Exploiting a Perdurantist Foundational Ontology and Graph Database for Semantic Data Integration

A thesis submitted towards the degree of

Doctor of Philosophy

By

George Foy

Business School

Brunel University London

November 2015

ABSTRACT

The view of reality that is inherent to perdurantist philosophical ontologies, often termed four dimensional (4D) ontologies, has not been widely adopted within the mainstream of information system design practice. However, as the closed world of enterprise systems is opened to Internet scale Semantic Web and Open Data information sources, there is a need to better understand the semantics of both internal and external data and how they can be integrated. Philosophical foundational ontologies can help establish this understanding and there is, therefore, an emerging need to research how they can be applied to the problem of semantic data integration. Therefore, a prime objective of this research was to develop a framework through which to apply a 4D foundational ontology and a graph database to the problem of semantic data integration, and to assess the effectiveness of the approach. The research employed design science, a methodology which is applicable to undertaking research within information systems as it encompasses methods through which the research can be undertaken and the resultant artefacts evaluated. This methodology has a number of discrete stages: problem awareness; a core design-build-evaluate iterative cycle through which the research is conducted; and a conclusion stage. The design science research was conducted through the development of a number of artefacts, the prime being the 4D-Semantic Extract Load (4D-SETL) framework. The effectiveness of the framework was assessed by applying it to semantically interpret and integrate a number of large scale datasets and to instantiate a prototype graph database warehouse to persist the resultant ontology. A series of technical experiments confirmed that directly reflecting the model patterns of 4D ontology within a prototype data warehouse proved an effective means of both structuring and semantically integrating complex datasets and that the artefacts produced by 4D-SETL could function at scale. Through illustrative scenario, the effectiveness of the approach is described in relation to the ability of the framework to address a number of weaknesses in current approaches. Furthermore the major advantages of the 4D-SETL are elaborated; which include ability of the framework is to combine foundational, domain and instance level ontological models in a single coherent system that dispensed with much of the translation normally undertaken between conceptual, logical and physical data models. Additionally, adopting a perdurantist realist foundational ontology provided a clear means of establishing and maintaining the identity of physical objects as their constituent temporal and spatial parts unfold over the course of time.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors Dr Sergio de Cesare and Professor Mark Lycett and thank them for their guidance and support during my research.

I would like also like to thank Chris Partridge and the many other people from Brunel University that have helped and supported me during the course of my studies.

Finally I would also like to thanks my wife for her patience and understanding and the inspiration to undertake this PhD in the first place.

PUBLICATIONS

The work in this thesis has led to the following publication:

de Cesare, S.; Foy, G.; Partridge, C. 2013. ‘Re-engineering Data with 4D Ontologies and Graph Databases’, Proceedings of the Advanced Information Systems Engineering Workshops Springer pp 304-16.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3D: Three Dimensional (pertaining to endurantist ontologies)

4D: Four Dimensional (pertaining to perdurantist ontologies)

ABOX: Assertional elements of an ontology

ACID: Atomicity, Consistency, Isolation and Durability (pertaining to transactions)

AI: Artificial Intelligence

API: Application Programme Interface

BORO: Business Object Reference Ontology

BORO UML: Universal Modelling Language Profile of Business Object Reference Ontology

BPMN: Business Process Modelling Notation

BFO: Basic Formal Ontology

CDM: Conceptual Data Model

DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering

DL: Description Logics

DR: Design Research

DS: Design Science

DSM: Domain Specific Model

DSRM: Design Science Research Methodology

ER: Entity Relationship (model)

FOL: First Order Logic

GPR: Geo-Political Region

HOL: Higher Order Logic

IDE: Integrated Development Environment

IS: Information System

LDM: Logical Data Model

LOD: Linked Open Data

NoSQL: Not Only SQL

OD: Open data

ODISE: Ontology-Driven Information Systems Engineering

OMG: Object Management Group

OO: Object Oriented (software)

ONS: Office of National Statistics

ONSPD: Office of National Statistics Postcode Directory

OP: Object (4D) Paradigm

OWL: Web Ontology Language

PDM: Physical Data Model

RDBMS: Relational Database Management System

RDF: Resource Description Framework

RDFS: Resource Description Framework Schema

SIC 2007: Standard Industry Codes 2007 Revision

SPARQL: SPARQL Protocol and RDF Query Language

STE: Spatio-Temporal Extent (4D Object)

TBOX: Terminological elements of an ontology

TDD: Test Driven Development

UK: United Kingdom

UML: Universal Modelling Language

UNA: Unique Name Assumption

UNSPSC: United Nations Standard Products and Services Code

URN: Unique Reference Number

W3C: World Wide Web Consortium

XMI: Extensible Mark-up Interchange Language

XML: Extensible Mark-up Language

CHAPTER 1. INTRODUCTION

1.1 Introduction

Following the advent of the Web, many organisations began to exploit the commercial potential of this new medium by extending internal business applications to provide Web access, thus enabling the rise of today's digital economy and the widespread use of electronic commerce in both a 'business to business' and 'business to consumer' context. Consequently, within the field of Information Systems, Web applications became the norm typically supported by an 'n' tiered architecture consisting of Web presentation, business logic, data storage and business systems integration layers. The advent of the Semantic Web (Berners-Lee, Hendler and Lassila, 2001), again, offers organisations that have the foresight, new opportunities to exploit the vast source of knowledge and market insight that the Semantic Web and Open Data represents. Integrating business applications with the Semantic Web will also impact current information systems architecture design and implementation practice.

The commercial adoption of the Semantic Web has begun; organisations can now employ standardised vocabularies such as Schema.org (Google, Inc., Yahoo, Inc., and Microsoft Corporation, 2015) to provide semantic mark-up to describe the information they present via their Web pages. Consequently, organisations now face the task of linking their internal information to these 'global schemas', which, in turn, leads to a fundamental question relating to the nature of the data they hold and what they represent. These data relate to a model, an abstraction of reality that was designed with reference to a wide range of implicit and explicit commitments that underlie what the domain model represents (Chandrasekaran, Josephson and Benjamins, 1999). Consequently, when integrating data from one system to another, in addition to data, models are also being integrated together with the implicit and explicit semantics on which they are based. Therefore, as stated by Sheth (1999), achieving successful semantic integration will require more than a syntactic mapping process, it will require a shift from contemporary modelling and software development practices that are constrained to isolated application domains, to methods that employ accurate and consistent models of reality that enable the integration of both internal and external data sources.

However, a stated by Kent (1978), a fundamental problem that will need to be addressed is to establish what exactly these models represent, as when an IS designer creates a model of reality, they are dealing with a number of 'worlds' - the past historical world; a possible world which is based on conjecture (e.g., a what-if scenarios) or fiction; the concrete physical world; or the world as modelled by another mind with differing perceptions. The complexity of this situation is further compounded by the disparate theories which can be employed to model reality and confusion as to where the boundaries between the artefacts of computation, abstract models, language, concepts, signs and physical reality lie. The heterogeneous constraints and structures imposed by the various incompatible modelling tools and practices employed within software, database and Semantic Web community also act as a barrier to integration; as do the misunderstandings between the people involved in the design process.

Furthermore, on examination of the various models produced by these tools, it is rare to find any which are grounded in any form of overarching foundational theory (Cregan, 2007). This is important, as without a consistent ‘view’ of reality it is difficult to achieve a consistency within semantic integration. A foundational ontology is a model that is limited to concepts that are meta-ontological, generic, abstract and philosophical and therefore are general enough to model, at the highest level, a broad range of domain areas (Niles and Pease, 2001). It can therefore help resolve some of the semantic data integration challenges by providing a philosophical standpoint, a fixed frame of reference.

