Y \subseteq M$, the formal concepts $(X^{II}, X^I)$ and $(Y^{II}, Y^I)$.

### 4.1.2 Concept lattice

For an object $g \in G$, its object concept $\gamma g := (g^{II}, g^I)$ is the smallest concept in $\mathfrak{B}(\mathbb{K})$ whose extent contains $g$. Additionally, for an attribute $m \in M$, its attribute concept $\mu m := (m^I, m^{II})$ is the greatest concept in $\mathfrak{B}(\mathbb{K})$ whose intent contains $m$.

As described earlier, a context can be depicted using a cross table (figure 8.13) and a concept lattice can be pictured as a labelled line diagram as seen in figure 4.2. The name of each object is attached to its represented object concept $\gamma g$ and the name of each attribute is attached to its attribute concept $\mu m$. Following all lines attached to a object / attribute concept, we can clearly see the intent and extent if we move up or down, respectively.

Even attribute implications can be observed from a line diagram of a lattice:

$$A \rightarrow B \Leftrightarrow A^I \subseteq B^I with A, B \subseteq M$$

Visually speaking, concepts on lower levels are more specific that concepts on higher level, appearing near the bottom of the lattice. Concepts on higher levels are more general than concepts on lower levels and appear near the top of the lattice.

In terms of relations between concepts, some concepts are connected to each other, as represented in the lattice by a line between two circles. A concept at the top of the line is called a parent concept in relation to the concept at the bottom end of the line which is called a child concept. If a parent concept has more than one child, the children all share a subset of attributes of the parent.

If two concepts are not connected with this grand-parent, parent and child relation, nothing can be said about their relation using the lattice.

## 4.2 Conexp Tool

ConExp (Concept Explorer) is a Java-based, open-source FCA project. It has context creation and visualisation in a single tool. A number of lattice layout algorithms can be selected including chain decomposition and spring-force algorithms. The line diagrams also support various forms of highlighting and implements the largest set of operations from Ganter and Willes FCA book[17] including calculation of association rules and the Duquenne-Guigues-base of implications and interactive attribute exploration.

The Conexp tool implements the basic functionality needed for Formal Concept Analysis(FCA). It is released under the BSD-style licence[47]. Conexp provides a number of functionalities such as context editing, building concept lattices from a context, finding bases of implications and association rules in a context and performing attribute exploration.

Constructing the concept included building up the cross table, with the objects and attributes labelled appropriately and each one could be included or excluded from the generated lattice as needed. Clicking on a particular concept let the tool highlight the intents and extents making associations easier.

Each node (ball) in a concept lattice is a single concept. The default settings for Conexp is that the radius of the nodes represents the relative number of objects that exist within the concept. This can be changed to a number of different representations such as: "to own objects", "fixed radius", "of object extent", and "stability". During the use of this tool, some errors were found in the setting of the stability selection.

If the drawing of a node contains blue filled upper semicircle, that means, that there is an attribute attached to this concept. If drawing of node contains black filled lower semicircle, that means, that there is a object attached to this concept[47].

This tool was used in this project for the purpose of building a context for the research carried out on software systems in the railway domain. The objects represent the papers being reviewed for the project. The attributes represent the knowledge or at least a classification of it that can be found in each paper. This includes but is not limited to:

1. The formal method techniques used
2. The sub-domains of the railway system that they target
3. The years in which the papers were published
4. The other relevant keywords to the software development

Building a context with these attributes (and others) allows us to draw some conclusions about the use of formal methods in the railway domain. This tool helps to visualise and streamline this process.

# Chapter 5

# Subdomains of the Railway Domain

There exists any number of subdomains within the railway domain depending on the purpose of the classification and the areas of interest for research. This project seeks a division of the railway domain in a way that is convenient to research in formal methods within the railway domain.

The reason this division is needed for the identification of settled knowledge in the railway domain is that a list of subdomains can help identify the use of formal methods in different areas of the railway domain. It can also provide more information on how the domain is structured and which areas might need more attention. New patterns might even be found with certain formal methods are preferred for particular subdomains or system problems. This also helps the future Body of Knowledge by providing a possible structure to the domain which in turn might relate to a possible structure in its knowledge.

Bjorner's work on formal descriptions of the railway domain[7] uses a list of subdomains for a related purpose. He proposes a mathematical description of the entire domain starting with the first subdomain on this list[8]. Although it is not explicitly stated in his work that this list is particular to formal methods research in the railway domain, this is a good starting point for this project because his intention is to formally describe the entire domain, which formal methods attempt to do by formalising the description and behaviour of systems.

But it is not enough to only rely on this list. Expertise in railway systems is needed to ensure that the list of subdomains used is complete and no major part of the railway domain is missing. For this, a number of experts in the industry were consulted for their opinion and if there was something obviously missing from it. The forum ResearchGate[25] as well as private emails to domain experts were used for this communication. For many experts, the list was complete and had no gaping holes in it, although ultimately, it was agreed that the division is quite subjective and could be modified as required according to the use of the list.

An extra addition to this list was made after reviewing some papers within the domain and identifying a missing section that is considered large enough for its own subdomain. It is called "Train Operation". Many articles and academic papers in the sources of knowledge dealt specifically with the modelling of behaviour and systems that support the operators of trains, i.e. their drivers. There was also a fair amount of research into driver aids and management in an effort to reduce human error in the operation of railway systems. All these were collected in this new subdomain and did not fit appropriately in any other domain.

## 5.1 List of Subdomains

A large part of the following list is taken from Bjorner's TRain effort for creating a domain theory for railway infrastructure[8].

### 5.1.1 The Net

This includes the design and structure of a network of tracks, the composition of rails, be they linear or curved simple pairs, or junctions, or crossovers, etc., the means of switching junctions and switchable crossovers, the means of setting signals, etc. Also the meaning of routes, open and closed in a net, of inserting and removing parts of the net (as in construction, maintenance or downsizing), etc[8].

### 5.1.2 Timetables

The subdomain includes timetables as seen by passengers, schedulers, dispatchers, signalling staff, engine men, etc.

### 5.1.3 Scheduling and Allocation

There are different levels of scheduling and allocation: from strategic to tactical to operational concerns.

**Spatial Resource Scheduling and Allocation:** Scheduling in time, and allocation in rail net topologically of trains to routes according to timetables. This includes optimally using single line stretches between stations of individual station topologies while obeying rules and regulations for the operation of trains.

**Task Scheduling and Allocation:** A domain description of the operational resource management aspects of signalling includes description of the rules and regulations normally characterising plans for the setting of junctions and signals.

### 5.1.4 Traffic - Monitoring and Control

Train traffic is the progression of trains across the network topology over time.

**Signalling:** Procedures where junctions are switched and signals set according to plans. Further a domain description of this facet must include the sensing of train positions, road level gates, etc.

**Despatch:** Covers the layout, use and update of train running maps, including the current interface between dispatchers and train engine men and the future automatic control interface between stationary and mobile control centers.

**Monitoring:** Covers means and ways of locating trains (and rolling stock) - including descriptions of the technology by means of which we record train locations and relate these to plans.

**Control:** A domain description must include varieties of control: from manual via partial to fully automatic control of the despatch of trains and of the setting of junctions and signals. These descriptions include real-time and safety critical aspects.

### 5.1.5 Rolling Stock

This includes the freight cars, passenger wagons, locomotives, and others that make up trains. At any one moment in time, some such stock stand idle on tracks at stations, others are shunted around the main parts of stations, or are marshalled in freight yards, or are subject to maintenance or preparation, or are part of scheduled trains.

### 5.1.6 Passenger Handling

In particular, this involves any handling of passenger behaviour or needs required for the operation of railway systems. Also the specific concerns that occur when transporting people in large numbers and across distances. Modelling the flow and behaviour of pedestrian traffic in stations and railways in various situations.

### 5.1.7 Freight Handling

Freight handling involves the tasks necessary to deal with the transport of goods using trains and the specific requirements that come with moving large amounts of goods from one place to another. This includes the efficient management of finances and resources. Much of this can be modelled using some common formal methods such as linear programming and petri.

### 5.1.8 Train Operation

This is an extra subdomain that was added to account for the people and the requirements to operate a train, including drivers, conductors and operators. It also includes the infrastructure and information that is necessary for these individuals to operate the trains efficiently.

Many railway-related accidents mitigating factor is human error and the ability of train operators to react appropriately to emergency situations. This project found a large amount of research into modelling the basics of train operation duties in a critical situation so that the risk of human error is reduced significantly. All this modelling and behaviour description is encompassed in this domain.

# Part II

# Own Work and Findings

# Chapter 6

# Data Sources

In order to extract settled domain knowledge from the railway domain we first need to find the sources of knowledge within this domain. This would consist of any technical document produced within the railway domain. The sources of knowledge are any of the following:

1. Industry Standards and guidelines laid out by governing bodies.

2. Papers and articles written as a result of research done in the domain.

3. Requirements documents produced by domain experts and specialists.

Industry standards contain a lot of domain specific knowledge but are often expensive. Their knowledge is often dispersed in other sources and can be identified much earlier. Requirements are subjective to a particular application of domain knowledge and to a particular group of experts within the domain. Therefore, the easiest and most reliable method is to survey a number of papers in the railway domain and extract possible knowledge from them.

The next question to answer is what conferences and journals to use. There are many possibilities since the railway domain is large and has been researched extensively. Possible conferences or journals should relate to the modelling or development of the use of computer science and software engineering in the railway domain. They should also span a significant amount of time so that the data obtained is widespread.

The sources used in this project were picked for the reasons that they covered a substantial amount of time and were relevant to the use of formal methods in applications to the railway domain. They were also suggested by experts in the domain (see Acknowledgements). They are:

1. Proceedings of the IFAC Symposiums on Control in Transportation Systems, Elsevier Publ. — 1975-2012[21].

2. Proceedings FORMS-FORMAT, Springer-Verlag, 2010-2014[21].

3. Proceedings SAFECOMP, LNCS, Springer, 2005-2014.

These conferences provided more than three hundred papers related to the railway domain. Many papers did not relate to the use of formal methods in railways or were discussions or predictions of the current status of the domain. Due to this, only one hundred and fifty were used in the lattice.

Other sources were also discovered while looking for more papers, though there was not enough time to single out the appropriate papers and include them into the lattice. If in future, there is interest in using this method, the following sources could be used to track down more domain knowledge on the subject:

1. Conferences and Journals such as:

   (a) FORMS/FORMAT, Springer-Verlag, 1998-2014.
   (b) ISSRE, International Symposium on Software Reliability Engineering, 1996-2012.
   (c) COMPRAIL, WIT Press, 1987-2010.
   (d) IEEE workshop on Real-Time Applications — RTAS, 2000-2015.

2. Standards, Guidelines and Regulations by governing bodies, such as, ISO/IEC 15288, IEC 61508 and IEEE 1558-2004.

3. Databases that contain knowledge from the domain, for example, IEEE Xplore, Springer, ACM.

4. Technical Reports From Industry, for example, reports from specific projects within the railway domain.

One large source of domain knowledge is the domain experts themselves. Some were used in this project to suggest sources of knowledge. Domain experts are responsible for creating requirements and documentation to accompany it for a particular project or problem within the domain. It is important that this knowledge is transferred correctly from the domain experts to the software engineering experts. A lot of formalisation on this topic has been explored, and it is important to look at when determining how domain knowledge is captured within the context of software engineering.

# Chapter 7

# Method

A systematic set of logical steps are used to conduct scientific research. In this chapter, the steps taken in this project are outlined and reasons behind each step are also given. There is also some discussion on alternative methods or important questions that can be answered at each step.

Methods of analysis are explained and some important questions are answered, such as: what is being searched for in the data? What is the criteria applied in the analysis? Why is one method chosen over another? Examples are also given where necessary and what information is expected at each step is also included. For a complete understanding of Formal Concept Analysis, refer to Chapter 4.

**Step 1: Collect Sources of Knowledge**

The result of the entire project may hinge on the decision of where knowledge in the domain is found. The main aim of this step is to collect all the relevant papers from the sources identified in Chapter 6 and also mentioned below.

There is something to be said about what sort of knowledge would appear in the sources that have been picked and why each source is important.

One could start by asking engineers that are currently practicing the domain. Their knowledge could be settled since the techniques and principles they use are already in use within the domain and have been applied to practical problems with successful results. However, the problem is that each practitioner will have their own approach to a problem. While most knowledge overlaps, there might be significant variations depending on the types of projects each engineer is exposed to.

Knowledge within codified standards or guidelines is probably settled as well. However, it might lag behind the current settled knowledge in the domain since rigorous testing is required for knowledge to be admitted into the documents. They also contain general rules rather than problem-specific ones.

Scientific conferences are strongly interested in the newest pieces of information at the cutting edge of research in a domain. This raises the question of whether scientific conferences are the right place to find out what is "settled".

Many papers in a conference focus on the refinements of already-known techniques. There are usually a number of papers from when the technique is first introduced until it is modified and refined to fit to the domain in question.

This seems to be appropriate for this project which will involve finding methods and techniques most used in the domain over a long period of time to determine hints at settled knowledge. Since they are also abundant and easy to access, they are the most plausible source of knowledge in the domain and the conferences used in the project are the following:

1. Proceedings of the IFAC Symposiums on Control in Transportation Systems, Elsevier Publ. — 1975-2012[21].
2. Proceedings FORMS-FORMAT, Springer-Verlag, 2010-2014[21].
3. Proceedings SAFECOMP, LNCS, Springer, 2005-2014.

The full reasons for selecting these specific conferences are given in Chapter 6. In short, a combination of recommendation by experts and availability were factors in choosing them.

**Step 2: Classification of Knowledge**

Determining what knowledge is to be extracted from sources and what it might look like is the next important step in the project. After collecting any sources of knowledge, it is necessary to know what to look for and in what form it could be found.

Since the selection of sources is narrowed down to research papers and technical reports within the domain, this provides a clue on what sort of information is collected in them, what exactly to look for and how best to extract it.

It is possible to form a classification system from these papers based on the characteristics of each paper. Papers could belong to certain categories depending on the type of knowledge they discuss, how they discuss it and which subdomain is dealt with.

There is important work that describes these categories of knowledge and the classification of bodies knowledge. Below are the recommendations used in this project to categorise knowledge and their sources.

**Vincenti and the Categories of Knowledge**

The difficulty of categorising knowledge is well described in Vincenti's book "What Engineers Know and How They Know It"[45]. He also admits that his own categories are also not entirely exclusive or limited to the categories he has isolated. Vincenti's six categories are:

1. Fundamental Design Concepts
2. Criteria and Specifications
3. Theoretical tools (e.g. formal methods)
4. Quantitative Data
5. Practical Considerations
6. Design Instrumentalities

This gives the project a starting point from which to identify some knowledge, settled or not, within a publication in the domain. By looking for information that fits into these categories, it is possible to isolate most of the design knowledge from the paper.

It also yeilds possible categories that could be used in the future Body of Knowledge to classify this knowledge and organise it in a form accessible to engineers and scientists. In addition to these categories, there might be some new categories that are specific to the domain. This is left for the future work on the Body of Knowledge, as it would need the involvement of experts in the field of railway software engineering to find.

The main advantage of using these categories to start our search is that the "trusted knowledge" is identified to begin with. This means that the appropriate scientific and engineering knowledge is filtered out in the sources used. If a paper contains knowledge in any of these categories, it is safe to say that it contains the appropriate sort of knowledge we need to include in a Body of Knowledge.

Since the procedure of isolating best practices and rules is subjective to a degree, for example in the categories of Practical Considerations and Design Instrumentalities, many experts may disagree on which knowledge they consider to be part of a category or which parts must be left out.

Therefore, for the purposes of this project, formal methods used in the railway domain will be identified and gathered to represent knowledge. These fall under the category of "Theoretical Tools". Because they are easily identifiable and are known parts of engineering design knowledge, formal methods are a straightforward choice with which to start gathering knowledge. They are also easily quantifiable and most can be represented by simple explanations and mathematical expressions.

**Mary Shaw and the Maturity of Discussion in a Domain**

Apart from the technical design knowledge, there is something to be said about the underlying knowledge that can be gleaned from academic papers and other discussions in the field in question.

Mary Shaw's work on "The Coming-of-Age of Software Architecture Research"[41] described what changes occur in the discussion of concepts when a field reaches maturity. The conversation starts with informal discussions among colleagues and progress to products in the market place. The paper develops a maturity model to describe and document natural characteristics of maturing software technology, also referencing the work by Redwine and Riddle[38].

Her paper is important to this project as it is another scale with which to measure the progress or maturity of the domain in question as well as the strength of the sources of data used, that is, formal methods in the railway domain. Using this method, it is easy to see the maturity of the field.

The following measure of the strength of each scientific paper depending on its outcome and method was particularly useful in creating more knowledge categories for the eventual lattice.

There are six phases that the typical software technology will go through to reach maturity.

1. Basic Research
2. Concept Formulation
3. Development and extension
4. Internal Enhancement and exploration
5. Popularization

It is hard to say at a glance which phase formal methods within the domain fit into, but by answering a few questions about the research done in the domain, we can reach a reasonable conclusion.

According to Shaw, the way research is done in the field and with what criteria can help determine the maturity. Here there are three things to consider in software engineering research:

1. Research setting — Research tends to address different classes of problems. So the question to answer here is: What type of problem is being solved

by the paper/article? There are five classes to consider here:

(a) Feasibility — Is there a possibility to accomplish A? Is there a rule/formula governing the behaviour of a particular system/phenomenon?

(b) Characterisation — What are the characteristics of A?

(c) Method/Means — How to accomplish a task or a method that might be superior to current methods.

(d) Generalisation — Is A always true of B?

(e) Selection — Choosing one strategy/method/model over another.

A paper/article can fall into more than one category.

2. Research approaches, methods and products — How did the paper approach this problem? Again there are five possibilities:

   (a) Qualitative or Descriptive Model

   (b) Technique

   (c) System

   (d) Empirical Predictive Model

   (e) Analytical Model

3. Validation techniques — Results of a research activity need to be evaluated as well. It is simply not enough to say that results were achieved without specifying the evaluation technique used. According to Shaw, "Good validation entails not only showing that the specific product of the research satisfies the idealised problem of the research setting, but also that the results help solve the original motivating problem"[41]. The five techniques of validation are:

(a) Persuasion — convincing description of the validation of results.

(b) Implementation — The system/method/idea proposed has been put into a working prototype or real-world situation and works.

(c) Evaluation — Check the final result against a set of criteria. I could also use mathematical evaluation techniques.

(d) Analysis — Rigorous validation through mathematical proofs or derivations along with confirmation by the prediction of the models used.

(e) Experience — Subjective evaluation based on author's experience and observations in actual practice perhaps with some statistical analysis as well.

One would expect settled knowledge to operate at the final three levels of validation techniques, that is, Evaluation, Analysis and Experience. Finally, papers using all validation techniques were found. This might indicate continuting research as well as a possible immaturity in the field.

Using this criteria, it is possible to perform a general and cursory analysis on the development of the field by looking at the sources of knowledge we have collected thus far. This technique has been used in evaluating the maturity of topics such as Abstract Data Types and Software Architecture and the research associated with each field, so it is feasible for our preliminary investigation as well.

This study is indirectly considered as attributes when creating the formal concept lattice, as described in Step 3 of this Chapter.

**Subdomains in the Railway Domain** The entire railway domain is very large and contains many subdomains.

This information is pertinent to the aim of the project. Putting each source of knowledge into a subdomain could provide more information on the frequency of use of software technologies in various parts of the railway domain. It is possible to extract more specific knowledge from this, such as which areas have been researched more on, and if certain areas are at a higher level of maturity than others.

That is, we should apply the general questions we are looking for to specific parts of the domain and then create a more accurate picture of the state of the domain. Perhaps there are even some subdomains that require little to no formal method techniques to function or areas lacking interest or research. All these question can be answered by assigning one or more subdomain to each paper or article reviewed by this project.

The next problem is creating a complete and concise list of subdomains for the railway domain. There have been relatively few studies on creating such a list and only one was found that attempted to put concrete descriptions of some, but not all subdomains of the railway domain. The website that was initially set up by Dines Bjorner for the project TRain is a good place to start[8].

This list seems to be generic, but there was no way of evaluating whether this list is complete or not, especially since it has not been altered in some time and full descriptions of some of the subdomains are missing. To refine this list, a number of experts were consulted on the website called Researchgate. The question thread and the suggestions given by many can be found on the website[25]. The suggestions are summarised below:

1. It strongly depends on the viewpoint and - above all - on the objective of the analysis.
2. There are so many subdomains that it will take too long to list.
3. Other subdomains that are a possiblity are: Signaling (interlocking and headway control), Electric traction (catenary system, substation, etc.), Telecommunication (radio, GSM-R, passenger information, etc.), Engineering structure (bridge, tunnel, etc.), Track (track Equipment, embankment/slope, etc.)