Thus a foundational ontology, when combined with a suitable development framework, can be applied to provide the ‘grounding’ in a fundamental view of reality through which an organisation can gain consistency when semantically interpreting data, developing specific domains of interest or interpreting and translating and integrating data that conform to differing theories. Furthermore, in addition to providing a consistent semantic view, the patterns drawn from a foundational ontology can be extended to form domain ontologies and reflected within an Information System to structure and store data within a database. As asserted by Partridge *et al.* (2013), due to the increasing use of semantics within Information Systems, there is an emerging need for understanding ontology in the philosophical sense. Geerts and McCarthy (2000) suggest that ontologies will be increasingly important as a means of providing consistent and machine-readable semantic definitions of economic phenomena that will become the language of Semantic Web based e-commerce.

1.2 **Research Aim and Objectives**

The aim of research was to examine the problem of semantic data integration and through the design and execution of a design science project test the hypothesis that a perdurantist foundational ontology and a graph database can be exploited to provide a semantic data integration solution that would prove effective in addressing a number of the weaknesses identified in current approaches. The research aim is achieved via objectives that are detailed in the following subsections.

1.2.1 The Problem of Sematic Data Integration and the Application of Foundational Ontology

The review of literature presented in Chapter two fulfils the first objective of the research, that of examining the problem of semantic data integration and the application of foundational ontologies to solve this problem. This review also sets the context of the research in terms of the exploitation of data derived from the Semantic Web, social media and other Open Data (OD) sources within business systems. Chapter two concludes with a summary of a number of weaknesses inherent in current practice.

1.2.2 The Research Question

Motivated by the need to look at new and innovative means of achieving semantic data integration the researcher hypothesised that a perdurantist (4D) foundational ontology and graph database could prove effective in addressing a number of the problems that are inherent to existing practice.

1.2.3 The Research Design and Execution

To address the research question required a research project be undertaken. Thus, an overarching design science research methodology was selected as a means of delivering the research aim and objectives. The general subject of Design Science is discussed within Chapter three together with how this methodology was adapted for the execution of this research project.

1.2.4 The 4D-SETL Framework Design

To test the hypothesis required the development of a framework. This objective was achieved through the development of the 4D Semantic Extract Transform and Load Framework (4D-SETL). This framework employs a perdurantist foundational ontology together with a graph database to perform semantic data integration. Thus 4D-SETL was developed to serves the aim of the research. The development of the 4D-SETL framework would be achieved during the course of the first iteration of the design science design-build-evaluate cycle which is discussed within Chapter four. The framework was also improved during the course of the two subsequent cycles (Chapters five and six).

1.2.5 The Application of the 4D-SETL Framework

In order to evaluate the effectiveness of the 4D-SETL and, consequently, the exploitation of the foundational perdurantist ontology the a graph database, the fourth objective was to apply the 4D-SETL framework during the course of the design science project to Extract, Transform and Load, and thus semantically integrate five experimental datasets of differing complexity and scale. The intended result of the application of the framework would be a prototype warehouse instantiation consisting of a graph database populated with an ontology that would consist of a foundation and domain ontologies that would include a large number of individual instances. This objective would be achieved via multiple applications of the design science cycle through which the 4D-SETL framework was to be evaluated and improved. These activities are discussed within chapters four through six.

1.2.6 Evaluation of the Effectiveness of the 4D-SETL

The fifth objective was to evaluate the 4D-SETL framework and the resultant prototype warehouse instantiation on the completion of each of the design science iterations to access the effectiveness of the approach. Therefore, during each of the design science iterations, technical experiment was employed to validate the utility of the approach, whilst illustrative scenario was employed to provide qualitative descriptive comparisons of the advantages of the approach. It is through these evaluation activities that the effectiveness of the 4D-SETL to partially solve the problem of semantic data integration is assessed. The effectiveness is assessed in relation to how the artefacts produced by the research address the weaknesses in the current approaches as detailed in Table 1.1.

	Weakness	Description
1	Lack of grounding	Many current models employed within information systems have no form of grounding in a more fundamental theory (Cregan, 2007). Thus the ontological commitments that underlie such models are unknown. On examination of many Linked Open Data ontologies, they are often ungrounded – or rely or reference other ungrounded ontologies.
2	Model strata and translations distortion	The strata of models formed by the software design process and the requirement to translate the high level models of reality that are created at the initial design time into a series of tabular statures that are focused of the execution environment can lose direct traceability to the initial model. This is analogues to the oft cited, within IS practice OO-RDBMS impedance mismatch (Ireland *et al.*, 2009).
3	Over simplification to fit a model of reality to a tractable theory	The need to simplify the abstraction of reality to it can fit neatly into a FOL theory, thus ignoring the fact that reality is not so simple and higher order objects exist (Bailey, 2011).
4	Integrating models which are founded on different semantics	There are many automatic translation techniques for translating RDBMS schema and data to an OWL ‘ontology’. However, there is a lack of recognition of the semantic differences that underlie the differing modelling constructs.
5	Dividing models into static and dynamic types	The separation of static and dynamic aspects of reality into different structural and process models leads to the development of incompatible abstractions together with exotic relations such as trans-ontological relations that are employed to bridge these static and dynamic worlds.
6	Naming and meaning confusion	That there is often naming and meaning confusion, as described by Frege (1948). The objects place in reality (and the ontology) define its meaning.
7	Establishing identify	Many modelling and information systems use ephemeral means of establishing an objects identity which do not function well over time.
8	Employing techniques that do not scale	Many of the software tools such as OWL tableau calculus based reasoners, as they are constrained by memory, cannot scale to inference over ontologies containing large scale instance population (Bock *et al.*, 2008).

Table 1.1: Weaknesses In Current Approaches

On completion of final design science iteration, the ontology developed is assessed using the criteria proposed by Gruber (1995).

1.3 Research Overview

Through the design and execution of a design science research project, this work examines the process of extracting the semantics from five existing datasets, translating the datasets to conform to the ontological patterns drawn from a perdurantist philosophical ontology, and incrementally loading the resultant domain ontologies (types and instances) to a graph database. This research is important as, whilst endurantist ontologies have been widely studied in the fields of Artificial Intelligence (AI) and medical research, limited research has been conducted on how frameworks employing perdurantist (4D) ontologies and graph databases can be exploited to achieve semantic integration. Graph database systems are an emerging technology with the ability to scale to encompass many billions of nodes and edges and are therefore applicable to data warehouse applications. Graph databases are particularly suitable for integration as models and instance level data may be added without the need to migrate existent information which is the norm for RDBMS based systems. They also offer a means of storing the ontological (model) and instance (individual instances) within a single environment that maintains the structural patterns of such models without the usual translation to tabular storage. The particular graph database implementation being employed by this research has the ability to store many billions of nodes and employs a directed parameter graph model which offers key-value storage at each node and edge which enables data to be stored, indexed and retrieved efficiently.

Investigating the effectiveness of applying a foundational ontology and graph database to the problem of semantic data integration represents a challenging and complex problem that will provide new insight into the use of 4D ontologies to model and integrate data related to a range of real-world entities. Furthermore, as the 4D-SETL framework is novel in that differs significantly from current information systems solutions, it is hoped the work, at some future date, will contribute to the corpus of knowledge related to Ontology-Driven Information Systems Engineering (ODISE).

1.4 Design Science Research

This project is a design science research project which March and Smith (1995) define as a method via which the boundaries of human and organisational capabilities can be extended through the development of innovative artefacts. This project relates to the design of information system which Hevner *et al*. (2004) describe as systems that capture and process information in support of human purposes that typically consist of complex organisations of hardware, software, procedures, data and people. To ensure that the project constitutes valid design science research, a recognised methodology is employed to provide the concise guidelines for executing such research. The methodology, as elaborated by Vaishnavi and Kuechler (2004), which builds on the work of March and Smith (1995) and others was adopted. This methodology was selected as it is particularly suitable to the purposes of this research as it provides a well-defined conceptual framework through which the design science project activities can be designed, executed and the results evaluated. Thus the core research is focused on the implementation of three design, build and evaluation iterations that form the primary source of the new knowledge produced by this research. An outline of the research iterations is depicted in Figure 1.1.

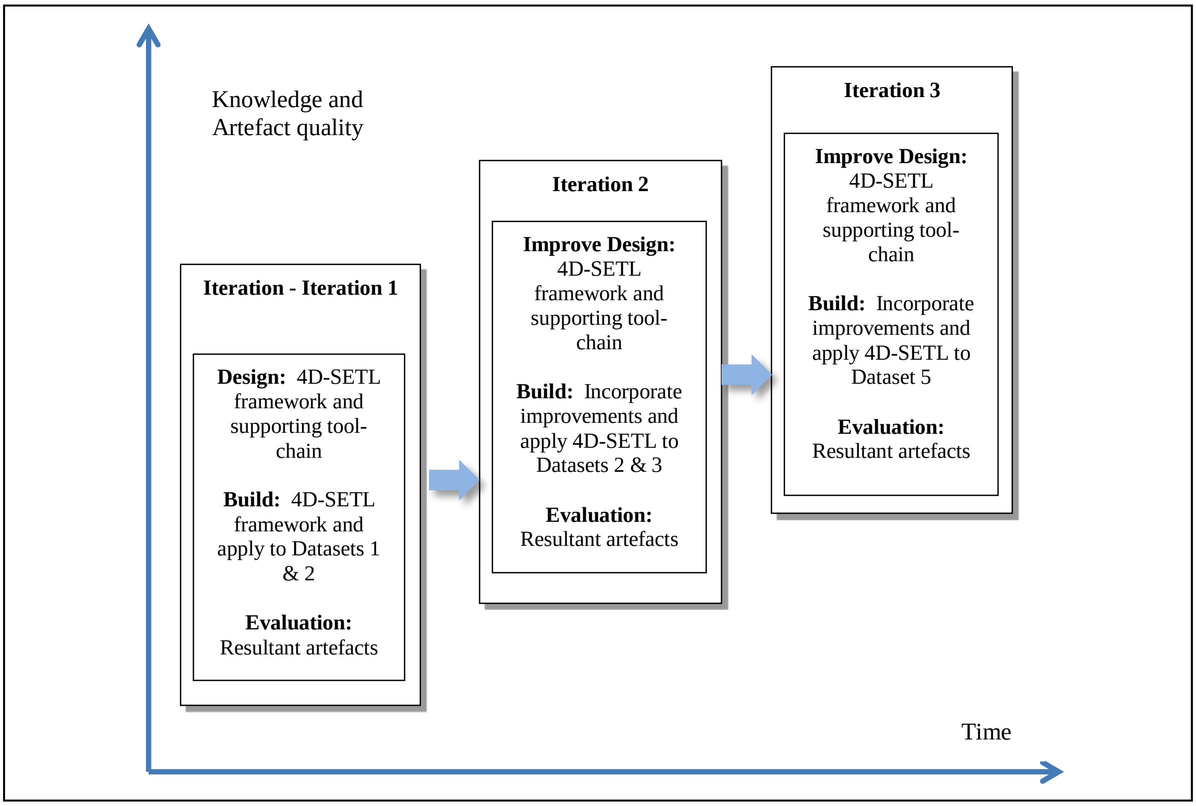

Knowledge and Artefact quality
Iteration - Iteration 1
Design: 4D-SETL framework and supporting tool-chain
Build: 4D-SETL framework and apply to Datasets 1 & 2
Evaluation: Resultant artefacts
Iteration 2
Improve Design: 4D-SETL framework and supporting tool-chain
Build: Incorporate improvements and apply 4D-SETL to Datasets 2 & 3
Evaluation: Resultant artefacts
Iteration 3
Improve Design: 4D-SETL framework and supporting tool-chain
Build: Incorporate improvements and apply 4D-SETL to Dataset 5
Evaluation: Resultant artefacts
Time

Figure 1.1: The Design Science - Design, Build, Evaluate Cycle

1.5 Thesis Structure

1.5.1 Literature Review

The aim of chapter two is to explore existing research literature related to the subject under investigation which is the problem of semantic data integration.

The review firstly examines the motivation for the interest in this subject by setting the context within the wider perspective of Semantic Web. The literature related to semantic data integration is then explored followed by that of the disparate modelling paradigms employed within the Semantic Web, Object Oriented (OO) software development and Relation Data Base Management System (RDBMS) design communities. The philosophical foundations of endurantist and perdurantist ontologies are then discussed. This aims to provide the reader with an outline understanding of the ontological commitments – the basic metaphysical beliefs that underlie foundational ontologies. Based on previous literature, a justification for the adoption of perdurantism is also provided. Subsequently, literature relating to the emerging polyglot persistence Fowler (2012) architecture is presented with a focus on graph databases. Such a database offers ability to directly mirror the ontological patterns derived from the application of the 4D-SETL framework within a data warehouse without the normal translation to OO or RDBMS format.

1.5.2 Design Science Research Methodology

Chapter three introduces the reader to the recognised research process adopted by this project; namely, the design science research methodology. The use of design science within the field of Information Systems (IS) is then discussed. Finally, a description of how the general design science research methodology was adapted and applied for the specific purpose of undertaking this research project is provided. This is in the form of an outline of the major activities undertaken during the course of the research project and their outputs. Consequently, the remaining sections of Chapter three mirror the stages of the design science research methodology itself. Design science differs from natural science in that it is concerned with engineered artificial entities rather that naturally occurring phenomena (Simon, 1996). However, in a similar manner to that of the investigation of a physical law, a researcher can use their intuition to theorise that a certain new type of artefact design will be better in some way than its predecessors. Furthermore, through the realisation of a design into an artefact that can be subject to evaluation, it is possible to decide whether this is true. However, there are many types of artefacts that can be created during course of such a design science investigation and many ways that such artefacts can be assessed. Therefore literature is consulted for guidance as to which evaluation methods to apply to such artefacts and to provide the rationale and justification for the use of a particular method (Peffers *et al.*, 2007).

1.5.3 First Iteration of the Design Science Cycle – 4D-SETL Development and Application

Chapter four provides an overview of the first iteration of the design science cycle during which the 4D-SETL framework and the technical architecture are realised. Chapter four also provides details of the first application of the framework, the resultant artefacts and the knowledge outputs that are used to evaluate the results. These findings are employed to inform the subsequent Design Science iterations. Rather than having discrete stages, the development and application of the 4D-SETL framework took place together as a series of sub-iterations during the course of this cycle as tools and techniques were either adapted from existing software or custom developed to meet the needs of the framework. The first two experimental datasets that are semantically integrated via the 4D-SETL framework to provide temporal and spatial location that are in the form of calendar and geographic domain ontologies. The calendar ontology consists of all the date information from 1862 to the present and a geographic ontology consisting of approximately 2.5 million postcode locations. The results of the iteration are evaluated via technical experiment and illustrative scenario which serve the research aim of evaluating the effectiveness of exploiting perdurantist ontology and a graph database for semantic integration.