The main point that many researchers and experts in the field pointed out was that the list of subdomains depends on the research you are trying to do and the focus you want on the domain. They also said that from a generalist point of view, the list covers most, if not all generic sections of the railway domain. The rest depend on our specific use.

Since our study is so broad, the following subdomains are a good general list to start with:

1. The Net
2. Timetables
3. Scheduling and Allocation
4. Traffic Monitoring and Control
5. Rolling Stock
6. Passenger Handling
7. Freight Handling

During this project a point was made to add any subdomains that might come up during our survey of papers and articles and add any relevant subdomains that are pertinent to the knowledge we find.

These subdomains were also considered as an addition to attributes in the concept lattice to be created. This is pertinent information as it provides information on what formal methods are used in which subdomain. There might be an underlying pattern. It can also tell us the aspects of the railway domain that use the most and the least formal methods. For the full list of subdomains and their reasonings, see Chapter 5.

**Adding Year Published as an Attribute**

One characteristic is the year the paper was published. This tells us how old the knowledge in the paper is. There may or may not be a pattern to be discovered in this information, but at the very least it might correlate to how long a particular method is used in the railway domain. It is also possible this may cause extra noise in the data which will require the use of noise-removing techniques on the lattice to take out.

As we have seen in Chapter 3, knowledge existing over a length of time has a greater chance to be settled than knowledge that is much more recent. Methods that have been studied over time show refinement in their use and therefore, like in the case with most engineering knowledge, are honed until they become standard practice.

Of course these are not the only criteria as newer methods are better at times, but the theory is that if they are indeed settled, the newer methods will be used more frequently and eventually find their way into bodies of knowledge. The best that can be done at the moment is deal with the knowledge known at the moment instead of trying to predict the future of newer methods. This makes the body of knowledge lag behind current research, which is evident in almost all areas of engineering knowledge as well.

**Why Formal Methods?**

Within this project, the focus is on the use of formal methods as an aspect of settled knowledge in the railway domain. Formal methods are an indication of mathematical certainty in solving a problem within the domain. This means that they have a high chance to be settled knowledge, since the understanding of the problem and its solution has become measurable and not just descriptive. This makes the knowledge easy to define unequivocally.

Formal methods are also easy to identify — either they are used or not used. This eases the problem of discovery of settled knowledge. However, this means that some settled knowledge is not recorded in this project. To identify most other aspects of settled knowledge, such as practical criteria or tacit knowledge, implicit information would have to be extracted from the knowledge base and after that, it would still be necessary to confirm with a number of experts whether they agree on this data.

It is the purpose of this project(as outlined in Chapter 1) to get a start on this collection (if indeed any knowledge is settled) and the most straightforward way to do so in the context of software engineering is with formal methods. Formal Concept Analysis gives us a way to identify long standing patterns within the data collected.

Finally, the classification of each paper included three separate areas:

1. By Year the paper was published.
2. By Subdomain the paper pertains to.
3. List of keywords that first and foremost includes any formal methods used. The paper may also be using any software engineering or computer science technique worth noting. Includes any formal languages used for describing a possible model of a system.

The last section was more subjective and the technique to classify papers was improved as more and more papers were examined.

**Step 3: Populate Formal Context**

After extracting where each paper falls in these categories, the Formal Concept Analysis tool called ConExp or Concept Explorer[37] is used to insert this data first into a cross table (refer to Figure 8.13 in Chapter 4) and then use the Concept Explorer tool to turn this into a lattice (Figure 4.2 in Chapter 4).

Part of the resulting cross table can be seen in Figure 7.3. The extent of the context was the various papers from which this information was gleaned. The intent of the context was the characteristics of each paper, i.e. the relevant subdomain, the formal methods used, the year published and any other keywords that might be useful for extraction of knowledge (e.g. any methods that might not be formal yet, but are widely used in software design and development).

Some special keywords were included into this context to make calculations and interpretation easier. They are mentioned below along with the reason for their inclusion.

| A | B | C | D | E | F | G | H | I | J | K | L | M | G |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | The Net | Timetables | Schedulin... | Traffic Mo... | Rolling St... | Passenge... | Freight H... | LinearTim... | UML | PetriNets | MILP | Train Ope... | |
| 12_03 | | X | | | | | | | | | X | | |
| 12_04 | X | | | | | | | | | X | | | |
| 12_16 | | | X | | X | | X | | | | X | | |
| 12_34 | | | | X | | | | | | X | | | |
| 12_35 | | | | X | | | | | | | | | |
| 12_36 | | | | X | | | | | | | | | |
| 12_38 | | | | X | | | | | X | | | | |
| 09_01 | | | | X | | | | | | | | | |
| 09_13 | X | | | | X | | | X | | | | | |
| 09_97 | | | X | X | | | | | X | X | | | |
| 06_251 | | | | | | | | | | | | X | |
| 06_331 | | | | | | | X | | | | | | |
| 06_261 | | | | X | | | | | | | | | |
| 06_337 | | | X | | | | | | | | | | |
| 06_343 | | | X | | | | | | | | X | | |
| 06_349 | | | | X | | | | | | | | | |
| 03_07 | | | | X | | | | | | X | | | |
| 03_06 | | | | | | | X | | | | | | |
| 03_05 | | | X | | | | | | | | | | |
| 03_04 | | | | X | | | | | | | | | |
| 03_03 | X | | | X | | | | | | | | | |
| 03_02 | | | | X | | | | | | | | | |
| 03_01 | X | | | | | | | | | | | | |
| 06_245 | | X | | | | | | | | X | | | |
| 00_1 | | | X | | | | | | | | | | |
| 00_2 | | | | X | | | | | | | | | |
| 00_3 | | | | X | | | | | | | | | |
| 00_4 | X | X | X | X | X | X | X | | | | | | |
| 00_5 | | | | X | | | | | | | | | |
| 00_6 | X | | | | | | | | | X | | | |
| 00_8 | | | | X | | | | | | | | | |
| 00_12 | X | | | X | | | | | | | | | |

Figure 7.3: Partial extract of the resulting cross table using data collected from sources of railway domain knowledge

1. **Scholarly Discussion** — This included papers where a discussion of either software design techniques in railway software or the use of formal techniques and their importance took place, but no actual formal model was explicitly pointed out. From the perspective of Mary Shaw[41], this would be a persuasion validation technique.

   At a first glance this may look as if it is not even relevant within the context of a Body of Knowledge. However, these discussions can help with the identification of the implied "tacit knowledge" as discussed by Vincenti[45] where practical considerations or an engineering way of thinking might be applied.

   This also gives us an interesting number of articles that still discuss formal methods in a conceptual light. It is possible that if there is too much discussion and little concrete models being described, that the field might still be immature for a Body of knowledge.

2. **Descriptive Model** — This keyword (and the next two) are inspired by Mary Shaw's work[41] and provide a way to better describe the type of modelling each paper or article had to offer. The descriptive model is an attribute for any paper that described a text-based description of a model, which could potentially be formalised but has not been yet attempted. This model is of the system or a part of the system or the behaviour of either that does not directly include a software development or design technique.

3. **Physical/Mathematical Model** — later simplified to just **Mathematical Models**. This is an attribute inclusive of all mathematical models of the railway system using physical and mechanical laws that do not directly reference a formal method in the software development area. This could be the modelling of a specific behaviour or characteristic of a system or aspect of a system, using mathematical tools and formulae. This attribute would provide more information about the type of modelling being developed within the railway domain.

4. **Analytical Model** — This is the last type of model being described and includes any mathematical model that aims to analyse a railway system or part of a system without directly describing the system at hand. This model's output would be to gauge an existing model's effectiveness/efficiency. This attribute would provide more information about the

type of modelling being developed within the railway domain.

5. **Languages** — An attribute for any paper describing the use of a specific language in the railway domain. This attribute was created so that even if there are not any specific languages that are being stable, a consistency of the use of languages can indicate the implementation and development of code within research.

6. **Failure Analysis** — This involves a range of different topics that are qualitative via semi-quantitative to quantitative risk analysis methods such as Failure Modes and Effects Analysis (FMEA), Fault Tree Analysis (FTA), Hazard and Operability Analysis (HAZOP), Preliminary Hazard Analysis (PHA), Event Tree Analysis, and so on. This is all combined together under the larger heading of Failure Analysis.

7. **Artificial Intelligence** - This is an attribute for any artificial intelligence techniques used in the development or modelling of railway systems. Includes areas such as neural networks, genetic algorithms, fuzzy logic and so on.

A major reason for some of these keywords are the general trends that the project was aimed at identifying. Grouping smaller techniques under a broad heading identifies the larger problem being solved by modelling techniques rather than risking that that information is perhaps lost in the vastness of the lattice structure after reviewing a large number of papers and articles.

After all this information was entered into the cross table, the Concept Explorer tool was able to build a lattice based on the information and the complete lattice can be seen in Chapter 8.

**Step 4: Calculate The Concept Stability Indices**

Calculating stability for each concept within the concept lattice is the next step. This gives us a method to identify noise within the structure of the lattice and have it removed. It also tells us how independent the intent of a concept is from its extent.

We expect the resulting lattice to give us the best representation of the structure of the knowledge. In order to do this we need to separate the important information and remove noise. The stability index gives us a concept-by-concept measure of the importance of a set of objects and attributes to the overall structure of the lattice and consequently the structure of the knowledge it represents.

The method for calculating the stability index was described in Kuznetsov's paper "On stability of a formal concept"[26]. Stability measures the dependence of a concept intent on objects of the concept extent.

The basics of formal concept analysis are already described in Chapter 4 and adding to that, a straightforward definition of stability from the paper by Buzmakov, Kuznetsov and Napoli[11] is as follows.

For a context $K = (G, M, I)$ and a concept $C = (A, B)$

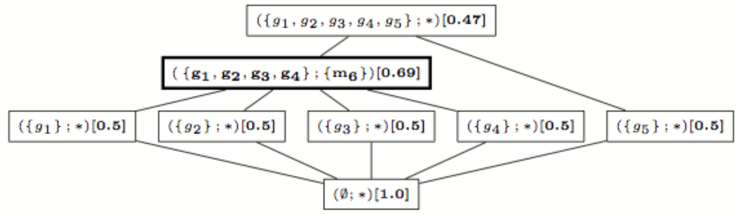
$$Stab(c) = \frac{\mid \{s \in \wp(Ext(c)) \mid s' = Int(c)\} \mid}{2^{|Ext(c)|}}$$

i.e. the relative number of subsets of the concept extent (denoted by $Ext(c)$), whose description(the result of applying I) is equal to the concept intent (denoted by Int(c)) where $\wp(P)$ is the power set of P.

**Example[11]**

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | $m_6$ |
|---|---|---|---|---|---|---|
| $g_1$ | x | | | | | x |
| $g_2$ | | x | | | | x |
| $g_3$ | | | x | | | x |
| $g_4$ | | | | x | | x |
| $g_5$ | | | | | x | |

(a)




(b)

(a) A toy formal context.

(b) Concept Lattice with corresponding stability indexes.

Figure 7.4: A toy example in calculating stability[11]

This example[11] refers to Figure 7.4 where a cross table and a corresponding concept lattice are shown side by side. Some intents are intentionally left out.

The extent of the concept in bold is $Ext(c) = \{g_1, g_2, g_3, g_4\}$. Therefore its power set contains $2^4$ elements. The intent of the overall concept c is $Int(c) = \{m_6\}$.

The intent of the subsets of the extent are:

$Int(\{\emptyset\}) = \{\}$

$Int(\{g_1\}) = \{m_1, m_6\}$

$Int(\{g_2\}) = \{m_2, m_6\}$

$Int(\{g_3\}) = \{m_3, m_6\}$

$Int(\{g_4\}) = \{m_4, m_6\}$

$Int(\{g_1, g_2\}) = \{m_6\}$

$Int(\{g_1, g_3\}) = \{m_6\}$

$Int(\{g_1, g_4\}) = \{m_6\}$

$Int(\{g_2, g_3\}) = \{m_6\}$

$Int(\{g_2, g_4\}) = \{m_6\}$

$Int(\{g_3, g_4\}) = \{m_6\}$

$Int(\{g_1, g_2, g_3\}) = \{m_6\}$

$Int(\{g_1, g_2, g_4\}) = \{m_6\}$

$Int(\{g_1, g_3, g_4\}) = \{m_6\}$

$Int(\{g_2, g_3, g_4\}) = \{m_6\}$

$Int(\{g_1, g_2, g_3, g_4\}) = \{m_6\}$

From the above list, only the first five subsets have an intent that is not $\{m_6\}$.Therefore, the stability index of the concept c is:

$$Stab(c) = \frac{2^4 - 5}{2^4} = 0.69$$

In this way, the stability of each concept in the lattice and consequently in the context is calculated. The final results can be found in Chapter 8.

In reference to the data, the stability indices show how likely we are to observe a particular formal method or subdomain or time period even if we ignore some papers. Stability does not only provide noise-resistance. A stable concept does not collapse when certain papers are removed from the context — that is, the concept does not merge with a different concept or divide up into smaller concepts.

We know settled knowledge to be integral to the body of knowledge in a domain, therefore we expect settled knowledge to be incorporated in stable concepts rather than unstable ones.

As a general rule, a concept with a large extent is usually stable and concepts that are small (with small extents) are usually unstable and are much more common than larger ones. So this is what is expected to be seen in the lattice.

**Step 5: Prune Lattice**

As with all data collected, we expect some noise in this data because of the apparent randomness of the selection process of the data. Some conferences were recommended and some were picked due to accessibility. There is no logical reason for not picking an entirely different set of data if there were no external factors such as time or resources.

Noise might also be as a result of possible mistakes in classification or characterisation that might occur due to human involvement. Lastly, the nature of the data means that there are avenues of research that have been unsuccessfully attempted. So by default, there is already redundancy within the data that helps to confirm the futility of some approaches.

It is also not necessary to include all of the data collected. If a context and its concepts are stable, the same lattice and the same relations could still be seen, even with a completely different subset of data. This is why it does not matter if a certain percentage of input data is discarded. The reasoning being that even if part of the data that makes up a stable context is removed, concepts are still stable with this smaller subset — which is a characteristic of stable concepts.

As stated before, stability measures the dependence of a concept intent on objects of the concept extent. That implies that if we omit some papers, the concept remains visible in the data.

An important subset of data can be extracted by selecting concepts with the highest stability indices, for example, limiting the concepts within an allowed stability index range (only taking the most stable concepts) so that the lattice remains small and informative. This subset provides the concepts that contain the most domain-specific relevant groups. But before this lattice can be constructed, it is important to decide what this threshold value could be.

A threshold stability index is selected and concepts that have a stability index higher than or equal to the threshold are considered without noise. Concepts with a stability index that is lower than the threshold are pruned away from the lattice. This corresponds to the idea that knowledge which is not settled appears in unstable concepts and can be considered noise.

There are also many alternative pruning techniques for concept lattices, resulting in a wide array of lattices to suit different purposes[40, 44, 22]. These techniques have not been used here. This is because these different techniques are specific to some characteristic of the data being analysed and the use of these methods are specific to a situation or data type. Since no underlying characteristics of the data in the lattice are known yet, the standard method for stability index calculation and pruning is used in this project.

Two techniques described in the paper by Kuznetsov and Ignatov[27] which involves either limiting the size of the extent (taking an "iceberg" of the total lattice containing all concepts with extents larger than a threshold) or taking the most stable concepts(concepts with stability indices larger than a threshold) are possibilities of how to prune the lattice.

The first approach provides very large groups of extents over the whole lattice. In this project, it means that it provides attributes that are common to "everybody". Looking at the body of knowledge, there are not many techniques common to the entire domain. We are also looking for more subtle stable intents within certain subdomains of the railway, so this method is too crude for our purposes. Also, a quick construction of the resulting lattice tells us that these common intents are also the most stable, so an application of a less brute force method, like using the stability indices is viable and might provide more information.

The second approach provides characteristically significant groups as well. They might be smaller, but their intents are more significant to the body of extents. This will be the approach used in this project, and is also recommended by a number of papers analysing similar bodies of data[26, 11, 40, 27].

**Selecting a Stability Threshold**

Many papers that discuss the selection of different stability threshold values and the reasoning behind them, are dependent mostly on the type of data in question and the intended data analysis.

After pruning based on stability indices, it is possible that the resulting set of concepts do not form a complete lattice. The paper by Roth, Obiedkov and Kourie[40] explains how a stability threshold value can be selected if further analysis is required on a constructed lattice. Unstable concepts can even be merged with their subconcepts to keep the lattice intact, but it does not specify how to go about selecting the concepts that can be merged. A complete lattice is not a main priority for the aim of this project because a relation between the stable concepts is not what we are trying to establish (yet). It is much more pertinent just to single out the stable concepts instead.

Firstly, consider using the approach of mean, median and mode as the simplest and most straightforward way of choosing a stability index as the threshold. Tempting though it is, to choose the stability threshold in this way provides no logical or scientific reasoning behind our choice beyond a statistical one. It also does not make use of the Formal Concept Analysis tools or even let the data contribute to the choice.

The data itself can help us fine tune this choice to our purpose. Many papers look at selecting this threshold value by turning to the specifics of the data involved[40, 27].

In our case, the question is "What percentage of the data is included in the lattice at each stability index?". Putting it another way, for each value of stability index, what percentage of data is included in the data set? That is, how much knowledge are we keeping within the lattice at each stability index value? We want to reduce noise, but we do not want to remove too much of our data as well. This fine balance can be achieved using a mathematical approach and plotting the results.

A graph of the amount of data included (in the lattice) is plotted against a variety of stability thresholds. Looking at this function we can hopefully see some good candidates for thresholds. If a gradual decrease is observed (perhaps a linear function), then this data has one of two results: either it is not useful in the indication of a threshold value or there is a negligible amount of noise in the data.

If a constant amount of data was excluded at each stability index, this means that there are lots of small(i.e. with a small extent) concepts with very different stability indices scattered over the stability index axis. This would also mean that either there is negligible noise in the data or that this approach does not give us any more information on a stability threshold. We would then need to include other techniques.

If there was an observable large drop at a particular stability threshold, it means that there were lots of small concepts (or a very large concept) at a particular stability index. If this is a lower stability index (generally $< 0.5$) it is more likely to be a number of small concepts. This can be confirmed by manually looking at the lattice. Generally speaking, removing a large concept would not be desirable. Removing lots of little concepts with a lower stability index value is more acceptable since the data in these concepts is more likely to be noise.

But before removing the small concepts immediately, they should also be examined. If they are all related, then combining them into a larger concept should be considered so that the data can be retained and brought closer to the top of the "iceberg". This would show a need to refine the classification techniques used in this project.

Repeating this process and refining the techniques (if and where necessary) should provide a reasonable stability threshold (close to the statistical value of 0.5)

**Step 6: Extract Stable Attributes**

After pruning the lattice, the remaining concepts all have stability indices higher than or equal to the stability threshold. The next step is to isolate the attributes of these concepts which are the stable keywords, formal concepts, years and subdomains in the final lattice.

If the final list of formal methods is not empty, then it means there is some stability in the domain with respect to formal methods. This would mean that there is a sizeable amount of research into these stable methods over the period of time that the data has been collected. And these are the likely signs of settled knowledge within the domain with the focus on formal methods.

There might also be some keywords that are not formal methods but still give us an insight into what software engineering techniques or general approaches are commonly used in the modelling and development of software in the railway domain.

If there are some subdomains or years that are also stable, then this means that the majority of research happened over these years and in the stable subdomains. This also shows the emphasis of formal methods on these subdomains in the years.

From these results, if there is a final lattice that can be constructed (although this is not necessary), there might also be some general relationships between the stable concepts that can be extracted. It is also possible that the final lattice might not provide much insight if every concept is on a similar level.

From the results of the last step in the method, it will be possible to say something concrete about our original hypothesis, as described in the Introduction chapter (Chapter 1). It will also allow a complete conclusion to be created based on our assumptions and method strengths.

# Chapter 8

# Findings

The following chapter details and discusses the results obtained from the Formal Concept Analysis exercise on these results. The Java program, *Analysis.java* was used to analyse some of the results and can be found in a public repository[24].

## 8.1 Stability Calculations

Calculation of stability is important in streamlining a lattice and drawing meaningful connections between the concepts. Streamlining is done by way of identifying and keeping the most relevant concepts, and pruning the irrelevant concepts. Concepts with a high stability indicate that their connections will not be altered by changes in a small subset of the data used to build the lattice. Conversely, concepts with lower stability are prone to more change should a small fraction of the data be different.

The method for the calculation of stability can be found in Chapter 7. In the next few subsections are the results of calculating stability on various concepts of the context, categorised by sub-domain. Each subdomain contains its own lattice and a table with stability indices, attributes and number of objects within the subdomain concerned. Each table is arranged in descending order of stability. For the sake of being concise, stability indices as a result of only one object have been left out. They are also insignificant on a conceptual level because our objective is to achieve a lattice that is stable even with the loss of a small subset of input data(See Chapter 7).