1.5.4 Second Iteration of the Design Science Cycle – 4D-SETL Improvement and Application to Larger Complex Datasets

Chapter five provides a description of the second iteration of the design science research cycle. The objectives of the second iteration was to improve and develop the 4D-SETL framework by firstly incorporating the design improvements identified during the initial iteration and secondly to applying the framework to datasets that differ in structure, scale and complexity. These objectives were met by applying the 4D-SETL framework to semantically integrate a dataset representing the UK Standard Industry Classification Version 2007 (SIC 2007) and a large Open Dataset representing all UK companies. The first dataset SIC 2007 is interpreted and modelled as a taxonomic classification system consisting of higher order taxonomic ranks and 'instance level' data that is loaded to the warehouse that consists exclusively of taxonomic classes i.e. no individual instances level data. The second dataset consists of a large Open Data set sourced from Companies House consisting of a complete set of records representing all UK based companies consisting of approximately 3.5 million records. This process again serves the aim of the research by enabling an assessment of the 4D-SETL approach through technical experiment and illustrative scenario. The chapter details the research knowledge output to inform the subsequent final iteration.

1.5.5 Third Iteration of the Design Science Cycle – Final 4D-SETL Improvement and Application to Integrate Final Dataset

Chapter six provides an overview of the third and final iteration of the three design, build and evaluation cycles. This chapter describes details of the 4D-SETL framework to integrate a dataset representing all UK company officers. This experiment again confirms the scalability of the 4D-SETL framework approach by integrating a very large dataset representing 12 million records and performing a data retrieval graph traversal queries to validate the accuracy of the results. The experiment therefore also assesses the accuracy of information retrieved and the performance of the query (graph traversal). The final 'lessons learned' provide the knowledge output of the iteration and concludes the experiment.

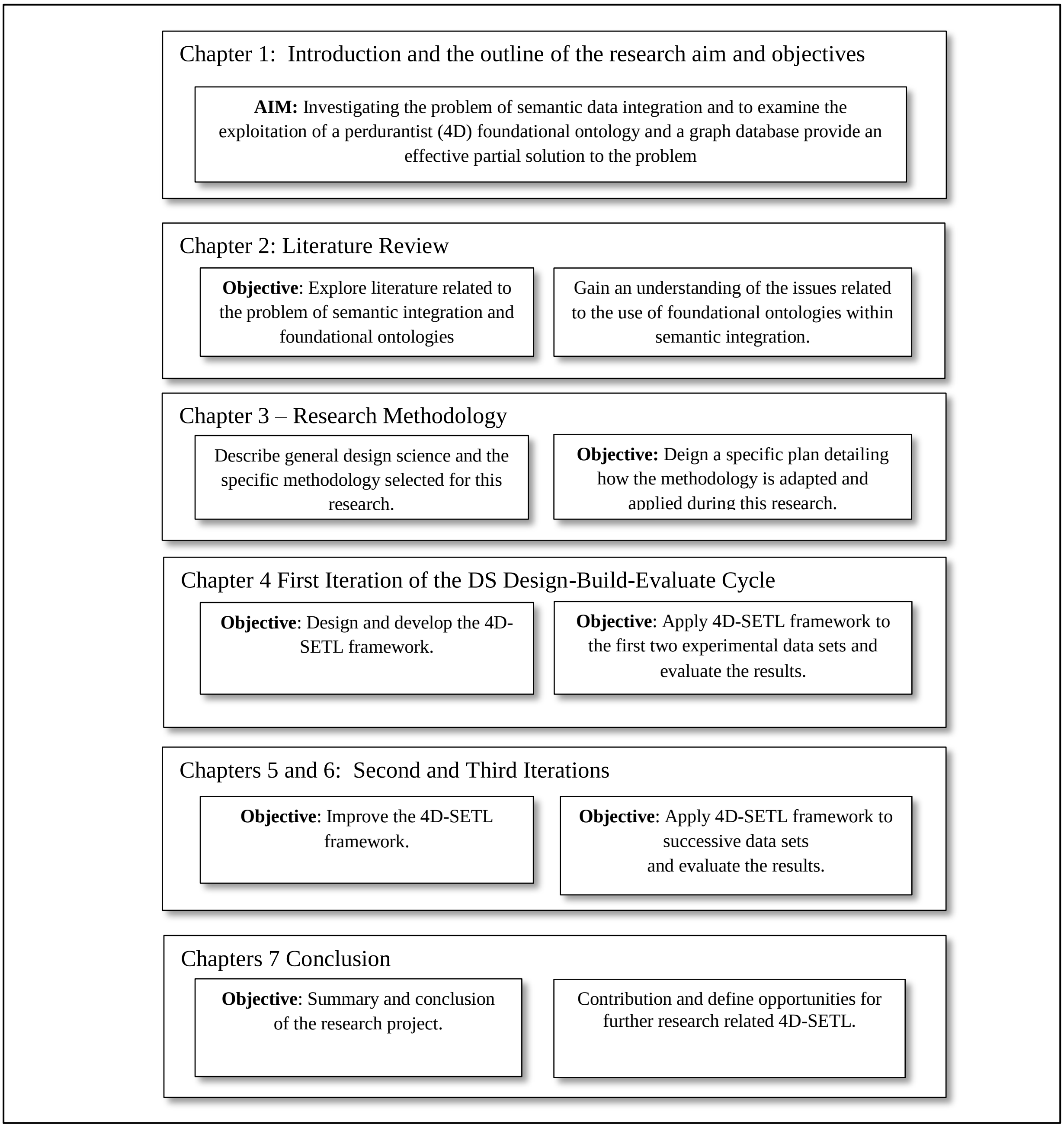
1.5.6 Summary and Conclusion

Chapter seven provides the summary and conclusion of the thesis detailing, firstly, the degree to which the project met the original research aim and objectives. The key research findings are then presented in relation to the previous research in this area. The original contribution to the corpus of knowledge is also discussed, together with details of the limitations of the research.

Finally, proposals for further research are outlined including specific subject areas that would benefit from further study.

1.6 **Thesis Overview**

Figure 1.2 provides a diagrammatic overview of the thesis structure.



Chapter 1: Introduction and the outline of the research aim and objectives

AIM: Investigating the problem of semantic data integration and to examine the exploitation of a perdurantist (4D) foundational ontology and a graph database provide an effective partial solution to the problem

Chapter 2: Literature Review

Objective: Explore literature related to the problem of semantic integration and foundational ontologies

Gain an understanding of the issues related to the use of foundational ontologies within semantic integration.

Chapter 3 – Research Methodology

Describe general design science and the specific methodology selected for this research.

Objective: Deign a specific plan detailing how the methodology is adapted and applied during this research.

Chapter 4 First Iteration of the DS Design-Build-Evaluate Cycle

Objective: Design and develop the 4D-SETL framework.

Objective: Apply 4D-SETL framework to the first two experimental data sets and evaluate the results.

Chapters 5 and 6: Second and Third Iterations

Objective: Improve the 4D-SETL framework.

Objective: Apply 4D-SETL framework to successive data sets and evaluate the results.

Chapters 7 Conclusion

Objective: Summary and conclusion of the research project.

Contribution and define opportunities for further research related 4D-SETL.

Figure 1.2: Overview of the Thesis

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

This chapter presents literature relating to the problem of semantic data integration and describes how foundational ontologies and graph databases can be exploited to provide a partial solution to this problem. This chapter aims to provide the reader with background information that outlines the rationale for this research by relating the importance of the subject of semantic data integration and its context within the wider perspective of the exploitation of Open Data and the Semantic Web (Berners-Lee, Hendler and Lassila, 2001). This chapter also presents and critiques research literature related to the problem of semantic data integration, foundational ontology and ontology driven data integration. The chapter concludes with a description of the limitations within current and past research and sets the course for the overall research project. This chapter is organised as follows.

Section 2.2 provides background and context for the research in relation to the rise of the semantic web and open data and the opportunities and challenges this will present to IS practitioners.

Section 2.3 describes the problem of semantic integration.

Section 2.4 provides a brief introduction to foundational ontologies and their grounding in more fundamental theories.

Section 2.5 describes the BORO foundational ontology and the object paradigm on which it is founded.

Section 2.6 provides a description of ontology driven semantic data integration.

Section 2.7 concludes the chapter and provides a summary of the literature findings together with the research direction and a number of the identified weaknesses inherent to existing semantic data integration practice.

2.2 Background

As the introduction of the Web and the widespread adoption of electronic commerce has led to major changes within Information Systems (IS) architecture, the rise of Semantic Web (Berners-Lee, Hendler and Lassila, 2001) will no doubt also lead to fundamental changes in the way IS, are designed, implemented and integrated. These changes will be necessitated by the transition from IS that process data that reside within local Relational Database Management Systems (RDBMS), to Internet connected systems that are able to access, integrate and process (big) data derived from the Semantic Web, social media and other Open Data (OD) sources. As a consequence of these changes, the current IS design and integration practices that tend to be limited to local applications and databases will need to embrace new techniques that will facilitate data integration on an Internet scale. However, this may prove problematic as many organisations already have information silos (Arsanjani, 2002), islands of applications that were not designed to interoperate or integrate, find integration demanding enough, integrating with externally sourced data represents a challenge on a different scale of difficulty.

Within a single organisation, there may be a high degree of agreement as to the vocabulary used to describe the universe of discourse; however, this is not the case when the scope of the integration includes information sourced from the Internet or other external sources. In addition, to compound the problem of heterogeneous vocabularies, system designers need not base their models (schemas or ontologies) on any agreed set of common terminological, representational or logical theory (DL) standards (Smart and Engelbrecht, 2008).

Consequently, there are many different models of the world, ranging in sophistication and expressivity from simple meta-tags to complex ontologies conforming to consistent formal logical theories.

Systems designers will also face the challenge of introducing Semantic Web modelling practices and languages such as the Resource Description Language Schema (RDFS) (Brickley and Guha, 2000) and Web Ontology Language (OWL) (Motik *et al.*, 2009), which are based on semantics which differ significantly from contemporary modelling languages used within information systems design. Thus, there exists a multitude of model types that can represent the same domain and many model elements that can ‘stand-in’ for the same real world individual entities. However, out of this rather chaotic situation, through self-organising phenomena such as preferential attachment (Barabási and Albert, 1999), together with the influence of Schema.org (Google, Inc., Yahoo, Inc., and Microsoft Corporation, 2015) de-facto standard vocabularies are emerging that can be employed to describe the entities and their relationships within a domain of interest. Thus, the advent of Schema.org may be considered a juncture in the development of the Semantic Web similar in nature to the release of the NCSA Mosaic Browser (Andreessen and Bina, 1994), which was a major contributing factor to the transition of the Web from the academic research community to becoming part of the fabric of everyday life.

Supplementing the terminological standardisation work being undertaken by Schema.org, governments are also publishing Open Data (OD) that provides authoritative reference data which has a level provenance and coverage not available from Semantic Web sources. One example that is used as an experimental data source within this project is company business data that are made available by Companies House (2013), the executive agency of the United

Kingdom (UK) Government responsible for the statutory registration and regulation of all companies and company officers within the jurisdiction of the UK. By releasing these datasets, Companies House is making available a complete listing of the business entities registered in the UK that includes unique identifiers through which data can be disambiguated and references to individual companies established. Therefore, such reference data may hold the promise of forming the semantic master data through which to enable the integration of other less exact Semantic Web resources. Although accurate and of good provenance, it is often made available as ‘raw data’ (Shadbolt *et al.*, 2012), that conform to formats such as tabular delimited text that provide little of the structure or the semantics that were explicit in the original data model (Bizer, Heath and Berners-Lee, 2009). Therefore integrating such data presents other challenges related recovering the semantics of the original data model.

The integration of resources at an internet scale also raises the challenge of processing ‘Big Data’ which is on a scale that is beyond the storage or processing capabilities of a single computer system. To address this problem, new distributed parallel architectures such as MapReduce (Dean and Ghemawat, 2008) are evolving that dispense with much of the imperative (procedural) programming paradigm and their associated resource locking mechanisms that are used to ensure consistency, in favour of a functional programming approach. A functional program is analogous to a mathematical function which maps one set of values to produce a new set (Hudak, 1989). Since this process does not result in a change of state (to the original data) the effects of concurrency are avoided as there is no change to coordinate. For example, storing and basing calculations upon an organisations formation date rather than its age will ensure that all parallel processes are dealing with the same immutable data, rather than for example, dealing with the constantly changing company age parameter.

This transition from mutable to monotonic stores and functional programming paradigm fits well with many of the NoSQL based methods of achieving data persistence; however, they differ from the standard RDBMS storage methods and therefore present yet more integration challenges for organisations and systems designers to manage and address.

In summary, the emergence of the Semantic Web and Big Data will present many opportunities to organisations that can successfully exploit the vast knowledge store that the Semantic Web and Open Data represent. However, there remains the problem of data integration which as it will be undertaken with a wide range of internally and externally sourced information must consider the semantics of and the model to which such data conforms. Therefore, finding new and innovative methods to effect semantic data integration is of importance to both the business process and information systems design community. Philosophical (foundational) ontologies can contribute to the solution of semantic integration by providing a lens through which to view reality and to model and structure and objects that stand-in for reality. Whether an integration project is dealing with tabular delimited files, RDBMS or expressive OWL based ontologies; all these source data sets have implicit and explicit semantics that need to be discovered and translated or mapped to a compatible form that can be integrated. A foundational ontology can provide the grounding through which these objectives can be achieved in a consistent manner. In addition to supporting integration processes, perdurantist foundational ontologies may also provide an effective means of providing the structure for large scale data warehouse systems that has time baked-in to the model – rather than being an external index.

2.3 The Problem of Semantic Data Integration

The work of many early pioneers of the field of computer science, such as Turing (1964), was founded on the premise that any symbol system employed within a computer system can be semantically interpreted by systematically assigning a meaning which stands-in for objects that describe a states of affairs (Harnad, 1990). In this sense, the meaning of the term semantics can be distinguished from the related linguistic sense of the word, where it refers to the study of meaning (Partridge and Stefanova, 2001).