Note that the concepts listed might not directly correlate to the lattices shown as the calculations of stability are done using the entire context. The data is also knowingly not omitted since that would jeopardise the relevance of some concepts. Therefore, the lattices for each subdomain are intended to give a general idea of the size of the subdomain and the important aspects of it. They do not hold much relevance for the calculations. Concepts that overlap many domains are presented in the last subsection labelled Complete Lattice.

### 8.1.1 The Net

| Concept # | $\|A\| = \|\text{extent}\|$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 28 | The Net | 0.999011229723692 |
| 2 | 9 | The Net, 2000 | 0.82421875 |
| 3 | 8 | The Net, Scholarly Discussion | 0.77734375 |
| 4 | 5 | The Net, 2000, Scholarly Discussion | 0.75 |
| 5 | 4 | The Net, Petri Nets | 0.625 |
| 6 | 4 | The Net, Mathematical Model | 0.5625 |
| 7 | 6 | The Net, 2014 | 0.484375 |
| 8 | 4 | The Net, Domain Specific Formalisms | 0.375 |

| | | | |
|---|---|---|---|
| 9 | 3 | The Net, 2012 | 0.375 |
| 10 | 2 | The Net, 1995 | 0.25 |
| 11 | 2 | The Net, 1997 | 0.25 |
| 12 | 2 | The Net, Analytical Model | 0.25 |
| 13 | 2 | The Net, Descriptive Model | 0.25 |
| 14 | 2 | The Net, Domain Specific Formalisms, RAISE | 0.25 |
| 15 | 2 | The Net, 2003 | 0.25 |
| 16 | 2 | The Net, Petri Nets, 2000 | 0.25 |

Table 8.1: Stability calculations for the concepts in The Net subdomain

In the subdomain of The Net, as shown in Table 8.1, petri nets and mathematical models are the two formal methods that were in the intent of concepts with a stability of greater than 0.5. Other concepts shown above have stability indices which may increase in significance with a different data set. Since so many of these concepts have two objects each, it is difficult to identify which concepts might be more relevant than others.

### 8.1.2 Timetables

| Concept # | $\lvert A\rvert = \lvert\text{extent}\rvert$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 14 | Timetables | 0.74981689453125 |
| 2 | 2 | Timetables, 2012 | 0.25 |

Table 8.2: Stability calculations for the concepts in Timetables subdomain

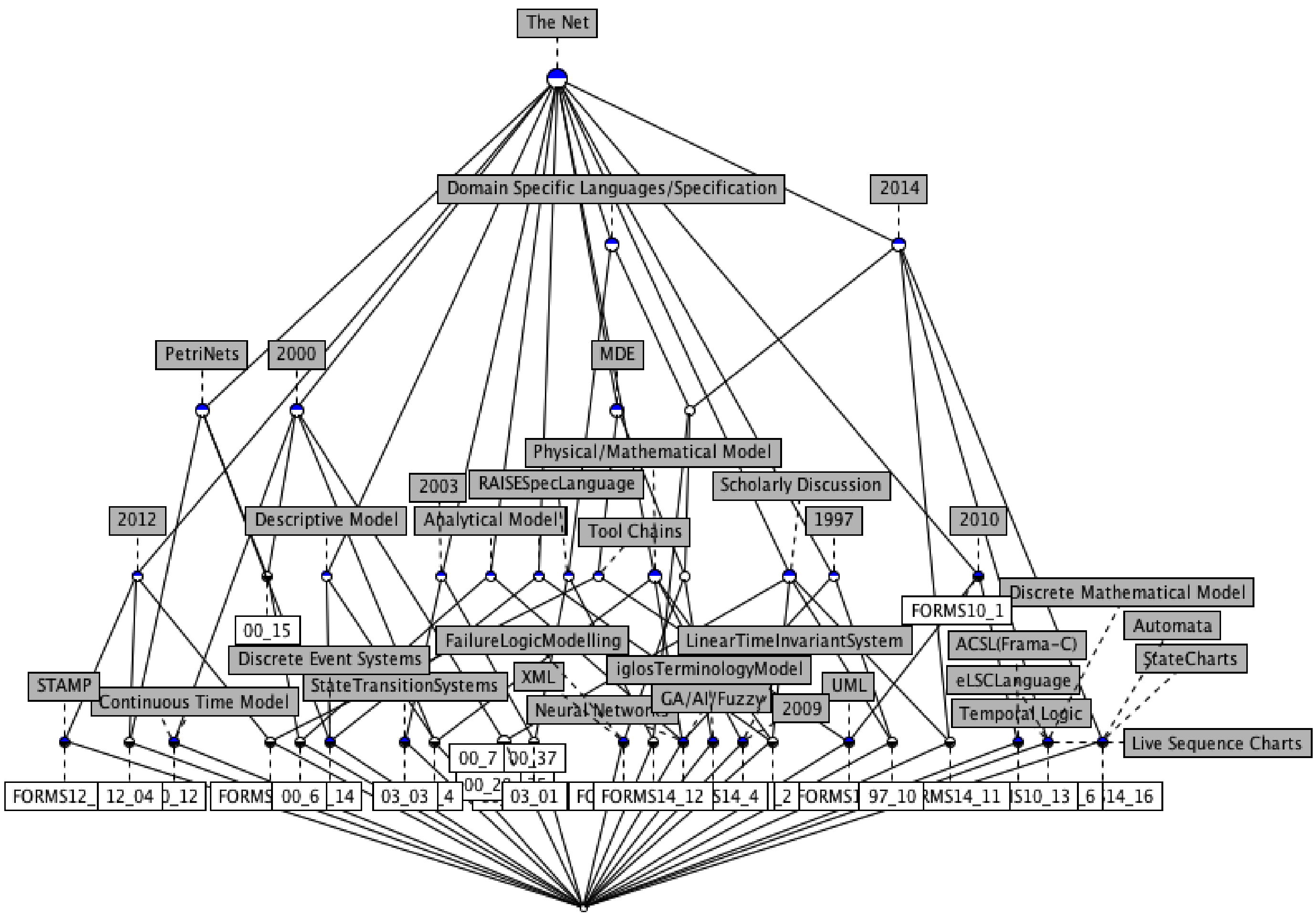



Figure 8.5: Lattice representing The Net subdomain

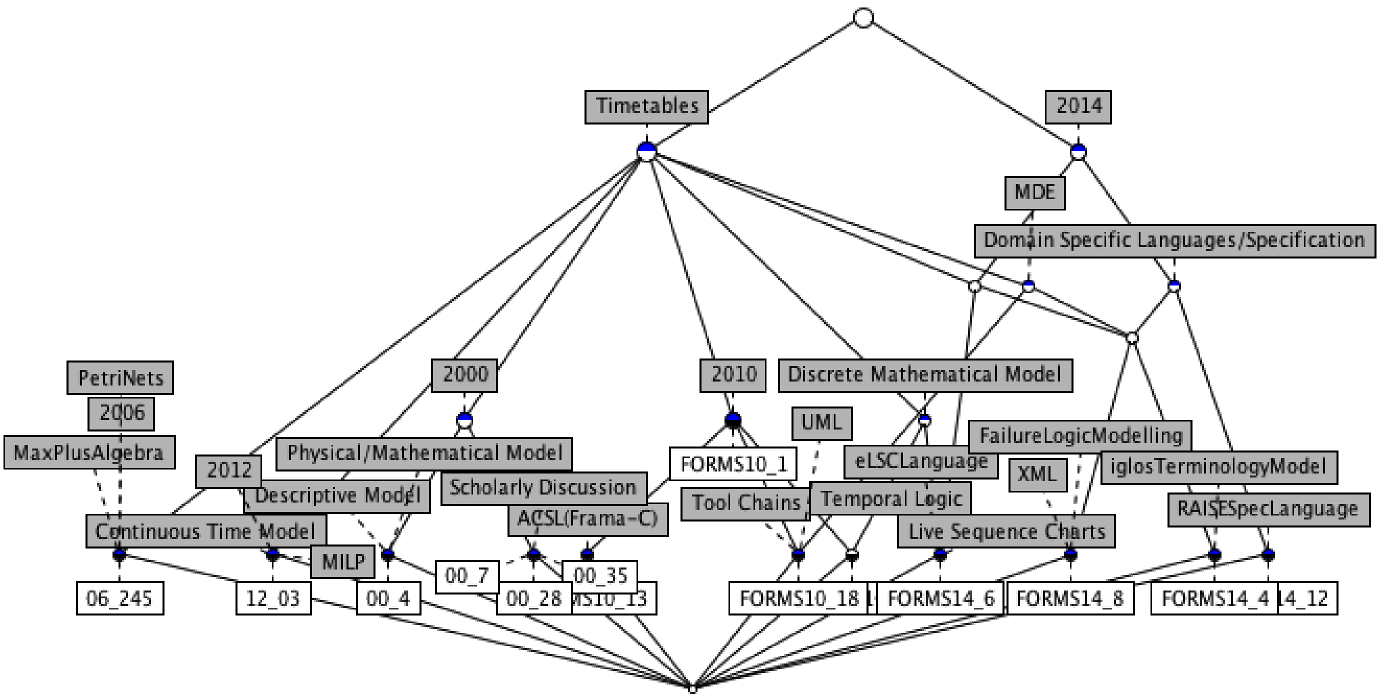



Figure 8.6: Lattice representing The Timetables subdomain

Table 8.2 contains the stability indices for the Timetables domain. The reasons for very few entries in this table is because this is a relatively small domain, most concepts either consist of one object or they contain intents which overlap a number of subdomains. These are documented in the global lattice table at the end of this section. The stability indices of concepts not shown in this table are unlikely to increase in size even if a different dataset is chosen by virtue of only one object containing them.

### 8.1.3 Scheduling and Allocation

| Concept # | $\|A\| = \|\text{extent}\|$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 20 | Scheduling and Alloc | 0.982369422912598 |

| | | | |
|---|---|---|---|
| 2 | 4 | Scheduling and Alloc, AI/GA/Fuzzy | 0.6875 |
| 3 | 2 | Scheduling and Alloc, 2006 | 0.5 |
| 4 | 5 | Scheduling and Alloc, Scholarly Discussion | 0.46875 |
| 5 | 4 | Scheduling and Alloc, 2000 | 0.46875 |
| 6 | 4 | Scheduling and Alloc, 2014 | 0.4375 |
| 7 | 2 | Scheduling and Alloc, MILP, Mathematical Model | 0.25 |

Table 8.3: Stability calculations for the concepts in Scheduling and Allocation subdomain

Concepts that only contain one object have been omitted. This domain also overlaps with many others such as Traffic Monitoring and Control. The combined concepts can be seen in the last subsection in this section.

### 8.1.4 Traffic Monitoring and Control

| Concept # | $\|A\| = \|\text{extent}\|$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 100 | Traffic M&C | 1.0 |
| 2 | 29 | Traffic M&C, 2000 | 0.999999295920134 |
| 3 | 24 | Traffic M&C, Scholarly Discussion | 0.999985098838806 |
| 4 | 15 | Traffic M&C, 2010 | 0.998382568359375 |
| 5 | 14 | Traffic M&C, Petri Nets | 0.998291015625 |
| 6 | 12 | Traffic M&C, 2012 | 0.995361328125 |

| | | | |
|---|---|---|---|
| 7 | 14 | Traffic M&C, 1995 | 0.99530029296875 |
| 8 | 14 | Traffic M&C, 2014 | 0.99456787109375 |
| 9 | 13 | Traffic M&C, Physical/Mathematical Model | 0.9912109375 |
| 10 | 11 | Traffic M&C, Domain Specific Formalisms | 0.97412109375 |
| 12 | 12 | Traffic M&C, Analytical Model | 0.965576171875 |
| 13 | 6 | Traffic M&C, Scholarly Discussion, 1995 | 0.9375 |
| 14 | 7 | Traffic M&C, 2000, Scholarly Discussion | 0.9296875 |
| 15 | 7 | Traffic M&C, 1997 | 0.9140625 |
| 16 | 7 | Traffic M&C, 2000, Analytical Model | 0.9140625 |
| 17 | 4 | Traffic M&C, Petri Nets, 2014 | 0.875 |
| 18 | 6 | Traffic M&C, Virtual Train Sets/Automatic Guided | 0.859375 |
| 19 | 6 | Traffic M&C, GA/AI/Fuzzy | 0.859375 |
| 20 | 5 | Traffic M&C, 2000, Mathematical Model | 0.84375 |
| 21 | 5 | Traffic M&C, Mathematical Model, 2010 | 0.78125 |
| 22 | 4 | Traffic M&C, Scholarly Discussion, 2014 | 0.75 |

| | | | |
|---|---|---|---|
| 23 | 2 | Traffic M&C, 2000, Markov Models, Analytical Model | 0.75 |
| 24 | 5 | Traffic M&C, FMEA | 0.75 |
| 25 | 3 | Traffic M&C, 2012, Scholarly Discussion | 0.75 |
| 26 | 2 | Traffic M&C, GA/AI/Fuzzy, 2000 | 0.75 |
| 27 | 4 | Traffic M&C, 2003 | 0.6875 |
| 28 | 5 | Traffic M&C, Domain Specific Formalisms, 2014 | 0.65625 |
| 29 | 3 | Traffic M&C, Mathematical Model, 1995 | 0.625 |
| 30 | 4 | Traffic M&C, Formal Risk Analysis | 0.625 |
| 31 | 4 | Traffic M&C, Domain Specific Formalisms, RAISE | 0.625 |
| 32 | 2 | Traffic M&C, 2000, FTA | 0.5 |
| 33 | 2 | Traffic M&C, Formal Risk Analysis, Mathematical Model, 2010 | 0.5 |
| 34 | 2 | Traffic M&C, Domain Specific Formalisms, 2000, Analytical Model | 0.5 |
| 35 | 3 | Traffic M&C, 2006 | 0.5 |
| 36 | 2 | Traffic M&C, Agents, 2000 | 0.5 |
| 37 | 2 | Traffic M&C, GA/AI/Fuzzy, 1995 | 0.5 |
| 38 | 2 | Traffic M&C, Petri Nets, 1997 | 0.5 |

| | | | |
|---|---|---|---|
| 39 | 2 | Traffic M&C, Petri Nets, 2012 | 0.5 |
| 40 | 2 | Traffic M&C, UML, 2012 | 0.5 |
| 41 | 4 | Traffic M&C, Descriptive Model | 0.4375 |
| 42 | 5 | Traffic M&C, MDE | 0.375 |
| 43 | 3 | Traffic M&C, MDE, 2014 | 0.375 |
| 44 | 3 | Traffic M&C, Markov Models, Analytical Model | 0.375 |
| 45 | 3 | Traffic M&C, 2000, Descriptive Model | 0.375 |
| 46 | 3 | Traffic M&C, Continuous Time Model | 0.375 |
| 47 | 3 | Traffic M&C, Discrete Time Model, Analytical Model | 0.375 |
| 48 | 3 | Traffic M&C, Discrete Time Model, 2000 | 0.375 |
| 49 | 3 | Traffic M&C, Domain Specific Formalisms, 2000 | 0.375 |
| 50 | 2 | Traffic M&C, Model Checking | 0.25 |
| 51 | 3 | Traffic M&C, Discrete Mathematical Model | 0.25 |
| 52 | 2 | Traffic M&C, Discrete Mathematical Model, 2014 | 0.25 |
| 53 | 2 | Traffic M&C, STAMP | 0.25 |
| 54 | 2 | Traffic M&C, Discrete Event Systems | 0.25 |
| 55 | 2 | Traffic M&C, Analytical Model, 2010 | 0.25 |

| | | | |
|---|---|---|---|
| 56 | 2 | Traffic M&C, Analytical Model, 1997 | 0.25 |
| 57 | 2 | Traffic M&C, Mathematical Model, Scholarly Discussion | 0.25 |
| 58 | 2 | Traffic M&C, Virtual Train Sets/Automatic Guided, 1995 | 0.25 |
| 59 | 2 | Traffic M&C, Virtual Train Sets/Automatic Guided, 2000 | 0.25 |
| 60 | 2 | Traffic M&C, FMEA, 2000 | 0.25 |
| 61 | 2 | Traffic M&C, Continuous Time Model, 2000 | 0.25 |
| 62 | 4 | Traffic M&C, Discrete Time Model | 0.25 |
| 63 | 2 | Traffic M&C, Discrete Time Model, 2000, Analytical Model | 0.25 |
| 64 | 2 | Traffic M&C, Domain Specific Formalisms, 2010 | 0.25 |
| 65 | 2 | Traffic M&C, Domain Specific Formalisms, RAISE, 2014 | 0.25 |
| 66 | 2 | Traffic M&C, 2009 | 0.25 |
| 67 | 3 | Traffic M&C, Agents | 0.25 |
| 68 | 2 | Traffic M&C, Agents, FMEA | 0.25 |

Table 8.4: Stability calculations for the concepts in Traffic Monitoring and Control subdomain