Semantic data integration can be described as the process of interpreting instance data and the models to which such data conforms, aligning type, relations and individuals so they conform to a common semantic structure, then identifying the common relationships that hold between different datasets, including their types, (classes), relationships, axioms and the real (or possible) world objects that they stand-in for.

However, although there may be limited agreement on standardised symbol systems, there is no agreed way to interpret or model the reality, the state of affairs they may represent. Furthermore, people perceive reality in different ways; even when a set of models is developed by the same individual, they can make different arbitrary choices about the same reality at different times and in different contexts and all these models might be correct (Kent, 1978). A designer may also confuse categorisation, i.e. categorising the representation, the model, rather than what is being represented in reality (Partridge, 2002). There are also the different structures and restrictions introduced by heterogeneous modelling methods and languages employed by a systems designer. For example, within the realm of contemporary computer systems development, the prevalent modelling tools employed are visual Unified Modelling Language (Booch, Jacobson and Rumbaugh, 2000) and within the Semantic Web community, modelling tools and languages are based on predicate or first order logic typically expressed in the Resource Description Framework (RDF) (W3C, 2004) and Web Ontology Language (OWL) (W3C, 2009). OWL may conform to OWL Full, or a range of different fragments of Description Logics (DL) raising more integration issues. Within the database community Entity Relationship (ER) (Chen, 1976) and Relational Modelling (Codd, 1970) are prevalent. Thus the different models and languages will introduce heterogeneity of syntax, structure and semantics. Consequently, when integrating data that originates from different sources, the problem of semantic heterogeneity arises. Sheth and Larson (1990) state that this can be described as the disagreement on (and differences in) meaning, interpretation, or intended use of related data and thus semantic heterogeneity is a significant problem as it forms a barrier to semantic data integration. This problem has been well researched by the database community, for example Doan *et al.* (2004), describe this problem in terms of schema matching or mapping, data deduplication, record linkage, entity/object matching. Figure 2.1 and table 2.1 are based on the work of Visser et al., (1997) who developed a taxonomy of the semantic integration mismatches that can occur which is presented in Figure 2.1 and Table 2.1. These mismatches can occur when performing semantic data integration. Any one of these mismatches can cause errors within systems that integrate heterogonous data. Thus, to avoid such mismatches there is a need to investigate semantic integration solutions that can help mitigate such errors. In the case of this research we choose to investigate a solutions based on foundational ontology that provide a consistent and coherent means of interpreting and integrating the semantics of model types, relations and individual elements.

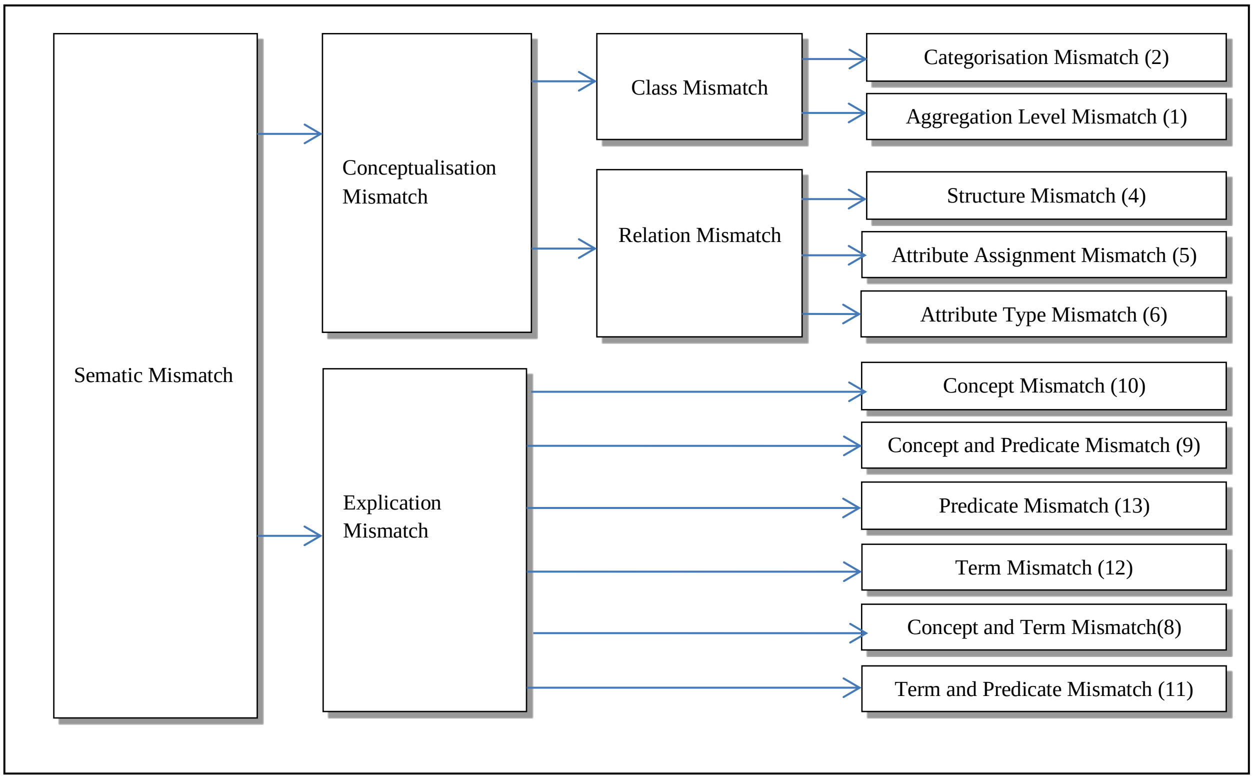

Sematic Mismatch
Conceptualisation Mismatch
Class Mismatch
Categorisation Mismatch (2)
Aggregation Level Mismatch (1)
Relation Mismatch
Structure Mismatch (4)
Attribute Assignment Mismatch (5)
Attribute Type Mismatch (6)
Explication Mismatch
Concept Mismatch (10)
Concept and Predicate Mismatch (9)
Predicate Mismatch (13)
Term Mismatch (12)
Concept and Term Mismatch(8)
Term and Predicate Mismatch (11)

Figure 2.1: Taxonomic hierarchy of semantic mismatch types adapted from Visser et al. (1997)

Item	Mismatch type	Description
1	Aggregation level	An aggregation level mismatch occurs if two ontologies both recognise the existence of a class, but define classes at different levels of abstraction.
2	Categorisation	Occurs when two ontologies contain the same class but each has different subclasses.
3	Relation	A relation mismatch concerns the relations distinguished in the ontology. Relation mismatches concern, for instance, the hierarchical relations between two classes or, the assignment of attributes to classes.
4	Structure	Occurs when two ontologies distinguish the same set of classes but differ in the way these classes are structured by means of relations.
5	Attribute assignment	An attribute assignment mismatch occurs when two ontologies differ in the way they assign an attribute (class) to other classes.
6	Attribute type	Occurs when two ontologies distinguish the same (attribute) class but differ in their assumed instantiations. For example different scalar values being used (miles vs kilometres).

Table 2.1: Semantic Mismatches – Adapted from Visser et al. (1997)

Item	Mismatch Type	Description
7	Explication	Explication mismatch occur when two ontologies have different definitions but their terms, their predicates and ontological concepts are identical.
8	Concept and term	Occurs when two ontologies use the same predicates but differ in both the concept they define and the term linked.
9	Concept and predicate	Occurs when the same term is used by two ontologies but differ in the concept the term defines. Such a mismatch implies that the term employed is a homonym.
10	Concept	Occurs when both ontologies have the same terms and predicates but differ in the concept they define. Such a mismatch implies that the term employed is a homonym.
11	Term and predicate mismatch	Occurs when two ontologies define the same concept but differ in the way they define it; both with respect to the term and the predicate. Such a mismatch implies that the two terms are synonyms.
12	Term mismatch	Occurs when two ontologies define the same concept using the same predicates but refer to it with different terms. This mismatch implies that the two terms are synonyms.
13	Predicate mismatch	Occurs when two ontologies define the same concept and use the same term to refer to the concept but use a different predicates.

Table 2.1: Continued. Semantic Mismatches – Adapted from (Visser et al., 1997)

Other fundamental problems exist, for example integrating models that are based on different levels of expressivity or abstraction, such as integrating elements of a lightweight ontology from Schema.org with an OWL Full ontology. Figure 2.2 presents a number of ontologies and their place on the ontology spectrum.

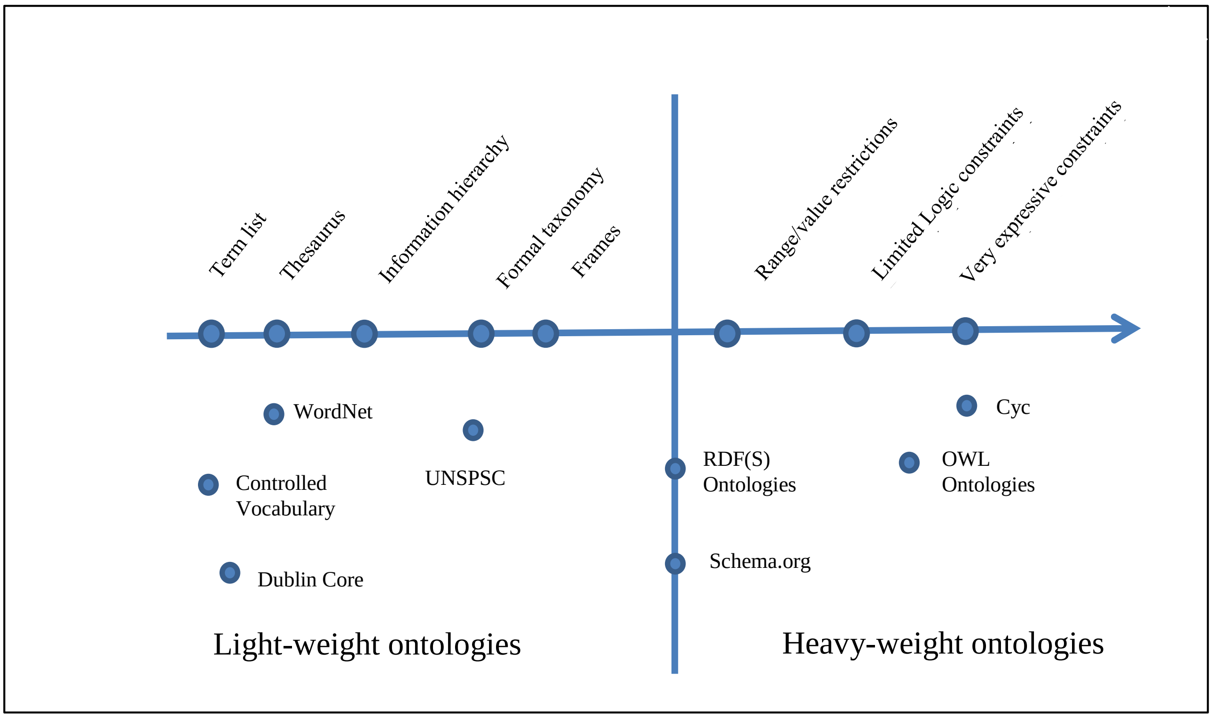


Figure 2.2: Ontology Spectrum (McGuinness, 2005).

Within IS and ontology design, models tend to be partitioned into structural and behavioural categories. Structural models provide a representation of a static domain such as Unified Modelling Language (UML) (Rumbaugh, Jacobson and Booch, 2004) class diagrams, whereas behavioural models describe the dynamics of a domain, examples include the Business Process Modelling Notation (BPMN) (Object Management Group (OMG), 2011) diagrams and UML Interaction, State-chart and Activity diagrams. This partition into static and behavioural models also exists within a number of foundational ontologies. For example the Basic Formal Ontology (BFO) contains two modules; SNAP a static structural snapshot and SPAN which encompasses the means of describing behaviour over time (Grenon and Smith, 2004). This partitioning adds further complexity to the integration problem - for example, within BFO trans-ontological relationships are introduced to connect SNAP and SPAN.

During the course of the development of a new system or service, it is common practice within the information systems design community to develop a number of models, for example conceptual, logical and physical data models (Codd, 1971). These models form strata with the Conceptual Data Model (CDM) forming the top stratum of the hierarchy. The CDM model is understandable by technical and non-technical stakeholders involved in the creation of the system designs but provides little technical detail. The middle layer is formed the Logical Data Model, this supplements the CDM with information related to the overarching technical solution. Finally the lowest layer is formed by the Physical Data Model (PDM) which is implementation, and in some cases software vendor dependant. The layering of the design models into a number of strata has a number of adverse effects. Firstly as the model is converted from CDM to LDM and then from LDM to PDM the original semantic structures may be distorted and or lost completely as the emphasis of the modelling activity moves from representing the real-world domain to representing data structures. Secondly, the semantic data contained within the higher level CDM, LDM models are typically not documented within the instantiated physical system; hence, such information is not accessible to applications that are attempting to integrate such information (Roddick, 1995). This sparse level of schema information may be further reduced when it is exported for use as tabular delimited 'Raw Data' (Bizer, Heath and Berners-Lee, 2009). It is worthy of note that some architecture model frameworks such as Zachman feature as many as five model layers from contextual through to detailed data definitions (Zachman International, 2015).

Domain Specific Models (DSM) models have also been developed to support the design and development of information systems. Such models have proved a valuable aid to information system development and are widely used, for example the Resource Event Agent (REA) (McCarthy, 1982) is a DSM that has a set of patterns specifically developed to model the accounting domain (Geerts and McCarthy, 2002). Many other Domain-Specific Models

(DSM) are available which can assist IS designers by providing a highly abstracted model that can be employed as the underpinning for specific software solutions and in some cases can also be employed to generate software program code. Although such models have many advantages from the pragmatic system development viewpoint as complex systems can be constructed from a range of such domain specific models (Warmer and Kleppe, 2006). This approach can also lead to a diversity of models and their associated explicit and explicit semantics. Large complex systems such as Enterprise Resource Planning (ERP) systems may employ a number of such domain specific models together with the applications that map and integrate data between them. This may present a problem to integration with external systems as the detail of how these internal system models are integrated and what transformational rules are applied may not be apparent from data made available to external systems integration.

In summary, the multitude of models that conform to differing structures, syntax and logical theories form a barrier to semantic integration. Therefore, finding some means of providing a unifying view may prove advantageous. The grounding provided by foundational ontologies can help identity what exactly the source data represents and consequently forms a major part of a solution to the problem of semantic integration. This grounding can also help eliminate/reduce the semantic mismatch errors described earlier (Visser *et al.* 1997). A foundational ontology is limited to concepts that are meta, generic, abstract and philosophical, and are therefore general enough to model, at the highest level, a broad range of domain areas (Pease and Niles, 2002). However, in addition to providing a solution, there are a diversity of foundational ontologies which are based on a multiplicity of metaphysical theories; therefore again the problems related to heterogeneity arise. The following section will present literature overview of the subject of ontology and some of the major metaphysical choices which can lead to such diversity.