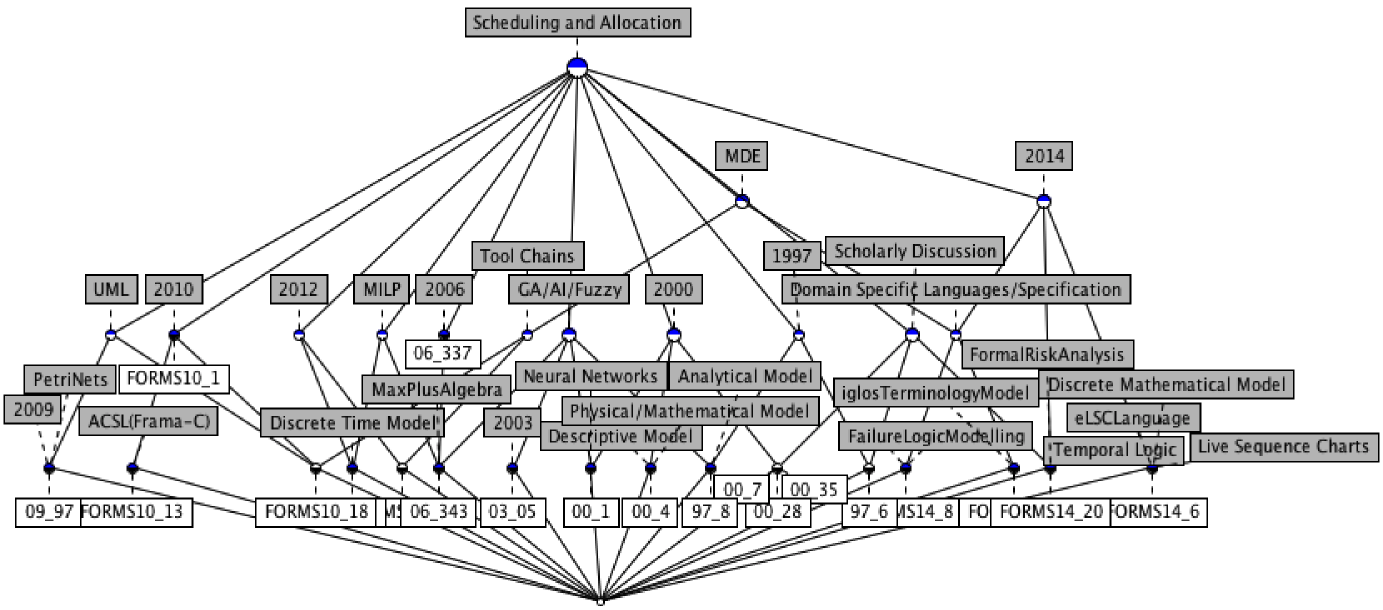



Figure 8.7: Lattice representing The Scheduling and Allocation sub-domain

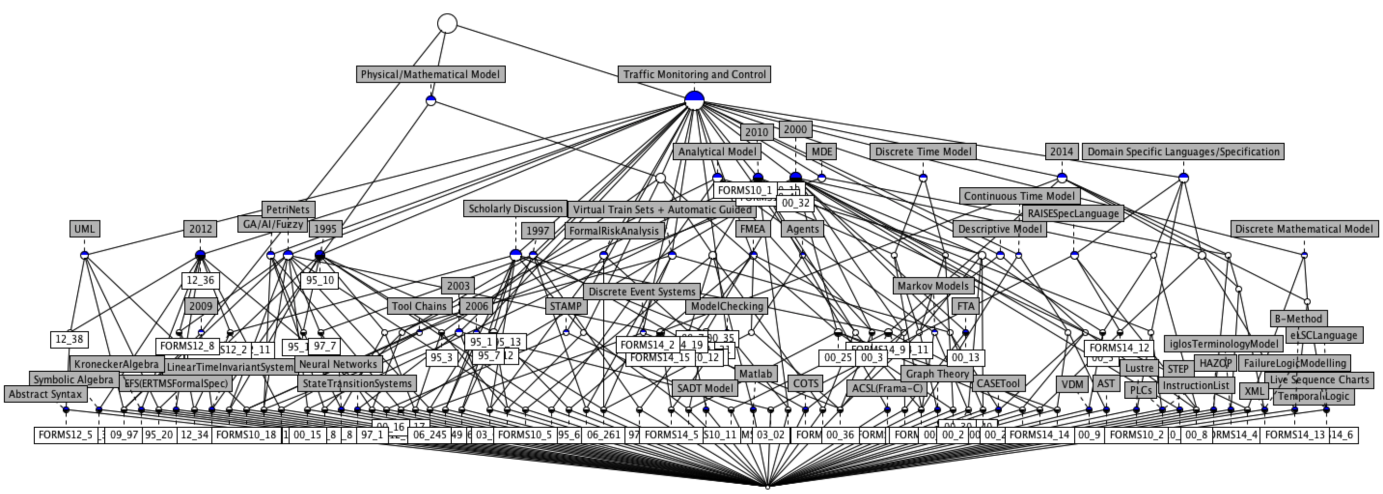



Figure 8.8: Lattice representing The Traffic Monitoring and Control sub-domain

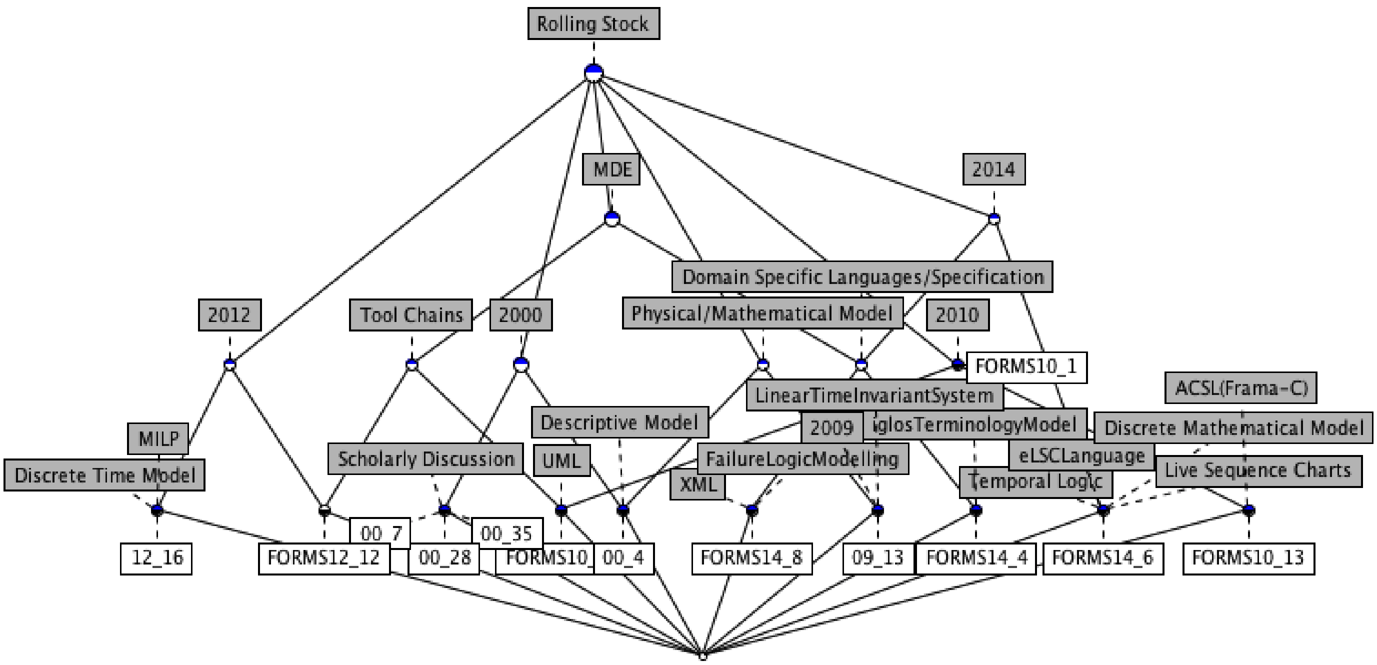



Figure 8.9: Lattice representing Rolling Stock sub-domain

### 8.1.5 Rolling Stock

| Concept # | $\lvert A\rvert = \lvert\text{extent}\rvert$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 13 | Rolling Stock | 0.25 |

Table 8.5: Stability calculations for the concepts in Rolling Stock subdomain

The Rolling Stock domain contains only one concept that completely belongs to itself. All other concepts within the domain are reliant on multiple subdomains of Railway Systems and are documented in the Complete Lattice section of this chapter.

### 8.1.6 Passenger Handling

No concepts are completely encompassed by this small subdomain and all overlapping concepts can be found in the Complete Lattice section later on in this chapter.

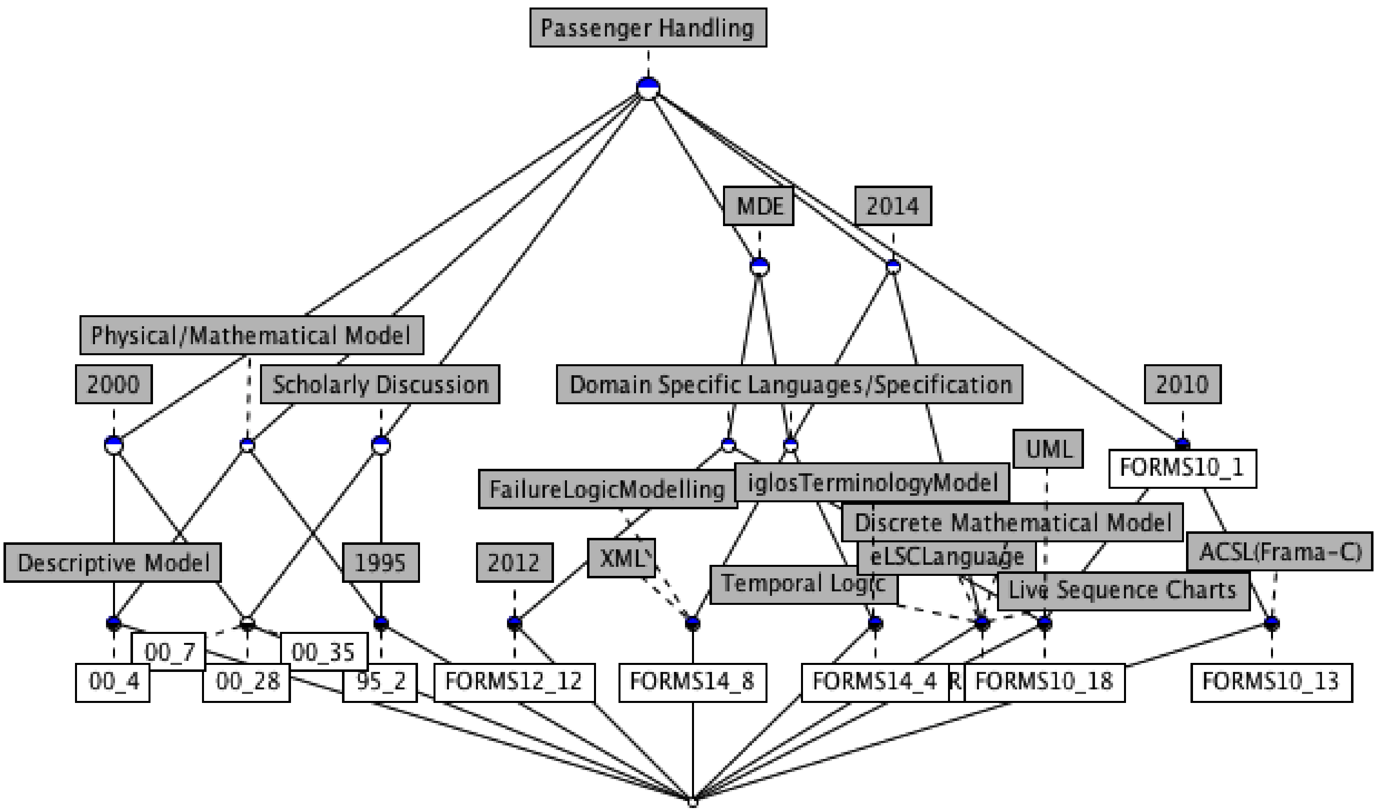



Figure 8.10: Lattice representing The Passenger Handling sub-domain

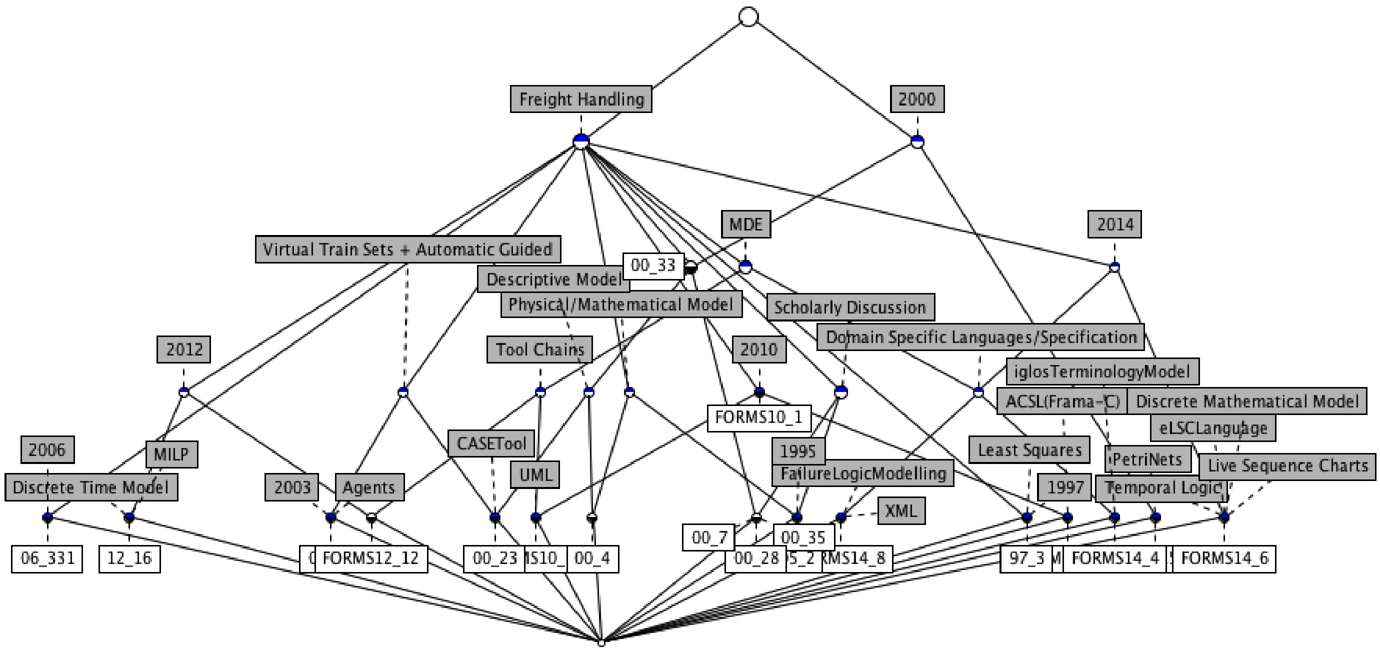



Figure 8.11: Lattice representing The Freight Handling sub-domain

### 8.1.7 Freight Handling

| Concept # | $\|A\| = \|\text{extent}\|$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 19 | Freight Handling | 0.980278015136719 |
| 2 | 7 | Freight Handling, 2000 | 0.7421875 |
| 3 | 2 | Freight Handling, Virtual Train Sets/Automatic Guided | 0.25 |

Table 8.6: Stability calculations for the concepts in Freight Handling subdomain

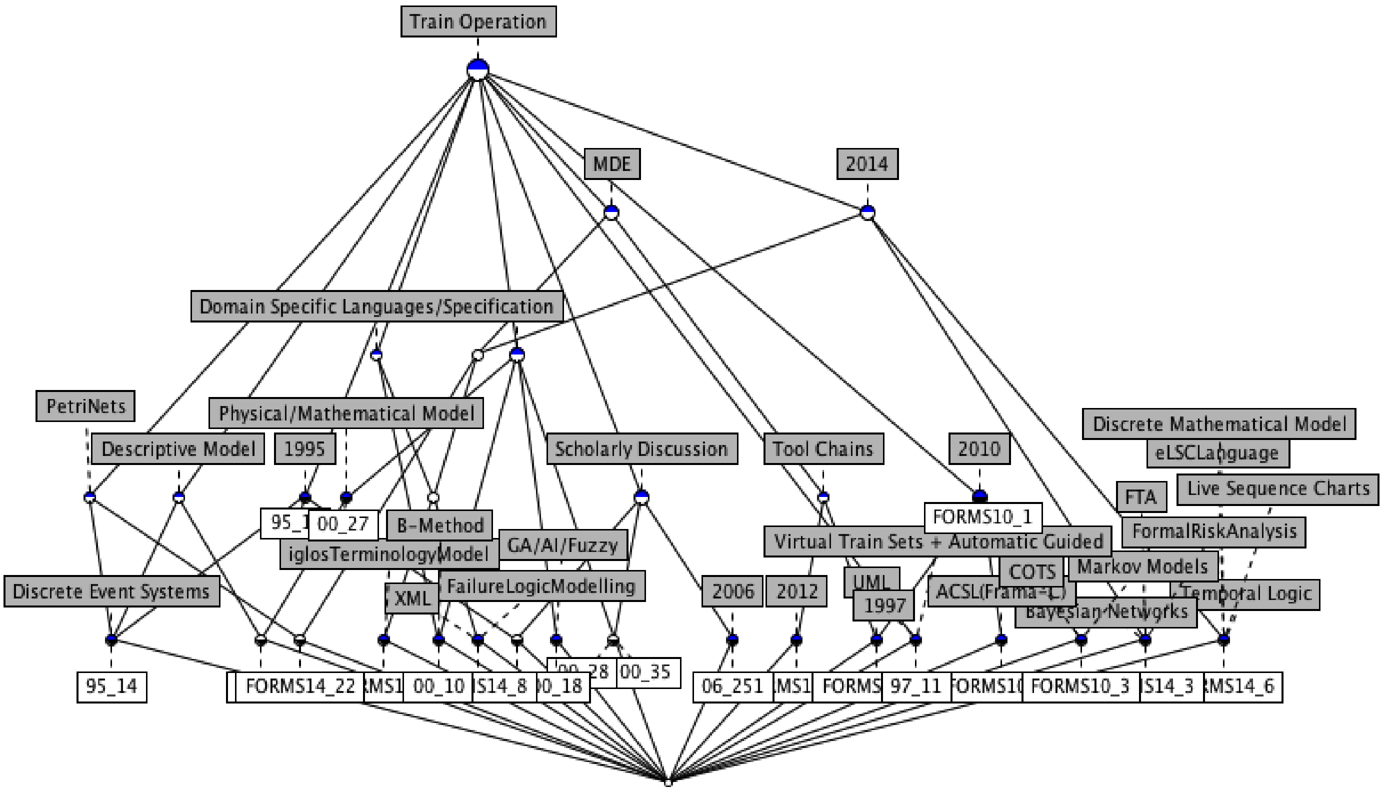



Figure 8.12: Lattice representing The Train Operation subdomain

Other concepts in this subdomain were too insignificant to be added to the table, that is, they contained an extent of only one object. Concepts that overlap other subdomains are documented in the later part of the chapter.

### 8.1.8 Train Operation

| Concept # | $\vert A\vert = \vert\text{extent}\vert$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 21 | Train Operation | 0.998721122741699 |
| 2 | 6 | Train Operation, 2000 | 0.859375 |
| 3 | 3 | Train Operation, 1995 | 0.625 |

| | | | |
|---|---|---|---|
| 4 | 5 | Train Operation, 2014 | 0.59375 |
| 5 | 2 | Train Operation, 2000, Mathematical Model | 0.5 |
| 6 | 4 | Train Operation, 2010 | 0.4375 |
| 7 | 4 | Train Operation, Scholarly Discussion | 0.4375 |
| 8 | 5 | Train Operation, MDE | 0.375 |
| 9 | 3 | Train Operation, MDE, 2014 | 0.375 |
| 10 | 4 | Train Operation, Scholarly Discussion | 0.4375 |
| 11 | 3 | Train Operation, Domain Specific Formalisms | 0.375 |
| 12 | 2 | Train Operation, Descriptive Model | 0.25 |
| 13 | 2 | Train Operation, Petri Nets | 0.25 |

Table 8.7: Stability calculations for the concepts in Train Operation subdomain

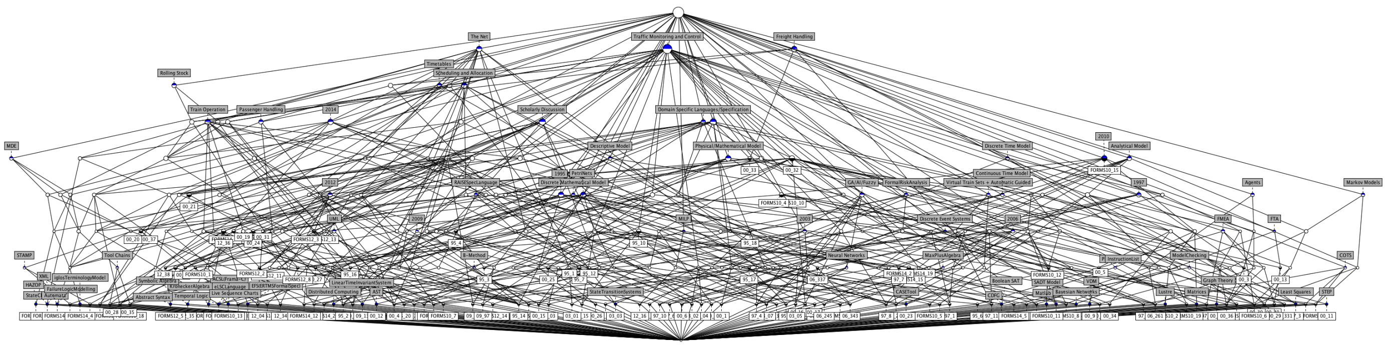


Figure 8.13: Lattice representing the complete railway domain

### 8.1.9 Complete Lattice

| Concept # | $\|A\| = \|\text{extent}\|$ | $B = \{\text{intent}\}$ | Stability $\sigma(A, B)$ |
|---|---|---|---|
| 1 | 9 | All subdomains | 0.943359375 |
| 2 | 2 | All subdomains, 2000, Scholarly Discussion | 0.75 |
| 3 | 3 | All subdomains, 2010 | 0.625 |
| 4 | 4 | All subdomains, MDE | 0.5625 |
| 5 | 3 | All subdomains, 2014 | 0.375 |
| 6 | 2 | All subdomains, MDE, Tool Chains | 0.25 |
| 7 | 2 | All subdomains, Domain Specific Formalisms, MDE, 2014 | 0.25 |
| 8 | 18 | The Net Traffic Monitoring and Control | 0.984031677246094 |
| 10 | 6 | The Net, Traffic Monitoring and Control, 2000 | 0.71875 |

| | | | |
|---|---|---|---|
| 11 | 5 | The Net, Traffic Monitoring and Control, 2014 | 0.59375 |
| 12 | 4 | The Net, Mathematical Model | 0.5625 |
| 14 | 12 | The Net, Traffic Monitoring and Control, Passenger Handling, Freight Handling | 0.497802734375 |
| 15 | 5 | The Net, Traffic Monitoring and Control, Scholarly Discussion | 0.46875 |
| 18 | 3 | The Net, Traffic Monitoring and Control, Domain Specific Formalisms, 2014 | 0.375 |
| 19 | 2 | The Net, Rolling Stock, Mathematical Model | 0.25 |
| 20 | 2 | The Net, Traffic Monitoring and Control, 2012 | 0.25 |
| 21 | 2 | The Net, Traffic Monitoring and Control, Passenger Handling, Freight Handling, Mathematical Model | 0.25 |
| 24 | 12 | Timetables, Traffic Monitoring and Control | 0.497802734375 |
| 25 | 4 | Timetables, Traffic Monitoring and Control, 2010 | 0.4375 |
| 26 | 2 | Timetables, Traffic Monitoring and Control, Discrete Mathematical Model | 0.25 |

| | | | |
|---|---|---|---|
| 27 | 2 | Timetables, Traffic Monitoring and Control, Discrete Mathematical Model | 0.25 |
| 28 | 14 | Scheduling and Allocation, Traffic Monitoring and Control | 0.874267578125 |
| 29 | 12 | Scheduling and Allocation, Rolling Stock, Freight Handling | 0.49951171875 |
| 30 | 4 | Scheduling and Allocation, Traffic Monitoring and Control, Scholarly Discussion | 0.4375 |
| 31 | 2 | Scheduling and Allocation, Rolling Stock, Freight Handling, 2012 | 0.25 |
| 32 | 2 | Scheduling and Allocation, Traffic Monitoring and Control, 1997 | 0.25 |
| 33 | 2 | Scheduling and Allocation, Traffic Monitoring and Control, UML | 0.25 |
| 34 | 11 | Traffic Monitoring and Control, Train Operation | 0.74755859375 |
| 35 | 13 | Traffic Monitoring and Control, Freight Handling | 0.498046875 |
| 36 | 5 | Traffic Monitoring and Control, Freight Handling, 2000 | 0.4375 |
| 37 | 3 | Traffic Monitoring and Control, Train Operation, Scholarly Discussion | 0.325 |

| | | | |
|---|---|---|---|
| 38 | 2 | Traffic Monitoring and Control, Freight Handling, 2000, Descriptive Model | 0.25 |

Table 8.8: Stability calculations for the concepts in Complete context and includes overlapping subdomains

These results are discussed in the Discussion chapter that follows.

# Chapter 9

# Discussion

## 9.1 Analysis of the Context

The results in the tables 8.1 to 8.8 are summarised versions of the complete results. The full text file can be found in the appendix under the section *Results.txt*.

Not only is it necessary to study what the results mean, but performing some calculations on said results is necessary in order to give more strength and depth to the final conclusions. There were a number of assumptions made and the impact of those decisions needs to be studied as well.

### 9.1.1 Data

The data that was used to construct this lattice and calculate stabilities for the lattice needs to be examined. The full details from where this data comes from and why these sources were picked can be found in Chapter 6 of this thesis.

The data that is used in this project is a subset of the data in the railway domain.

It is a combination of papers collected from accessible sources and relevant conferences that span a substantially long period of time — almost forty years. An assumption can be made that this is an appropriate representation of the data in this domain. The reasons for this assumption (also detailed in Chapter 6) are the following:

1. The sources used to collect the data were recommended by practitioners within the domain of railway software systems(See acknowledgements for the full list).

2. The data collected covers a substantial period of time — over thirty years. This means, the data hopefully captures the changes in the domain's discussions as well.

3. The data is picked from sources that are accessible to all stakeholders in the domain including those who practice in the domain and also research it for academic purposes. In this way, a broad view of the domain as it applies to all interested parties can be achieved.

Therefore, it is reasonable to assume that the dataset with which to do an evaluation using Formal Concept Analysis is satisfactory.

In the end, a total of one hundred and fifty usable papers were recorded and included in the final context. Some papers in these conferences were not usable due to irrelevancy and some sought after conferences were not available even through an interlibrary loan or online stores. There was also not enough time to search for even more papers. Also, an expert in the field of FCA, namely Professor Sergei Obiedkov was consulted and in his opinion, one hundred and fifty papers were enough to provide reasonable stability index values from a statistical perspective.

Therefore, due to these limitations and the knowledge from experts, the assumption can also be made that the volume of the dataset was reasonably large enough to be considered relevant. At the very least, this is a promising start to the data analysis, even if some of the underlying patterns could only be observed with more data.

### 9.1.2 FCA and calculations

This subsection will now discuss some of the results and analyse the calculations that are in Chapter 8. After all of the data is put into the Conexp tool, the results and some discussions on them are listed below:

1. **There are 320 concepts altogether.** This means there are roughly double the number of concepts than items. This indicates that there are complex relationships within the lattice and perhaps many varying patterns in it. This is expected since the attributes chosen for the context are from a number of areas, such as year, keywords and subdomains. These three characteristics recorded are bound to have overlaps and create a more chaotic lattice.

2. **The size of the extent is 150.** That is, there are 150 objects or papers within the context. The papers extend from 1980 to 2014 and come from a variety of conferences. While there is a lot of potential for more papers, this is a substantial size for reasonable results — if the availability and relevance of the papers that exist in the railway domain are taken into account.

3. **The size of the intent is 80.** That is, there are 80 attributes within the context. This size was not pre-planned. The number of attributes increased

as more formal methods were found to be used in the papers, the more years in which the papers were published and also depending on the number of subdomains. The method used to find the appropriate subdomains can be found in Chapter 7 of this report and were fixed throughout the data collection.The lower number of attributes means that either there is a lot of overlap across the papers or that there is not much useful recorded information from these papers. In the case of this context, both reasons are true.

4. **The lowest stability is 0.1875.** This shows us that using the tools of Formal Concept Analysis, all concepts have at least some stability. This means that it not possible to ignore any of the concepts immediately due to the non-zero stability indices.

5. **The highest stability is 1.0.** This value does not tell us much except that there is at least one concept which is completely stable, so in at least one case where a significant relationship has been found in the data.

6. **The concept with the largest extent(all objects) but no intent has a stability of 0.9999. The concept with the largest intent (all attributes) but no extent has a stability of 1.0.** This result means that the objects and attributes in the lattice are pertinent to each other. This allows us to construct a meaningful lattice where these relationships between the concepts can be seen.

7. **There are 98 concepts with only one object in their extent — roughly 31% of total concepts.** This result indicates that no conclusion can be extracted from these concepts. It is unclear whether the concepts are stable or not, even though almost all of them have stability indices of 0.5.

It is also important to account for noise in the lattice. If a concept contains only one object, it could well be noise, even if it has a high stability value. This is the major reason the papers by Kunetzov recommend performing stability calculations on concepts that have an extent of size 2 or larger.

It is possible that with more data a larger concept would emerge. However, this is not something we can detect with the current dataset so these concepts are ignored in our further calculations at the risk of missing some relationships between concepts. The reason that this is not alarming is that most of these concepts are subconcepts or child concepts of larger, more stable concepts at a higher level. Therefore, if nothing, at least more general relationships can be seen and the larger concepts will retain characteristics of the subconcepts.

### 9.1.3 Threshold Exploration

It is now needed to refine the data further to decide what the results mean. As discussed in Chapter 7, the most common way to analyse formal contexts is the use of stability in determining which data is pertinent to the structure of the context and which is not.

The focus here will be on stability thresholds for the sake of noise-reduction in the data.

As said in the previous section, the lowest stability is 0.1875 and the highest value is 1.0. A method is needed to determine which stability value makes a certain concept relevant to our discussion and which is safe to leave out.

Firstly, consider using mean, median or mode as the simplest and most straightforward way of choosing a stability index as the threshold. The mean value of stability in this context is 0.5219. The median is 0.5 and the mode is also 0.5. Therefore, solely from a statistical point of view, 0.5 is probably a good "guess" at the threshold for stability indices.

The data itself can fine tune this choice to our purpose. In this case, we can observe what percentage of the data is included in the lattice at each possible stability index we might use as a threshold.

Figures 9.14 and 9.15 show the possible threshold values plotted against the percentage of data inclusion for each one. Where needed, very close threshold values are calculated to observe the sudden drop in data inclusion. These will be the most likely candidates for thresholds since loss of a large amount of data is undesirable.

Figure 9.14 shows the stability index values as percentages and plots it against the data inclusion at each stability index. This graph uses the raw data without any removal. A clear drop is observed at the 50% or 0.5 stability index where 40% of the data is lost in the context — from about 65% of data to 25% on average.

The reason for this sharp drop is that the data lost has a lot of small concepts (concepts with small extents) with the stability index of exactly 0.5. 0.5 is the stability value obtained when calculating the stability index for a concept with only one object in it(See Chapter 7). Simply looking at the concept lattice shows us that there are numerous concepts with only one object in their extent. As mentioned in the earlier section, concepts with one object in their extent are unknown quantities and can be left out on the basis of noise reduction. This gives a strong argument for the median value of 0.5 to be considered as a threshold.

However, this graph is based on raw data. It is worthwhile to consider how the graph would look with some of the unnecessary data already filtered out. Then, the

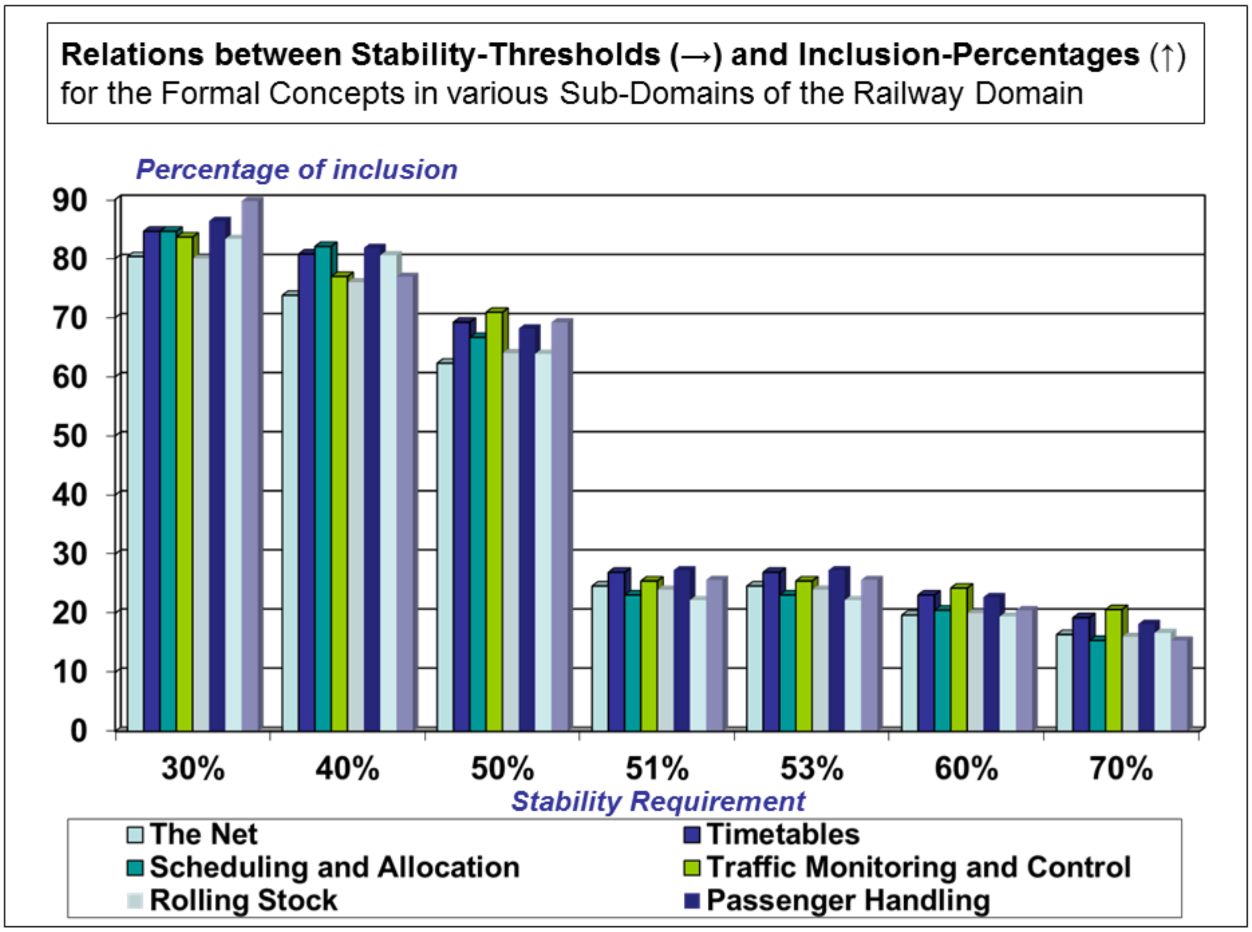



Figure 9.14: Changes of stability indices with inclusion changes with all concepts in the context

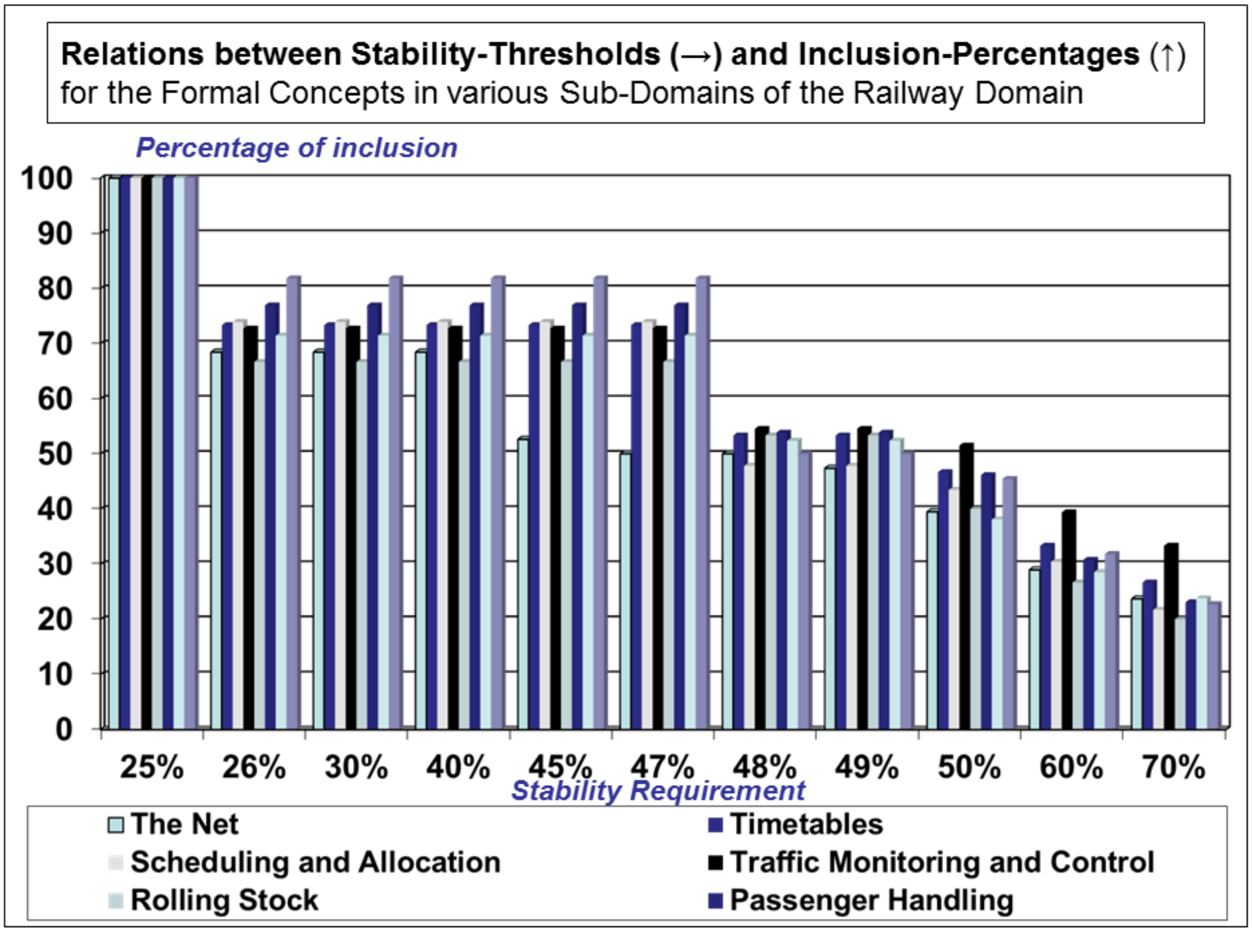



Figure 9.15: Changes of stability indices with inclusion changes with all concepts with more than one object

graph changes to look like Figure 9.15.

Here, it is easy to notice that there are two large drops in data inclusion. One at 25% and one drop at 47%.

The drop at 25% is largely because the calculation of stability indices of a concept with an extent of 2 objects but only one subset that is completely encompassed by the concept's intent. This makes it more likely to be noise as the stability index is dependent on 2 objects sharing a particular intent. While less likely to be noise than one-object concepts, no definite conclusion can be made about these results as well.

The drop at 47% is a result of the calculations of stability indices on concepts that contain 3, 4 and 5 objects. These are substantial enough to not be considered as noise. But they are still many of these concepts within the lattice which is why a drop is seen.

Since keeping 100% of the data is not needed and defeats the purpose of noise-reduction, the data inclusion drop at 25% is not so meaningful. However, the drop of 20% of the data between the thresholds 47% and 48%, is much more reasonable and would probably highlight the most important concepts within the context.

Of course this also means that smaller concepts and their relationships are most likely removed. This can be fixed by using a larger sample size or relaxing thresholds. For now, it is reasonable to find any general patterns and trends that would indicate the stable (or unstable) use of formal methods in the railway domain — which is the aim.

The lattice with these thresholds implemented can be seen in Figure 9.16. The resulting clean up of noise does indeed form a lattice(which is not necessarily the case) and displays some of the most important concepts within the data. There are 38 objects in the extent in total.

It is now safe to say that the intents of this lattice are extremely pertinent to the railway domain.

After the results are analysed and filtered done, the formal methods with the highest stability values found are:

1. Petri Nets

2. Mathematical Models

3. Max Plus Algebra
4. Discrete Event Systems
5. Fuzzy Logic (Artifical Intelligence)

Other methods include:

1. UML
2. Artificial Intelligence (Genetic Algorithms, Fuzzy Logic, Neural Networks)
3. Formal Risk Analysis
4. Virtual Train Sets & Automatic Guided Systems
5. Scholarly Discussions

The years with the highest stability indices are: 1995, 1997, 2000, 2006, 2010, 2012 and 2014. And finally, the subdomains with the highest stability indices are:

1. Traffic Monitoring and Control
2. Timetables
3. Scheduling and Allocation
4. The Net
5. Train Operation
6. Freight Handling

For a description of any of these labels and categories, please see Chapter 7, Step 3.

Looking at these lists, it is interesting to note that some groups that were created (in Step 3, Chapter 7) are missing such as “languages”, but specific languages such as “UML” are filtered into the final lattice. In other cases some detail is missing which could have created a concept of its own, such as “Fuzzy Logic” and “Genetic Algorithms” that was combined into a single concept under “Artificial Intelligence”. Additionally, there are general concepts that do not give much meaning to the knowledge described in the sources, such as “Mathematical Models” which could include fundamental concepts, theoretical tools or even just practical engineering considerations. This stems from a lack of expertise in the railway domain when creating these attribute categories. Therefore, it can be said that this process would need refinement in future.

Ultimately, there are at least five formal methods that are settled in this subset of data. The fact that all of these lists are not empty means that, according to our original criteria, there is definitely settled knowledge to be found in the railway domain and it has possibly been available at least since the year 1995.

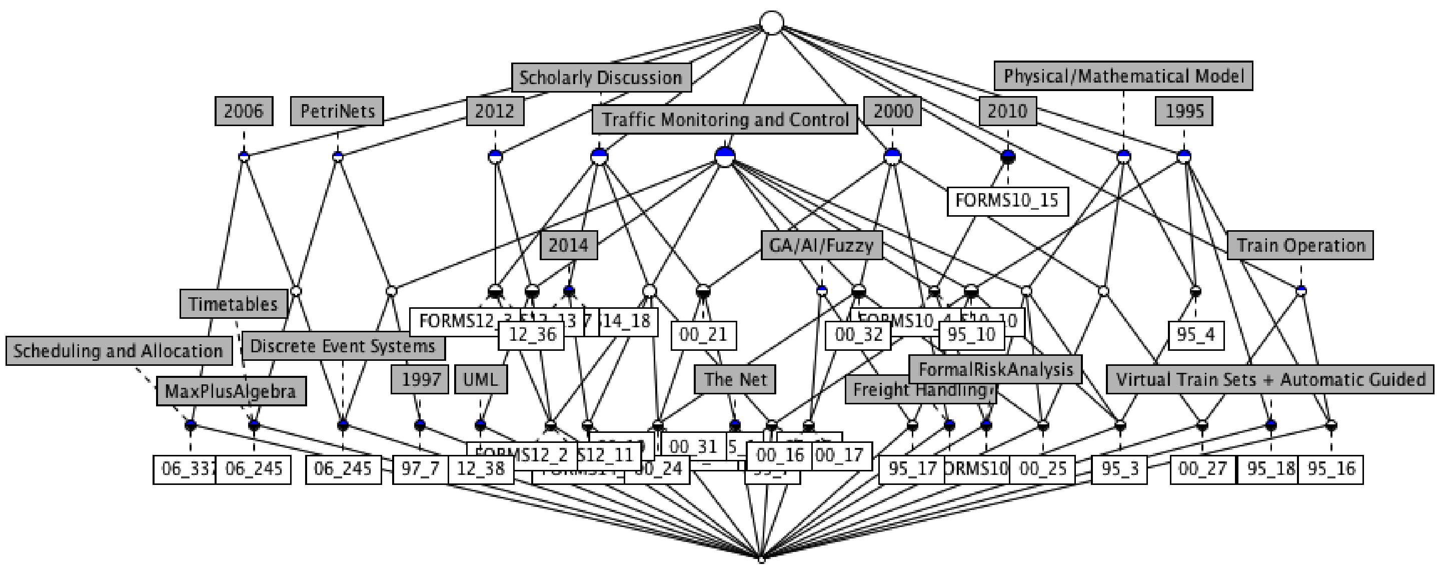



Figure 9.16: Final lattice. Each concept has an extent size of > 1 and stability index of ≥ 0.47

# Chapter 10

# Conclusion

This chapter starts with a recapitulation of settled knowledge and how the process of putting formal methods into a concept lattice would help us determine settled knowledge in the domain. Next the results of the project and their meaning for the aim of the project is outlined and discussed. Lastly, a case is made for or against starting work on a Body(Book) of Knowledge for the railway domain.

## 10.1 Settled Knowledge

From the discussion in Chapter 3, settled knowledge can be best described by answering a set of four questions, as follows.
Firstly, we can describe the structure of settled knowledge much more concisely and therefore answer the question: "*what is settled knowledge exactly?*". It could be in the form of:

1. Fundamental design concepts that consist of:

   (a) Operational theories.

   (b) Normal configuration.

2. Criteria and specifications.

3. Theoretical tools — mathematical models and mature scientific theories.

4. Quantitative data.

5. Practical considerations.

6. Design Instrumentalities.

7. Tacit knowledge[45] — that teaches the practitioner how to think in order to approach a certain problem within the domain.

Secondly, *what does settled knowledge look like?* It could be in the form of graphs, diagrams, formulae or a list of steps in a method. Settled knowledge appears in a format for best describing problems and their solutions within a chosen domain.

Thirdly, *what are the sources of this settled knowledge?* This would be almost any document produced within the domain. This would include papers in journals and conferences, standards and guidelines from governing bodies or even requirement documentation from domain experts and stakeholders.

Lastly, *how long does settled knowledge exist?* The use of knowledge consistently over a period of time for similar problems within the domain is essential. That is, the common use of a piece of knowledge for a specific problem in the domain over a majority of the time being studied — in particular, skewed towards later times rather than older. This accounts for changes or refinements in the knowledge as time passes.

All of this information mentioned above is expected to be collected and catalogued in a future Body of Knowledge (BoK) for the railway domain.
On the other hand, settled knowledge is not:

1. Phased out by new designs or methods.

2. Only used in research.

3. Used by some practitioners and not others.

4. Able to disregard practical issues.

Within this project, the focus is on the use of formal methods as an aspect of settled knowledge in the railway domain. Formal methods are an indication of mathematical certainty in solving a problem within the domain.

However, this means that many other types of settled knowledge are left out in the collection of the data. Although, this does not give us a holistic view on settled knowledge within the domain, it is the first step to collecting settled knowledge. The problem of collecting other types of knowledge, such as practical criteria or tacit knowledge[45], is not dealt with in the project but can be done with the right knowledge sources and the cooperation of experts in the domain.

## 10.2 Final Lattice

A lattice is a partially ordered set of concepts(see Chapter 4).

From the discussion chapter, the final list of stable concepts can be extracted from the lattice. The final lattice is reproduced here in figure 10.17. The full description

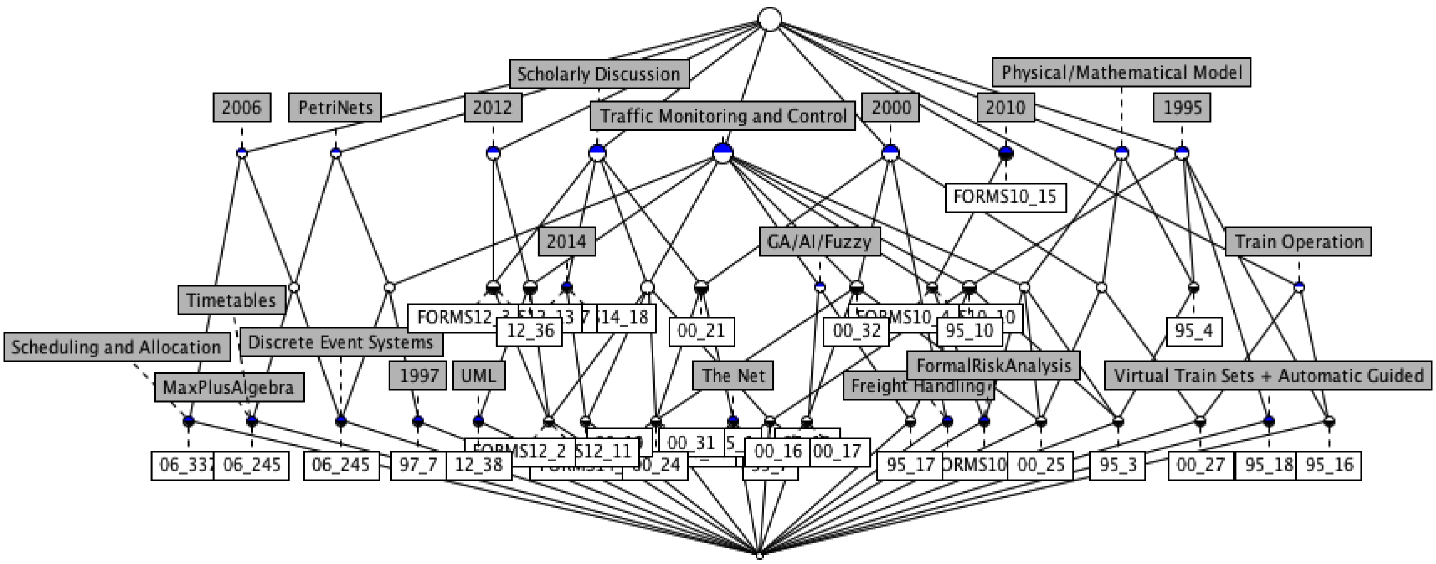



Figure 10.17: Final lattice. Each concept has an extent size of $> 1$ and stability index of $\geq 0.47$

of the evaluation of the techniques used to reach this lattice can be found in Chapter 9.

Looking at the structure of the lattice, three layers are clearly visible and there are some deductions that can be made from this. For the extraction of meaning from lattices, see Chapter 4.

There are a few things that can be said about the lattice.

1. All years(excluding 2003,2006 and 2009) are present and most of them are on the top layer. That means they are largely super concepts or parent concepts of the others. Therefore, the data included within the final lattice includes data from 1995 to 2014. However, 1997 and 2014 are less independent from the general dataset than other years. There seems to be a larger focus on only the subdomain of Traffic Monitoring and Control during this time.

2. There is a large general concept for scholarly discussions, mainly in 2014.

3. Mathematical Models and Petri Nets are large general concepts with almost no dependency on year, subdomain or other techniques.

4. Automatic Guided Systems(and Virtual Train Sets) were largely explored in the 2000s.

5. There are some unknown concepts that exist within the lattice, of some complicated pattern to be found between many attributes.

6. Traffic Monitoring and Control is the largest concept in terms of extents related to it and uses major techniques such as Artificial Intelligence, Formal Risk Analysis, Petri Nets, Mathematical Models in the years ranging from 1995 right up to 2014.

7. UML was used to model largely in the subdomain of Traffic Monitoring and Control.

8. Petri Nets were combined with other techniques such as Max Plus Algebra and Discrete Event Systems for use in the subdomains of Timetables and Traffic Monitoring and Control.

9. Mathematical Models is a technique usually coupled with Formal Risk Analysis and Automatic Guided Systems.

These are just the first conclusions we can draw from a small lattice such as this, and it shows us the power of Formal Concept Analysis in categorizing data in this way.

Using Formal Concept Analysis, it is possible to find common formal methods that are used over a long period of time and across particular subdomains in the sources of knowledge. The first iteration of this process with a subset of data has been completed here.

High stability of these concepts shows the independence of the intent from the objects of the concept[27]. Stability provides noise-resistance in this way. Also, significant concepts surface instead of just the large ones. This gives us a chance to find characteristics of the data and subdomains, rather than just a blind method to find the "most common" methods. It is also known that high stability values means concepts are less likely to merge into other larger concepts, or break up into smaller ones when some objects are missing from the extents. This means we miss none of the underlying patterns that some object descriptions might hide.

The final list of the intents (formal methods, subdomains, years) with the highest stability values can be seen in the last section of Chapter 9.

Now it is safe to say that the remaining attributes in the final lattice best characterise the information from the sources. The most significant intents are isolated and it contains a list of subdomains that use the most modelling and formal techniques. Valuable and useful techniques for a particular problem are considered one characteristic of settled knowledge.

The final lattice also contains the formal methods as well as the years in which the use of these techniques were published. The wide range of years indicates a spread of this information over a number of decades. This is one of the most important conditions of settled knowledge as described in Chapter 3.

Additionally, there are formal methods particular to a subdomain(Max-Plus Algebra) as well as some that are generally used throughout the various subdomains(Petri

Nets, Mathematical Models). This shows us particular practices for a specific subdomain or problem, along with general solutions to problems in the railway domain as a whole. This is again, are the hallmarks of settled knowledge.

With the use of this method and starting with more sources of knowledge, the use of other formal methods can be extracted. Since it is possible to say that within the subset of sources chosen in this project, settled knowledge has been found, therefore this can be extended to the rest of the knowledge within the domain and say that settled knowledge can be found within the whole domain as well.

This in turn provides the answer to the question of whether it is possible to start constructing a Body of Knowledge (BoK) in the Railway domain. The answer is, yes it is possible, and the first step to isolating the knowledge that goes into it is using Formal Concept Analysis as a filter to find the settled knowledge.

## 10.3 Improvements and Future Work

In hindsight, a greater knowledge about the railway domain could provide a better classification system for the knowledge that is encountered in the sources. Also, approaching the domain from a detailed examination of the problems being solved rather than their solutions, might provide a better arrangement of knowledge for a Book of Knowledge.

Lattices could also have been divided by other aspects of the knowledge, for example, years published or even type of problems being solved. Some of the larger subdomain categories could have been further broken down to give more depth to the structure.

A selection of engineering papers as sources of knowledge would also give a better view of industry techniques as well as the current state of the domain in practice at the moment.

Future work could involve finding a much more realistic structure of the knowledge within the domain using more advanced techniques in Formal Concept Analysis. It is commonly used for knowledge ontologies and many tools use attribute exploration and implication relations to ask the user a series of questions to complete the unknown parts of the lattice using data it is representing[35].

Other FCA tools like Toscana allows a fixed structure to be created first and then the user can insert data into the structure confirming his/her hypothesis. There is also support for line diagrams that allow nested structures so that more complicated lattices are possible[35]. In this way, FCA can also be applied in great detail to particular subdomains or sub-subdomains so that finer details in the structure of the knowledge can be identified. This will help in the collection and categorisation of knowledge in the railway domain when constructing the BoK along with the identification of said settled knowledge.

The use of Formal Concept Analysis can also be extended to other areas of knowledge within the railway domain (not just formal methods) although not all knowledge is straightforward to identify.

The same work from Vincenti[45], Bjorner[8] and Shaw[41] can give more dimensions to the categories of knowledge and the relevance of each to particular aspects of the railway domain.

Finally, the method outlined in this project can be used on an already "mature" engineering field, such as Control Systems. From this activity, a model can be created representing how a concept lattice containing knowledge evolves as the domain settles down. This model can be mapped to the behaviour of the railway domain (and indeed, any domain that needs to be studied) to determine the extent of its maturity.

# Appendix A

# Appendix

## A.1 Results.txt

(12_34 12_35 12_36 12_38 09_01 09_97 06_261 06_349 03_07 03_04 03_03 03_02 00_2 00_3 00_4 00_5 00_8 00_12 00_13 00_15 00_16 00_23 00_29 00_30 00_32 00_7 00_9 00_11 00_17 00_19 00_22 00_24 00_25 00_26 00_28 00_31 00_34 00_35 00_36 00_39 00_40 97_1 97_2 97_5 97_6 97_7 97_8 97_11 95_1 95_2 95_3 95_6 95_7 95_8 95_9 95_10 95_12 95_13 95_15 95_17 95_19 95_20 06_245 FORMS10_1 FORMS10_2 FORMS10_4 FORMS10_5 FORMS10_6 FORMS10_7 FORMS10_8 FORMS10_9 FORMS10_10 FORMS10_11 FORMS10_12 FORMS10_13 FORMS10_14 FORMS10_18 FORMS10_19 FORMS14_1 FORMS14_2 FORMS14_4 FORMS14_5 FORMS14_6 FORMS14_8 FORMS14_9 FORMS14_11 FORMS14_12 FORMS14_13 FORMS14_14 FORMS14_15 FORMS14_19 FORMS14_21 FORMS12_2 FORMS12_5 FORMS12_6 FORMS12_8 FORMS12_10 FORMS12_11 FORMS12_12 FORMS12_15 , Traffic Monitoring and Control ) 1

(, The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling LinearTimeInvariantSystem UML PetriNets MILP Train Operation GA/AI/Fuzzy Agents Symbolic Algebra 2003 2006 2009 2012 Domain Specific Languages/Specification Discrete Time Model Continuous Time Model FMEA MaxPlusAlgebra Virtual Train Sets + Automatic Guided RAISESpecLanguage FormalRiskAnalysis StateTransitionSystems Neural Networks 2000 AST FTA Markov Models CASETool Physical/Mathematical Model Descriptive Model Analytical Model Scholarly Discussion VDM B-Method STEP Graph Theory Discrete Event Systems 1997 COTS Least Squares 1995 Distributed Computing Live Sequence Charts MDE Bayesian Networks STAMP Satellite Comms Automata HAZOP KroneckerAlgebra Lustre Discrete Mathematical Model InstructionList Boolean SAT Matlab Matrices CDFG ACSL(Frama-C) SADT Model ModelChecking Tool Chains PLCs iglosTerminologyModel eLSCLanguage Temporal Logic FailureLogicModelling XML StateCharts EFS(ERTMSFormalSpec) Abstract Syntax 2010 2014 ) 1

(12_03 12_04 12_16 12_34 12_35 12_36 12_38 09_01 09_13 09_97 06_251 06_331 06_261 06_337 06_343 06_349 03_07 03_06 03_05 03_04 03_03 03_02 03_01 06_245 00_1 00_2 00_3 00_4 00_5 00_6 00_8 00_12 00_13 00_14 00_15 00_16 00_18 00_23 00_29 00_30 00_32 00_33 00_7 00_9 00_10 00_11 00_17 00_19 00_20 00_21 00_22 00_24 00_25 00_26 00_27 00_28 00_31 00_34 00_35 00_36 00_37 00_38 00_39 00_40 97_1 97_2 97_3 97_4 97_5 97_6 97_7 97_8 97_9 97_10 97_11 95_1 95_2 95_3 95_4 95_5 95_6 95_7 95_8 95_9 95_10 95_11 95_12 95_13 95_14 95_15 95_16 95_17 95_18 95_19 95_20 06_245 FORMS10_1 FORMS10_2 FORMS10_3 FORMS10_4 FORMS10_5 FORMS10_6 FORMS10_7 FORMS10_8 FORMS10_9 FORMS10_10 FORMS10_11 FORMS10_12 FORMS10_13 FORMS10_14 FORMS10_15 FORMS10_16 FORMS10_17 FORMS10_18 FORMS10_19 FORMS10_20 FORMS14_1 FORMS14_2 FORMS14_3 FORMS14_4 FORMS14_5 FORMS14_6 FORMS14_8 FORMS14_9 FORMS14_11 FORMS14_12 FORMS14_13 FORMS14_14 FORMS14_15 FORMS14_16 FORMS14_17 FORMS14_19 FORMS14_20 FORMS14_21 FORMS12_2 FORMS12_3 FORMS12_5 FORMS12_6 FORMS12_8 FORMS12_10 FORMS12_11 FORMS12_12 FORMS12_13 FORMS12_14 FORMS12_15 FORMS14_18 FORMS14_22 , ) 0.999999999999993

(00_2 00_3 00_4 00_5 00_8 00_12 00_13 00_15 00_16 00_23 00_29 00_30 00_32 00_7 00_9 00_11 00_17 00_19 00_22 00_24 00_25 00_26 00_28 00_31 00_34 00_35 00_36 00_39 00_40 , Traffic Monitoring and Control 2000 ) 0.999999295920134

(12_34 09_01 06_349 03_04 00_7 00_19 00_24 00_28 00_31 00_35 00_39 97_6 95_1 95_2 95_6 95_7 95_12 95_13 FORMS14_1 FORMS14_9 FORMS14_11 FORMS14_14 FORMS12_2 FORMS12_11 , Traffic Monitoring and Control Scholarly Discussion ) 0.999985098838806

(12_34 09_01 06_251 06_349 03_04 00_7 00_19 00_20 00_21 00_24 00_28 00_31 00_35 00_37 00_39 97_6 97_9 97_10 95_1 95_2 95_6 95_7 95_11 95_12 95_13 FORMS14_1 FORMS14_9 FORMS14_11 FORMS14_14 FORMS14_17 FORMS14_20 FORMS12_2 FORMS12_3 FORMS12_11 FORMS12_13 FORMS12_14 FORMS14_18 , Scholarly Discussion ) 0.999877919784922

(00_1 00_2 00_3 00_4 00_5 00_6 00_8 00_12 00_13 00_14 00_15 00_16 00_18 00_23 00_29 00_30 00_32 00_33 00_7 00_9 00_10 00_11 00_17 00_19 00_20 00_21 00_22 00_24 00_25 00_26 00_27 00_28 00_31 00_34 00_35 00_36 00_37 00_38 00_39 00_40 , 2000 ) 0.999511716938287

(12_04 09_13 03_03 03_01 00_4 00_6 00_12 00_15 00_7 00_20 00_28 00_35 00_37 97_4 97_10 95_2 95_14 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_11 FORMS14_12 FORMS14_16 FORMS12_10 FORMS12_12 , The Net ) 0.999011229723692

(06_251 00_18 00_10 00_27 00_28 00_35 00_38 97_11 95_13 95_14 95_16 FORMS10_1 FORMS10_3 FORMS10_13 FORMS10_18 FORMS14_3 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 FORMS14_22 , Train Operation ) 0.9987211227416

(FORMS10_1 FORMS10_2 FORMS10_4 FORMS10_5 FORMS10_6 FORMS10_7 FORMS10_8 FORMS10_9 FORMS10_10 FORMS10_11 FORMS10_12 FORMS10_13 FORMS10_14 FORMS10_18 FORMS10_19 , Traffic Monitoring and Control 2010 ) 0.998382568359375

(12_34 09_97 03_07 00_15 97_1 97_7 95_15 06_245 FORMS10_5 FORMS14_2 FORMS14_5 FORMS14_15 FORMS14_19 FORMS12_8 , Traffic Monitoring and Control PetriNets ) 0.998291015625

(12_34 12_35 12_36 12_38 FORMS12_2 FORMS12_5 FORMS12_6 FORMS12_8 FORMS12_10 FORMS12_11 FORMS12_12 FORMS12_15 , Traffic Monitoring and Control 2012 ) 0.995361328125

(95_1 95_2 95_3 95_6 95_7 95_8 95_9 95_10 95_12 95_13 95_15 95_17 95_19 95_20 , Traffic Monitoring and Control 1995 ) 0.99530029296875 (FORMS14_1 FORMS14_2 FORMS14_4 FORMS14_5 FORMS14_6 FORMS14_8 FORMS14_9 FORMS14_11 FORMS14_12 FORMS14_13 FORMS14_14 FORMS14_15 FORMS14_19 FORMS14_21 , Traffic Monitoring and Control 2014 ) 0.99456787109375

(09_13 00_2 00_4 00_25 00_26 00_27 00_38 00_39 97_4 95_2 95_3 95_4 95_19 FORMS10_5 FORMS10_8 FORMS10_11 FORMS10_12 FORMS10_14 FORMS10_16 FORMS10_17 , Physical/Mathematical Model ) 0.991981506347656

(00_2 00_4 00_25 00_26 00_39 95_2 95_3 95_19 FORMS10_5 FORMS10_8 FORMS10_11 FORMS10_12 FORMS10_14 , Traffic Monitoring and Control Physical/Mathematical Model ) 0.9912109375

(12_04 12_34 09_97 03_07 06_245 00_6 00_14 00_15 97_1 97_7 95_14 95_15 06_245 FORMS10_5 FORMS14_2 FORMS14_5 FORMS14_15 FORMS14_19 FORMS12_8 FORMS14_22 , PetriNets ) 0.984332084655762

(95_1 95_2 95_3 95_4 95_5 95_6 95_7 95_8 95_9 95_10 95_11 95_12 95_13 95_14 95_15 95_16 95_17 95_18 95_19 95_20 , 1995 ) 0.984293937683105

(FORMS14_1 FORMS14_2 FORMS14_3 FORMS14_4 FORMS14_5 FORMS14_6 FORMS14_8 FORMS14_9 FORMS14_11 FORMS14_12 FORMS14_13 FORMS14_14 FORMS14_15 FORMS14_16 FORMS14_17 FORMS14_19 FORMS14_20 FORMS14_21 FORMS14_18 FORMS14_22 , 2014 ) 0.984189033508301

(12_03 12_04 12_16 12_34 12_35 12_36 12_38 FORMS12_2 FORMS12_3 FORMS12_5 FORMS12_6 FORMS12_8 FORMS12_10 FORMS12_11 FORMS12_12 FORMS12_13 FORMS12_14 FORMS12_15 , 2012 ) 0.984115600585938

(03_03 00_4 00_12 00_15 00_7 00_28 00_35 95_2 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_11 FORMS14_12 FORMS12_10 FORMS12_12 , The Net Traffic Monitoring and Control ) 0.984031677246094

(12_16 09_97 06_337 06_343 03_05 00_1 00_4 00_7 00_28 00_35 97_6 97_8 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_20 FORMS12_12 , Scheduling and Allocation ) 0.982369422912598

(12_16 06_331 03_06 00_4 00_14 00_23 00_33 00_7 00_28 00_35 97_3 95_2 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Freight Handling ) 0.980278015136719

(03_02 00_5 00_8 00_9 FORMS10_2 FORMS10_19 FORMS14_4 FORMS14_8 FORMS14_12 FORMS14_13 FORMS14_14 , Traffic Monitoring and Control Domain Specific Languages/Specification ) 0.97412109375

(FORMS10_1 FORMS10_2 FORMS10_3 FORMS10_4 FORMS10_5 FORMS10_6 FORMS10_7 FORMS10_8 FORMS10_9 FORMS10_10 FORMS10_11 FORMS10_12 FORMS10_13 FORMS10_14 FORMS10_15 FORMS10_16 FORMS10_17 FORMS10_18 FORMS10_19 FORMS10_20 , 2010 ) 0.96863842010498

(00_5 00_30 00_9 00_22 00_34 00_36 00_40 97_1 97_8 95_9 FORMS10_6 FORMS10_8 , Traffic Monitoring and Control Analytical Model ) 0.965576171875 (06_343 03_07 03_05 00_1 00_16 00_18 00_17 97_4 97_8 95_8 95_17 , GA/AI/Fuzzy ) 0.953125

(00_28 00_35 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation ) 0.943359375

(95_1 95_2 95_6 95_7 95_12 95_13 , Traffic Monitoring and Control Scholarly Discussion 1995 ) 0.9375

(03_02 03_01 00_5 00_8 00_9 00_10 FORMS10_2 FORMS10_17 FORMS10_19 FORMS10_20 FORMS14_4 FORMS14_8 FORMS14_12 FORMS14_13 FORMS14_14 , Domain Specific Languages/Specification ) 0.936065673828125

(06_251 06_331 06_261 06_337 06_343 06_349 06_245 06_245 , 2006 ) 0.93359375

(97_1 97_2 97_3 97_4 97_5 97_6 97_7 97_8 97_9 97_10 97_11 , 1997 ) 0.931640625

(06_261 03_04 FORMS10_11 FORMS10_12 FORMS10_16 FORMS10_17 FORMS14_3 FORMS14_20 FORMS12_14 , FormalRiskAnalysis ) 0.9296875

(00_7 00_19 00_24 00_28 00_31 00_35 00_39 , Traffic Monitoring and Control 2000 Scholarly Discussion ) 0.9296875

(97_1 97_2 97_5 97_6 97_7 97_8 97_11 , Traffic Monitoring and Control 1997 ) 0.9140625

(00_5 00_30 00_9 00_22 00_34 00_36 00_40 , Traffic Monitoring and Control 2000 Analytical Model ) 0.9140625

(FORMS14_2 FORMS14_5 FORMS14_15 FORMS14_19 , Traffic Monitoring and Control PetriNets 2014 ) 0.875

(09_97 00_4 00_7 00_28 00_35 97_6 97_8 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Scheduling and Allocation Traffic Monitoring and Control ) 0.874267578125

(FORMS14_1 FORMS14_9 FORMS14_11 FORMS14_14 FORMS14_17 FORMS14_20 FORMS14_18 , Scholarly Discussion 2014 ) 0.8671875

(00_5 00_6 00_14 00_30 00_9 00_22 00_34 00_36 00_40 97_1 97_4 97_8 95_9 FORMS10_6 FORMS10_8 , Analytical Model ) 0.862884521484375

(12_34 FORMS12_2 FORMS12_3 FORMS12_11 FORMS12_13 FORMS12_14 , 2012 Scholarly Discussion ) 0.859375

(00_18 00_10 00_27 00_28 00_35 00_38 , Train Operation 2000 ) 0.859375

(06_261 00_23 00_34 97_11 95_8 95_19 , Traffic Monitoring and Control Virtual Train Sets + Automatic Guided ) 0.859375

(03_07 00_16 00_17 97_8 95_8 95_17 , Traffic Monitoring and Control GA/AI/Fuzzy ) 0.859375

(00_7 00_19 00_20 00_21 00_24 00_28 00_31 00_35 00_37 00_39 , 2000 Scholarly Discussion ) 0.8515625

(00_2 00_4 00_25 00_26 00_39 , Traffic Monitoring and Control 2000 Physical/Mathematical Model ) 0.84375

(03_07 03_06 03_05 03_04 03_03 03_02 03_01 , 2003 ) 0.828125

(00_4 00_6 00_12 00_15 00_7 00_20 00_28 00_35 00_37 , The Net 2000 ) 0.82421875

(FORMS10_5 FORMS10_8 FORMS10_11 FORMS10_12 FORMS10_14 , Traffic Monitoring and Control Physical/Mathemat Model 2010 ) 0.78125

(00_7 00_20 00_28 00_35 00_37 97_10 95_2 FORMS14_11 , The Net Scholarly Discussion ) 0.77734375

(00_4 00_8 00_23 00_38 97_2 95_14 FORMS10_16 , Descriptive Model ) 0.7578125

(FORMS14_1 FORMS14_9 FORMS14_11 FORMS14_14 , Traffic Monitoring and Control Scholarly Discussion 2014 ) 0.75

(00_30 00_40 , Traffic Monitoring and Control 2000 Markov Models Analytical Model ) 0.75

(06_261 00_29 00_36 97_5 95_6 , Traffic Monitoring and Control FMEA ) 0.75

(12_34 FORMS12_2 FORMS12_11 , Traffic Monitoring and Control 2012 Scholarly Discussion ) 0.75

(00_16 00_17 , Traffic Monitoring and Control GA/AI/Fuzzy 2000 ) 0.75

(12_38 09_97 95_20 FORMS10_18 FORMS12_5 , Traffic Monitoring and Control UML ) 0.75

(00_7 00_20 00_28 00_35 00_37 , The Net 2000 Scholarly Discussion ) 0.75

(00_28 00_35 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation 2000 Scholarly Discussion ) 0.75

(12_03 06_245 00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_7 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Timetables ) 0.74981689453125

(00_28 00_35 97_11 95_13 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Traffic Monitoring and Control Train Operation ) 0.74755859375

(00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling ) 0.744140625

(00_4 00_14 00_23 00_33 00_7 00_28 00_35 , Freight Handling 2000 ) 0.7421875

(00_5 00_6 00_14 00_30 00_9 00_22 00_34 00_36 00_40 , 2000 Analytical Model ) 0.73828125

(06_261 03_06 00_23 00_34 97_11 95_8 95_18 95_19 , Virtual Train Sets + Automatic Guided ) 0.72265625

(00_2 00_4 00_25 00_26 00_27 00_38 00_39 , 2000 Physical/Mathematical Model ) 0.71875

(00_4 00_12 00_15 00_7 00_28 00_35 , The Net Traffic Monitoring and Control 2000 ) 0.71875

(12_16 00_2 00_22 00_36 97_9 FORMS10_6 , Discrete Time Model ) 0.703125

(03_07 03_04 03_03 03_02 , Traffic Monitoring and Control 2003 ) 0.6875

(06_343 03_05 00_1 97_8 , Scheduling and Allocation GA/AI/Fuzzy ) 0.6875

(FORMS10_5 FORMS10_8 FORMS10_11 FORMS10_12 FORMS10_14 FORMS10_16 FORMS10_17 , Physical/Mathematical Model 2010 ) 0.65625

(12_03 00_12 00_26 97_9 FORMS10_6 , Continuous Time Model ) 0.65625

(FORMS14_4 FORMS14_8 FORMS14_12 FORMS14_13 FORMS14_14 , Traffic Monitoring and Control Domain Specific Languages/Specification 2014 ) 0.65625

(FORMS10_11 FORMS10_12 FORMS10_16 FORMS10_17 , FormalRiskAnalysis Physical/Mathematical Model 2010 ) 0.625

(00_1 00_16 00_18 00_17 , GA/AI/Fuzzy 2000 ) 0.625

(95_13 95_14 95_16 , Train Operation 1995 ) 0.625

(95_2 95_3 95_19 , Traffic Monitoring and Control Physical/Mathematical Model 1995 ) 0.625

(06_261 03_04 FORMS10_11 FORMS10_12 , Traffic Monitoring and Control FormalRiskAnalysis ) 0.625

(03_02 00_8 FORMS14_12 FORMS14_14 , Traffic Monitoring and Control Domain Specific Languages/Specificati RAISESpecLanguage ) 0.625

(12_04 00_6 00_15 95_14 , The Net PetriNets ) 0.625

(FORMS10_1 FORMS10_13 FORMS10_18 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation 2010 ) 0.625

(FORMS14_3 FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_22 , Train Operation 2014 ) 0.59375

(FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_11 FORMS14_12 , The Net Traffic Monitoring and Control 2014 ) 0.59375

(FORMS10_2 FORMS10_17 FORMS10_19 FORMS10_20 , Domain Specific Languages/Specification 2010 ) 0.5625

(09_13 00_4 97_4 95_2 , The Net Physical/Mathematical Model ) 0.5625

(FORMS10_18 FORMS14_4 FORMS14_8 FORMS12_12 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation MDE ) 0.5625

(97_2 95_5 FORMS10_3 , COTS ) 0.5

(95_5 , COTS 1995 ) 0.5

(95_11 , Scholarly Discussion 1995 Distributed Computing ) 0.5

(97_6 97_9 97_10 , Scholarly Discussion 1997 ) 0.5

(95_2 95_3 95_4 95_19 , Physical/Mathematical Model 1995 ) 0.5

(03_04 FORMS14_20 FORMS12_14 , FormalRiskAnalysis Scholarly Discussion ) 0.5

(FORMS10_16 , FormalRiskAnalysis Physical/Mathematical Model Descriptive Model 2010 ) 0.5

(95_8 95_18 95_19 , Virtual Train Sets + Automatic Guided 1995 ) 0.5

(97_9 , Discrete Time Model Continuous Time Model Scholarly Discussion 1997 ) 0.5

(FORMS10_20 , Domain Specific Languages/Specification Discrete Mathematical Model InstructionList Boolean SAT CDFG PLCs 2010 ) 0.5

(FORMS10_17 , Domain Specific Languages/Specification FormalRiskAnalysis Physical/Mathematical Model Matrices 2010 ) 0.5

(FORMS12_14 , 2012 FormalRiskAnalysis Scholarly Discussion ) 0.5

(00_1 97_4 95_8 , GA/AI/Fuzzy Neural Networks ) 0.5

(FORMS10_3 , Train Operation Markov Models COTS 2010 ) 0.5

(00_27 00_38 , Train Operation 2000 Physical/Mathematical Model ) 0.5

(00_38 , Train Operation 2000 Physical/Mathematical Model Descriptive Model ) 0.5

(FORMS14_3 , Train Operation FormalRiskAnalysis FTA Bayesian Networks 2014 ) 0.5

(00_10 , Train Operation Domain Specific Languages/Specification 2000 B-Method ) 0.5

(06_251 , Train Operation 2006 Scholarly Discussion ) 0.5

(00_18 , Train Operation GA/AI/Fuzzy 2000 ) 0.5

(FORMS14_22 , PetriNets Train Operation MDE 2014 ) 0.5

(97_3 , Freight Handling 1997 Least Squares ) 0.5

(06_331 , Freight Handling 2006 ) 0.5

(03_06 , Freight Handling Agents 2003 Virtual Train Sets + Automatic Guided ) 0.5

(00_14 , Freight Handling PetriNets 2000 Markov Models Analytical Model ) 0.5

(FORMS10_9 , Traffic Monitoring and Control ModelChecking 2010 ) 0.5

(FORMS14_21 , Traffic Monitoring and Control MDE HAZOP 2014 ) 0.5

(FORMS14_1 , Traffic Monitoring and Control Scholarly Discussion STAMP 2014 ) 0.5

(95_9 , Traffic Monitoring and Control Analytical Model 1995 ) 0.5

(97_2 , Traffic Monitoring and Control Descriptive Model 1997 COTS ) 0.5

(FORMS10_14 , Traffic Monitoring and Control Physical/Mathematical Model Matlab 2010 ) 0.5

(FORMS10_8 , Traffic Monitoring and Control Physical/Mathematical Model Analytical Model 2010 ) 0.5

(00_11 , Traffic Monitoring and Control 2000 STEP ) 0.5

(00_39 , Traffic Monitoring and Control 2000 Physical/Mathematical Model Scholarly Discussion ) 0.5

(00_13 00_29 , Traffic Monitoring and Control 2000 FTA ) 0.5

(FORMS10_11 FORMS10_12 , Traffic Monitoring and Control FormalRiskAnalysis Physical/Mathematical Model 2010 ) 0.5

(FORMS10_11 , Traffic Monitoring and Control FormalRiskAnalysis Physical/Mathematical Model SADT Model 2010 ) 0.5

(95_19 , Traffic Monitoring and Control Virtual Train Sets + Automatic Guided Physical/Mathematical Model 1995 ) 0.5

(00_34 , Traffic Monitoring and Control Virtual Train Sets + Automatic Guided 2000 Analytical Model ) 0.5

(97_5 , Traffic Monitoring and Control FMEA 1997 ) 0.5

(95_6 , Traffic Monitoring and Control FMEA Scholarly Discussion 1995 ) 0.5

(00_29 , Traffic Monitoring and Control FMEA 2000 FTA ) 0.5

(00_26 , Traffic Monitoring and Control Continuous Time Model 2000 Physical/Mathematical Model ) 0.5

(00_22 , Traffic Monitoring and Control Discrete Time Model 2000 Analytical Model Graph Theory ) 0.5

(00_2 , Traffic Monitoring and Control Discrete Time Model 2000 Physical/Mathematical Model ) 0.5

(FORMS10_6 , Traffic Monitoring and Control Discrete Time Model Continuous Time Model Markov Models Analytical Model 2010 ) 0.5

(FORMS10_19 , Traffic Monitoring and Control Domain Specific Languages/Specification InstructionList PLCs 2010 ) 0.5

(FORMS10_2 , Traffic Monitoring and Control Domain Specific Languages/Specification Lustre 2010 ) 0.5

(FORMS14_13 , Traffic Monitoring and Control Domain Specific Languages/Specification B-Method Discrete Mathematical Model 2014 ) 0.5

(00_5 00_9 , Traffic Monitoring and Control Domain Specific Languages/Specification 2000 Analytical Model ) 0.5

(00_9 , Traffic Monitoring and Control Domain Specific Languages/Specification 2000 Analytical Model VDM ) 0.5

(FORMS14_14 , Traffic Monitoring and Control Domain Specific Languages/Specification RAISESpecLanguage Scholarly Discussion 2014 ) 0.5

(00_8 , Traffic Monitoring and Control Domain Specific Languages/Specification RAISESpecLanguage 2000 Descriptive Model ) 0.5

(FORMS12_6 , Traffic Monitoring and Control 2012 EFS(ERTMSFormalSpec) ) 0.5

(FORMS12_15 , Traffic Monitoring and Control 2012 KroneckerAlgebra ) 0.5

(09_01 , Traffic Monitoring and Control 2009 Scholarly Discussion ) 0.5

(06_261 06_349 06_245 , Traffic Monitoring and Control 2006 ) 0.5

(06_349 , Traffic Monitoring and Control 2006 Scholarly Discussion ) 0.5

(03_04 , Traffic Monitoring and Control 2003 FormalRiskAnalysis Scholarly Discussion ) 0.5

(03_02 , Traffic Monitoring and Control 2003 Domain Specific Languages/Specification RAISESpecLanguage ) 0.5

(12_35 , Traffic Monitoring and Control Symbolic Algebra 2012 ) 0.5

(00_3 00_36 , Traffic Monitoring and Control Agents 2000 ) 0.5

(00_36 , Traffic Monitoring and Control Agents Discrete Time Model FMEA 2000 Analytical Model Discrete Event Systems ) 0.5

(06_261 , Traffic Monitoring and Control Agents 2006 FMEA Virtual Train Sets + Automatic Guided FormalRiskAnalysis ) 0.5

(95_8 95_17 , Traffic Monitoring and Control GA/AI/Fuzzy 1995 ) 0.5

(95_8 , Traffic Monitoring and Control GA/AI/Fuzzy Virtual Train Sets + Automatic Guided Neural Networks 1995 ) 0.5

(95_13 , Traffic Monitoring and Control Train Operation Scholarly Discussion 1995 ) 0.5

(97_11 , Traffic Monitoring and Control Train Operation Virtual Train Sets + Automatic Guided 1997 ) 0.5

(FORMS14_5 , Traffic Monitoring and Control PetriNets ModelChecking 2014 ) 0.5

(95_15 , Traffic Monitoring and Control PetriNets 1995 ) 0.5

(97_1 97_7 , Traffic Monitoring and Control PetriNets 1997 ) 0.5

(97_1 , Traffic Monitoring and Control PetriNets Analytical Model 1997 ) 0.5

(FORMS10_5 , Traffic Monitoring and Control PetriNets Physical/Mathematical Model 2010 ) 0.5

(12_34 FORMS12_8 , Traffic Monitoring and Control PetriNets 2012 ) 0.5

(12_34 , Traffic Monitoring and Control PetriNets 2012 Scholarly Discussion ) 0.5

(06_245 , Traffic Monitoring and Control PetriNets 2006 Discrete Event Systems ) 0.5

(03_07 , Traffic Monitoring and Control PetriNets GA/AI/Fuzzy 2003 ) 0.5

(95_20 , Traffic Monitoring and Control UML 1995 ) 0.5

(12_38 FORMS12_5 , Traffic Monitoring and Control UML 2012 ) 0.5

(FORMS12_5 , Traffic Monitoring and Control UML 2012 Abstract Syntax ) 0.5

(00_23 , Traffic Monitoring and Control Freight Handling Virtual Train Sets + Automatic Guided 2000 CASETool Descriptive Model ) 0.5

(FORMS14_20 , Scheduling and Allocation FormalRiskAnalysis Scholarly Discussion 2014 ) 0.5

(06_337 06_343 , Scheduling and Allocation 2006 ) 0.5

(00_1 , Scheduling and Allocation GA/AI/Fuzzy Neural Networks 2000 ) 0.5

(03_05 , Scheduling and Allocation GA/AI/Fuzzy 2003 ) 0.5

(06_343 , Scheduling and Allocation MILP GA/AI/Fuzzy 2006 MaxPlusAlgebra ) 0.5

(12_16 , Scheduling and Allocation Rolling Stock Freight Handling MILP 2012 Discrete Time Model ) 0.5

(97_6 , Scheduling and Allocation Traffic Monitoring and Control Scholarly Discussion 1997 ) 0.5

(97_8 , Scheduling and Allocation Traffic Monitoring and Control GA/AI/Fuzzy Analytical Model 1997 ) 0.5

(09_97 , Scheduling and Allocation Traffic Monitoring and Control UML PetriNets 2009 ) 0.5

(12_03 , Timetables MILP 2012 Continuous Time Model ) 0.5

(06_245 , Timetables PetriNets 2006 MaxPlusAlgebra ) 0.5

(FORMS10_7 , Timetables Traffic Monitoring and Control Discrete Mathematical Model 2010 ) 0.5

(FORMS14_16 , The Net Automata StateCharts 2014 ) 0.5

(97_10 , The Net Scholarly Discussion 1997 ) 0.5

(03_01 , The Net 2003 Domain Specific Languages/Specification RAISESpecLanguage ) 0.5

(97_4 , The Net GA/AI/Fuzzy Neural Networks Physical/Mathematical Model Analytical Model 1997 ) 0.5

(00_6 , The Net PetriNets 2000 Analytical Model ) 0.5

(12_04 , The Net PetriNets 2012 ) 0.5

(95_14 , The Net PetriNets Train Operation Descriptive Model Discrete Event Systems 1995 ) 0.5

(09_13 , The Net Rolling Stock LinearTimeInvariantSystem 2009 Physical/Mathematical Model ) 0.5

(FORMS14_11 , The Net Traffic Monitoring and Control Scholarly Discussion 2014 ) 0.5

(00_12 , The Net Traffic Monitoring and Control Continuous Time Model 2000 AST ) 0.5

(FORMS14_12 , The Net Traffic Monitoring and Control Domain Specific Languages/Specification RAISESpecLanguage 2014 ) 0.5

(FORMS12_10 , The Net Traffic Monitoring and Control 2012 STAMP ) 0.5

(03_03 , The Net Traffic Monitoring and Control 2003 StateTransitionSystems ) 0.5

(00_15 , The Net Traffic Monitoring and Control PetriNets 2000 ) 0.5

(95_2 , The Net Traffic Monitoring and Control Passenger Handling Freight Handling Physical/Mathematic Model Scholarly Discussion 1995 ) 0.5

(00_7 00_28 00_35 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling 2000 Scholarly Discussion ) 0.5

(00_4 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling 2000 Physical/Mathematical Model Descriptive Model ) 0.5

(FORMS10_13 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation ACSL(Frama-C) 2010 ) 0.5

(FORMS14_6 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation Live Sequence Charts Discrete Mathematical Model eLSCLanguage Temporal Logic 2014 ) 0.5

(FORMS14_8 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation Domain Specific Languages/Specification MDE FailureLogicModelling XML 2014 ) 0.5

(FORMS14_4 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation Domain Specific Languages/Specification MDE iglosTerminologyModel 2014 ) 0.5

(FORMS12_12 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation 2012 MDE Tool Chains ) 0.5

(FORMS10_18 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling UML Train Operation MDE Tool Chains 2010 ) 0.5

(12_16 00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Scheduling and Allocation Rolling Stock Freight Handling ) 0.49951171875

(09_13 00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , The Net Rolling Stock ) 0.49951171875

(00_28 00_35 95_14 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , The Net Train Operation ) 0.4990234375

(00_4 00_23 00_7 00_28 00_35 95_2 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Traffic Monitoring and Control Freight Handling ) 0.498046875

(00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_7 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Timetables Traffic Monitoring and Control ) 0.497802734375

(00_4 00_7 00_28 00_35 95_2 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , The Net Traffic Monitoring and Control Passenger Handling Freight Handling ) 0.497802734375

(95_1 95_2 95_6 95_7 95_11 95_12 95_13 , Scholarly Discussion 1995 ) 0.4921875

(FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_11 FORMS14_12 FORMS14_16 , The Net 2014 ) 0.484375

(FORMS14_2 FORMS14_5 FORMS14_15 FORMS14_19 FORMS14_22 , PetriNets 2014 ) 0.46875

(00_7 00_28 00_35 97_6 FORMS14_20 , Scheduling and Allocation Scholarly Discussion ) 0.46875

(00_1 00_4 00_7 00_28 00_35 , Scheduling and Allocation 2000 ) 0.46875

(00_7 00_28 00_35 95_2 FORMS14_11 , The Net Traffic Monitoring and Control Scholarly Discussion ) 0.46875

(00_14 00_30 00_40 FORMS10_3 FORMS10_6 , Markov Models ) 0.4375

(00_5 00_8 00_9 00_10 , Domain Specific Languages/Specification 2000 ) 0.4375

(FORMS10_1 FORMS10_3 FORMS10_13 FORMS10_18 , Train Operation 2010 ) 0.4375

(06_251 00_28 00_35 95_13 , Train Operation Scholarly Discussion ) 0.4375

(00_4 00_8 00_23 97_2 , Traffic Monitoring and Control Descriptive Model ) 0.4375

(00_4 00_23 00_7 00_28 00_35 , Traffic Monitoring and Control Freight Handling 2000 ) 0.4375

(FORMS14_4 FORMS14_6 FORMS14_8 FORMS14_20 , Scheduling and Allocation 2014 ) 0.4375

(00_7 00_28 00_35 97_6 , Scheduling and Allocation Traffic Monitoring and Control Scholarly Discussion ) 0.4375

(FORMS10_1 FORMS10_7 FORMS10_13 FORMS10_18 , Timetables Traffic Monitoring and Control 2010 ) 0.4375

(00_7 00_28 00_35 95_2 , The Net Traffic Monitoring and Control Passenger Handling Freight Handling Scholarly Discussion ) 0.4375

(00_4 00_7 00_28 00_35 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling 2000 ) 0.4375

(03_02 03_01 00_8 FORMS14_12 FORMS14_14 , Domain Specific Languages/Specification RAISESpecLanguage ) 0.40625

(00_4 00_38 FORMS10_16 , Physical/Mathematical Model Descriptive Model ) 0.375

(00_13 00_29 FORMS14_3 , FTA ) 0.375

(00_4 00_8 00_23 00_38 , 2000 Descriptive Model ) 0.375

(00_14 00_30 00_40 , 2000 Markov Models Analytical Model ) 0.375

(09_01 09_13 09_97 , 2009 ) 0.375

(06_261 03_06 00_3 00_36 , Agents ) 0.375

(FORMS10_18 FORMS14_4 FORMS14_8 FORMS12_12 FORMS14_22 , Train Operation MDE ) 0.375

(FORMS14_4 FORMS14_8 FORMS14_22 , Train Operation MDE 2014 ) 0.375

(00_10 FORMS14_4 FORMS14_8 , Train Operation Domain Specific Languages/Specification ) 0.375

(00_6 00_14 97_1 , PetriNets Analytical Model ) 0.375

(12_04 12_34 FORMS12_8 , PetriNets 2012 ) 0.375

(FORMS10_18 FORMS14_4 FORMS14_8 FORMS14_21 FORMS12_12 , Traffic Monitoring and Control MDE ) 0.375

(FORMS14_4 FORMS14_8 FORMS14_21 , Traffic Monitoring and Control MDE 2014 ) 0.375

(00_30 00_40 FORMS10_6 , Traffic Monitoring and Control Markov Models Analytical Model ) 0.375

(00_4 00_8 00_23 , Traffic Monitoring and Control 2000 Descriptive Model ) 0.375

(00_12 00_26 FORMS10_6 , Traffic Monitoring and Control Continuous Time Model ) 0.375

(00_22 00_36 FORMS10_6 , Traffic Monitoring and Control Discrete Time Model Analytical Model ) 0.375

(00_2 00_22 00_36 , Traffic Monitoring and Control Discrete Time Model 2000 ) 0.375

(00_5 00_8 00_9 , Traffic Monitoring and Control Domain Specific Languages/Specification 2000 ) 0.375

(00_28 00_35 95_13 , Traffic Monitoring and Control Train Operation Scholarly Discussion ) 0.375

(03_01 FORMS14_4 FORMS14_8 FORMS14_12 , The Net Domain Specific Languages/Specification ) 0.375

(12_04 FORMS12_10 FORMS12_12 , The Net 2012 ) 0.375

(FORMS14_4 FORMS14_8 FORMS14_12 , The Net Traffic Monitoring and Control Domain Specific Languages/Specification 2014 ) 0.375

(FORMS14_4 FORMS14_6 FORMS14_8 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation 2014 ) 0.375

(FORMS10_7 FORMS10_20 FORMS14_6 FORMS14_13 , Discrete Mathematical Model ) 0.3125

(FORMS10_7 FORMS10_20 , Discrete Mathematical Model 2010 ) 0.25

(FORMS14_4 FORMS14_8 FORMS14_21 FORMS14_22 , MDE 2014 ) 0.25

(00_36 95_14 06_245 , Discrete Event Systems ) 0.25

(97_1 97_4 97_8 , Analytical Model 1997 ) 0.25

(97_4 FORMS10_8 , Physical/Mathematical Model Analytical Model ) 0.25

(FORMS10_3 FORMS10_6 , Markov Models 2010 ) 0.25

(00_14 00_30 00_40 FORMS10_6 , Markov Models Analytical Model ) 0.25

(00_4 00_38 , 2000 Physical/Mathematical Model Descriptive Model ) 0.25

(FORMS14_3 FORMS14_20 , FormalRiskAnalysis 2014 ) 0.25

(97_9 FORMS10_6 , Discrete Time Model Continuous Time Model ) 0.25

(FORMS10_19 FORMS10_20 , Domain Specific Languages/Specification InstructionList PLCs 2010 ) 0.25

(FORMS10_20 FORMS14_13 , Domain Specific Languages/Specification Discrete Mathematical Model ) 0.25

(00_10 FORMS14_13 , Domain Specific Languages/Specification B-Method ) 0.25

(06_251 06_349 , 2006 Scholarly Discussion ) 0.25

(06_343 06_245 , 2006 MaxPlusAlgebra ) 0.25

(03_02 03_01 , 2003 Domain Specific Languages/Specification RAISESpecLanguage ) 0.25

(06_261 03_06 , Agents Virtual Train Sets + Automatic Guided ) 0.25

(97_4 97_8 , GA/AI/Fuzzy Analytical Model 1997 ) 0.25

(03_07 03_05 , GA/AI/Fuzzy 2003 ) 0.25

(00_38 95_14 , Train Operation Descriptive Model ) 0.25

(12_03 12_16 06_343 , MILP ) 0.25

(12_03 12_16 , MILP 2012 ) 0.25

(95_14 95_15 , PetriNets 1995 ) 0.25

(95_14 06_245 , PetriNets Discrete Event Systems ) 0.25

(00_6 00_14 00_15 , PetriNets 2000 ) 0.25

(00_6 00_14 , PetriNets 2000 Analytical Model ) 0.25

(06_245 06_245 , PetriNets 2006 ) 0.25

(95_14 FORMS14_22 , PetriNets Train Operation ) 0.25

(03_06 00_23 , Freight Handling Virtual Train Sets + Automatic Guided ) 0.25

(12_16 09_13 00_4 00_7 00_28 00_35 FORMS10_1 FORMS10_13 FORMS10_18 FORMS14_4 FORMS14_6 FORMS14_8 FORMS12_12 , Rolling Stock ) 0.25

(FORMS10_9 FORMS14_5 , Traffic Monitoring and Control ModelChecking ) 0.25

(FORMS10_7 FORMS14_6 FORMS14_13 , Traffic Monitoring and Control Discrete Mathematical Model ) 0.25

(FORMS14_6 FORMS14_13 , Traffic Monitoring and Control Discrete Mathematical Model 2014 ) 0.25

(FORMS14_1 FORMS12_10 , Traffic Monitoring and Control STAMP ) 0.25

(00_36 06_245 , Traffic Monitoring and Control Discrete Event Systems ) 0.25

(FORMS10_6 FORMS10_8 , Traffic Monitoring and Control Analytical Model 2010 ) 0.25

(97_1 97_8 , Traffic Monitoring and Control Analytical Model 1997 ) 0.25

(00_39 95_2 , Traffic Monitoring and Control Physical/Mathematical Model Scholarly Discussion ) 0.25

(95_8 95_19 , Traffic Monitoring and Control Virtual Train Sets + Automatic Guided 1995 ) 0.25

(00_23 00_34 , Traffic Monitoring and Control Virtual Train Sets + Automatic Guided 2000 ) 0.25

(00_29 00_36 , Traffic Monitoring and Control FMEA 2000 ) 0.25

(00_12 00_26 , Traffic Monitoring and Control Continuous Time Model 2000 ) 0.25

(00_2 00_22 00_36 FORMS10_6 , Traffic Monitoring and Control Discrete Time Model ) 0.25

(00_22 00_36 , Traffic Monitoring and Control Discrete Time Model 2000 Analytical Model ) 0.25

(FORMS10_2 FORMS10_19 , Traffic Monitoring and Control Domain Specific Languages/Specification 2010 ) 0.25

(FORMS14_12 FORMS14_14 , Traffic Monitoring and Control Domain Specific Languages/Specification RAISESpecLanguage 2014 ) 0.25

(09_01 09_97 , Traffic Monitoring and Control 2009 ) 0.25

(06_261 00_3 00_36 , Traffic Monitoring and Control Agents ) 0.25

(06_261 00_36 , Traffic Monitoring and Control Agents FMEA ) 0.25

(00_4 00_23 , Traffic Monitoring and Control Freight Handling 2000 Descriptive Model ) 0.25

(12_16 06_343 , Scheduling and Allocation MILP ) 0.25

(12_16 FORMS12_12 , Scheduling and Allocation Rolling Stock Freight Handling 2012 ) 0.25

(97_6 97_8 , Scheduling and Allocation Traffic Monitoring and Control 1997 ) 0.25

(09_97 FORMS10_18 , Scheduling and Allocation Traffic Monitoring and Control UML ) 0.25

(12_03 FORMS12_12 , Timetables 2012 ) 0.25

(FORMS10_7 FORMS14_6 , Timetables Traffic Monitoring and Control Discrete Mathematical Model ) 0.25

(95_2 95_14 , The Net 1995 ) 0.25

(97_4 97_10 , The Net 1997 ) 0.25

(00_6 97_4 , The Net Analytical Model ) 0.25

(00_4 95_14 , The Net Descriptive Model ) 0.25

(03_01 FORMS14_12 , The Net Domain Specific Languages/Specification RAISESpecLanguage ) 0.25

(03_03 03_01 , The Net 2003 ) 0.25

(00_6 00_15 , The Net PetriNets 2000 ) 0.25

(09_13 00_4 , The Net Rolling Stock Physical/Mathematical Model ) 0.25

(FORMS12_10 FORMS12_12 , The Net Traffic Monitoring and Control 2012 ) 0.25

(00_4 95_2 , The Net Traffic Monitoring and Control Passenger Handling Freight Handling Physical/Mathem Model ) 0.25

(FORMS10_18 FORMS12_12 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation MDE Tool Chains ) 0.25

(FORMS14_4 FORMS14_8 , The Net Timetables Scheduling and Allocation Traffic Monitoring and Control Rolling Stock Passenger Handling Freight Handling Train Operation Domain Specific Languages/Speci
MDE 2014 ) 0.25

(FORMS10_18 FORMS14_4 FORMS14_8 FORMS14_21 FORMS12_12 FORMS14_22 , MDE ) 0.1875

# Bibliography


[1] *Proceedings of the Control in Transportation Systems for The International Federation of Automatic Control (IFAC 1974 - 2013)*, The International Federation of Automatic Control, Amsterdam, Netherlands, 1975-. Elsevier.

[2] Alain Abran, Pierre Bourque, Robert Dupuis, and James W Moore. *Guide to the software engineering body of knowledge-SWEBOK*. IEEE Press, 2001.

[3] Aristidis Arageorgis and Aristides Baltas. Demarcating technology from science: Problems and problem solving in technology. *Zeitschrift für allgemeine Wissenschaftstheorie*, 20(2):212–229, 1989.

[4] Gabriela Arévalo, Stéphane Ducasse, and Oscar Nierstrasz. Lessons learned in applying formal concept analysis to reverse engineering. In *Formal Concept Analysis*, pages 95–112. Springer, 2005.

[5] Robert L. Baber. Comparison of electrical engineering of heaviside's times and software engineering of our times. *Annals of the History of Computing, IEEE*, 19(4):5–17, 1997.

[6] Carlo Batini, Maurizio Lenzerini, and Shamkant B. Navathe. A comparative

analysis of methodologies for database schema integration. *ACM computing surveys (CSUR)*, 18(4):323–364, 1986.

[7] Dines Bjørner. Formal software techniques in railway systems. In *9th IFAC Symposium on Control in Transportation Systems*, pages 1–12. VDI/VDE-Gesellschaft Mess-und Automatisieringstechnik, VDI-Gesellschaft fur Fahrzeug-und Verkehrstechnik. Technical University, Braunschweig, Germany, 2000.

[8] Dines Bjørner. Train: The railway domain. http://euler.fd.cvut.cz/railwaydomain/, 2005. Accessed: 2015-07-01.

[9] Gernot Böhme, Wolfgang Van Den Daele, Wolfgang Krohn, R Hohlfeld, and Wolf Schäfer. Finalization in science. *Social Science Information*, 15(2-3):307–330, 1976.

[10] Peter Burmeister. *Formal concept analysis with ConImp: Introduction to the basic features*. Fachbereich Mathematik, Technische Universität Darmstadt, 2003.

[11] Aleksey Buzmakov, Sergei O Kuznetsov, and Amedeo Napoli. Is concept stability a measure for pattern selection? *procedia computer science*, 31:918–927, 2014.

[12] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res.(JAIR)*, 24:305–339, 2005.

[13] Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In *Concept Lattices*, pages 189–207. Springer, 2004.

[14] Geza Tarnai Eckehard Schnieder, editor. *Proceedings of the Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT, 2010)*, FORMS/FORMAT, Berlin, Germany, 1998-. Springer.

[15] Alessandro Fantechi. Twenty-five years of formal methods and railways: What next? In *Software Engineering and Formal Methods*, pages 167–183. Springer, 2014.

[16] Bernhard Ganter, Gerd Stumme, and Rudolf Wille. *Formal concept analysis: foundations and applications*, volume 3626. Springer Science & Business Media, 2005.

[17] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.

[18] Robert L. Glass, Iris Vessey, and Venkataraman Ramesh. Research in software engineering: an analysis of the literature. *Information and Software technology*, 44(8):491–506, 2002.

[19] Stefan Gruner, Anne Elisabeth Haxthausen, Tom Maibaum, and Markus Roggenbach. Towards a formal methods body of knowledge for railway control and safety systems: Fm-rail-bok workshop 2013. Technical report, Technical University of Denmark, 2013.

[20] Armando Martin Haeberer and TSE Maibaum. Scientific rigour, an answer to a pragmatic question: a linguistic framework for software engineering. In

*Proceedings of the 23rd International Conference on Software Engineering*, pages 463–472. IEEE Computer Society, 2001.

[21] Anne Haxthausen, Markus Roggenbach, Stefan Gruner, and Tom Maibaum. Ssfmgroup. https://ssfmgroup.wordpress.com/rel/, 2013. Accessed: 2015-05-22.

[22] KY Hu, YC Lu, and CY Shi. Advances in concept lattice and its application. *Journal of Tsinghua University (Science and Technology)*, 9:019, 2000.

[23] Mikhail Klimushkin, Sergei Obiedkov, and Camille Roth. Approaches to the selection of relevant concepts in the case of noisy data. In *Formal concept analysis*, pages 255–266. Springer, 2010.

[24] Apurva Kumar. Fca results analysis. https://github.com/hisana/ConceptLatticeAnalysis, 2015. Accessed: 2015-07-01.

[25] Apurva Kumar. Which subdomains are in the railway domain? - researchgate. https://www.researchgate.net/post/Which_Subdomains_are_in_the_Railway_Domain, 2015. Accessed: 2015-07-01.

[26] Sergei O Kuznetsov. On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):101–115, 2007.

[27] Sergei O. Kuznetsov and Dmitry I. Ignatov. Concept stability for constructing taxonomies of web-site users. *CoRR*, abs/0905.1424, 2009.

[28] Stéphane Lopes, Jean-Marc Petit, and Lotfi Lakhal. Functional and approximate dependency mining: database and fca points of view. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):93–114, 2002.

[29] Tom Maibaum. Formal methods versus engineering. *ACM SIGCSE Bulletin*, 41(2):6–12, 2009.

[30] Tom Maibaum. What is a bok? large–extended abstract–. In *Software Engineering and Formal Methods*, pages 184–188. Springer, 2014.

[31] Sergei Obiedkov. Modeling preferences over attribute sets in formal concept analysis. In *Formal Concept Analysis*, pages 227–243. Springer, 2012.

[32] Frank R Pfetsch. The'finalization'debate in germany: Some comments and explanations. *Social studies of science*, 9(1):115–124, 1979.

[33] Hans Poser. On structural differences between science and engineering. 1998.

[34] Uta Priss. Linguistic applications of formal concept analysis. In *Formal Concept Analysis*, pages 149–160. Springer, 2005.

[35] Uta Priss. Formal concept analysis in information science. *Annual review of information science and technology*, 40(1):521–543, 2006.

[36] Uta Priss. Formal concept analysis homepage. http://www.fcahome.org.uk/fcaintro.html, 2007. Accessed: 2015-07-01.

[37] Dr. rer. nat. Serhiy Yevtushenko. ConExp: Concept Explorer. online, 2006. http://conexp.sourceforge.net/.

[38] William E Riddle. The magic number eighteen plus or minus three: a study of software technology maturation. *ACM SIGSOFT Software Engineering Notes*, 9(2):21–37, 1984.

[39] Camille Roth and Paul Bourgine. Lattice-based dynamic and overlapping taxonomies: The case of epistemic communities. *Scientometrics*, 69(2):429–447, 2006.

[40] Camille Roth, Sergei Obiedkov, and Derrick Kourie. Towards concise representation for taxonomies of epistemic communities. In *Concept Lattices and Their Applications*, pages 240–255. Springer, 2008.

[41] Mary Shaw. The coming-of-age of software architecture research. In *Proceedings of the 23rd international conference on Software engineering*, page 656. IEEE Computer Society, 2001.

[42] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2013.

[43] Gerd Stumme. Using ontologies and formal concept analysis for organizing business knowledge. In *Wissensmanagement mit Referenzmodellen*, pages 163–174. Springer, 2002.

[44] Gerd Stumme and Alexander Maedche. Fca-merge: Bottom-up merging of ontologies. In *IJCAI*, volume 1, pages 225–230, 2001.

[45] Walter G. Vincenti. *What Engineers Know and How they Know It: Analytical Studies from Aeronautical History*. Johns Hopkins University Press, 1990.

[46] Rudolf Wille. Formal concept analysis as mathematical theory of concepts and concept hierarchies. In *Formal Concept Analysis*, pages 1–33. Springer, 2005.

[47] Serhiy A Yevtushenko. System of data analysis concept explorer. In *Proceedings of the 7th national conference on Artificial Intelligence KII*, volume 2000, 2000.