2.4 **Ontology**

The aim of this section is to present literature related to foundational ontology and to provide the reader with an outline understand of the relationship between the philosophical paradigms described in the remainder of the thesis.

The term ontology originated in the field of philosophy with a history dating back to the Ancient Greeks. Hofweber (2014) states that the ontology is a discipline consisting of the study of:

a) belief in what exists in reality - ontological commitment,

b) what exists in reality,

c) the most general features of reality ,

d) how things in reality are related,

e) meta-ontology (Hofweber, 2014).

Lowe (1998) stated that an ontology is “the set of things whose existence is acknowledged by a particular theory or system of thought” (Lowe, 1998, p.634). Citing Lowe (1998), Partridge *et al.* (2013) assert that considering ontology as a theory, has led to the ‘objectification’ sense of ontology which is the sense that is relevant to information systems; as information within a computer system can be considered to be a theory that represents a set of things in reality.

Gruber (1993) defined the term within the context of computer science to mean a specification of a conceptualisation:

> “An ontology is a description (like a formal specification of a program) of the concepts and relationships that can formally exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set of concept definitions, but more general. And it is a different sense of the word than its use in philosophy’ (Gruber, 1993, p.199).

Inherent to Gruber’s (1993) definition that a system (ontology) designer can only know reality through his or hers own senses and experience, is an adherence to an idealist philosophical stance. There are people within the field of ontological study that are critical of this stance, Smith (2004), for example, concludes that Gruber’s (1993) view has become predominant within IS due to the practices inherent in the field of Artificial Intelligence (AI) where it is assumed that an ontology is a form of representation of the concepts conceived by the human mind; rather than reality. As language is used to frame thought and OWL, the predominant Semantic Web Ontology language was derived from earlier AI languages. This view has also reinforced by ontology development tools such as Protégé (Stanford Center for Biomedical Informatics Research, 2015). The OWL language is modelled on different assumptions and semantics that of database and Object Oriented (OO) software. The following table 2.2 provides an overview of equivalent features and major differences between the paradigms.

Feature	Relational Database	Ontology
Uniqueness	A key unique within a table	Internationalised Resource Identifier (IRI) unique within the global area network (The Semantic Web)
World Assumption	Closed world: Missing information is assumed to be false	Open World: Assumption (OWA) and reasoning i.e. negation means provably false. Expressions not present result in a not known state rather than false
Naming Assumption	Unique Names	No Unique Name Assumption (UNA) An assertion must be made to indicate that entities are different.

Table 2.2: Comparison of the Assumptions RDBMS and Ontologies (Horrocks, 2008)

Within Object Oriented software design (UML), classes are regarded as templates for instances with sub-classing being a mechanism through which to inherit the behaviour of the superclass, whereas within an ontology the semantics of the super-subclass relationship is set thematic.

2.4.1 Ontology Classification

Ontologies are typically classified by their intended scope. Ontologies that are subject-independent are termed upper-level or foundational ontologies, whereas, domain ontologies that are limited to a particular area of interest, are often termed the universe of discourse (Geerts and McCarthy, 2000).

2.4.2 Foundational Ontologies

A foundational ontology is limited to concepts that are meta, generic, abstract and philosophical, and are therefore general enough to model, at the highest level, a broad range of domain areas (Pease and Niles, 2002). Philosophical ontologies can therefore be employed to provide the ‘grounding’ in a fundamental view of reality and thus provide a common reference through which to model and integrate relevant semantic resources. Therefore, philosophical ontologies can be used to provide a consistent foundational theory and the associated entities and constructs that can be employed to support semantic interoperability.

In terms of a concrete implementation of a software system, foundational ontologies can be used to establish the fundamental meta-ontology objects and relations used to construct more specific domain ontologies. If a common foundational theory is extended and specialised to model a number of domain ontologies, the objects common to each of these domains will have a common grounding to enable semantic integration. Consequently, foundational ontologies are important as they can provide a philosophical standpoint for the view of reality which underpins all the domain models to be integrated and therefore offer semantic grounding.

They can also provide the structures through which to store and organise semantically integrated data to a database. However, foundational ontology is itself founded on a metaphysical theory. These meta-ontology theories provide a criterion for the ontological commitments, which are principally the things believed to exist within the context of a particular theory (Bricker, 2014). Willard van Orman Quine (1948) is often quoted on the subject of the ontological diversity that exists within metaphysics.

> “A curious thing about the ontological problem is its simplicity. It can be put in three Anglo Saxon monosyllables: "What is there?‟ It can be answered, moreover, in a word "Everything‟ and everyone will accept this answer as true. However, this is merely to say that there is what there is. There remains room for disagreement over cases; and so the issue has stayed alive down the centuries” (Quine, 1948, p1).

Although most philosophers agree on the existence of individuals (or particulars), types (or classes) and the relations that connect them, there is no common theory that describes reality - philosophers may even disagree about the nature of their disagreement (Dorr, 2005). This lack of consensus at the metaphysical level is an obstacle to semantic integration (Campbell and

Shapiro, 1995). As a result of the lack of consensus there are several foundational ontologies, each of which adopt a different philosophical view, such as those that are grounded in the world of mental concepts and language (idealist) and those that adopt a realist stance; a mind independent view of reality, i.e. the planet Earth would exist as an object with or without a human mind to observe it. Although, no universal foundation ontology exists, selecting a particular foundation and thus starting from the grounded view provides advantages in terms of the quality and interoperability of domain ontologies (Keet, 2011).

In summary, there are a number of foundational ontologies each of which adopts one of the diverse foundational metaphysical theories. Consequently, in adopting a foundational ontology and a related philosophical view, a commitment needs to be made as to what one believes to exist (Quine, 1952). It is therefore useful to understand what commitments are being made when a particular foundational ontology is selected for purposes of semantic integration and to appreciate the impact of integrating data that may conform to fundamentally different metaphysical theories. An overview of the subject of ontological commitment is provided in the following section.

2.4.3 Foundational Ontologies and Ontological Commitment

This section provides an overview of a number of the major concepts related to the, meta-ontological philosophical ontological commitments and how they relate to the models developed for use within information systems.

Realism vs Idealism

There are two major schools of thought, idealism and realism, which have for millennia been the subject of the debate on how the human mind can know reality. Idealist philosophers view reality as fundamentally a construct of the mind which is dependent on human perception. As asserted by Partridge *et al.* (2013) this leads to the erroneous conclusion that reality is nothing more than a construction built by each individual person's concept system. Thus, as each information system is designed by an individual who reflects their own view of reality, the result is a multiplicity of conceptual models and the instantiation of incompatible systems. Quine (1948), who also adheres to the realist philosophical stance, stated that what exists can be determined by observing which entities are endorsed by scientific theories. This realist stance is also supported by Sider (2001) and Van Inwagen (1998) who all assert that what exists can be clearly recognised as the 'real world' outside the mind. This realist stance is often reflected in the context of IS system design, in the phrase 'real world models' or 'real-world semantics'.

The following figure 2.3 depicts the often cited semiotic triangle of Ogden *et al* (1946) which has been adapted to depict the relationships between conceptual (idealist) and referent (realist) models.

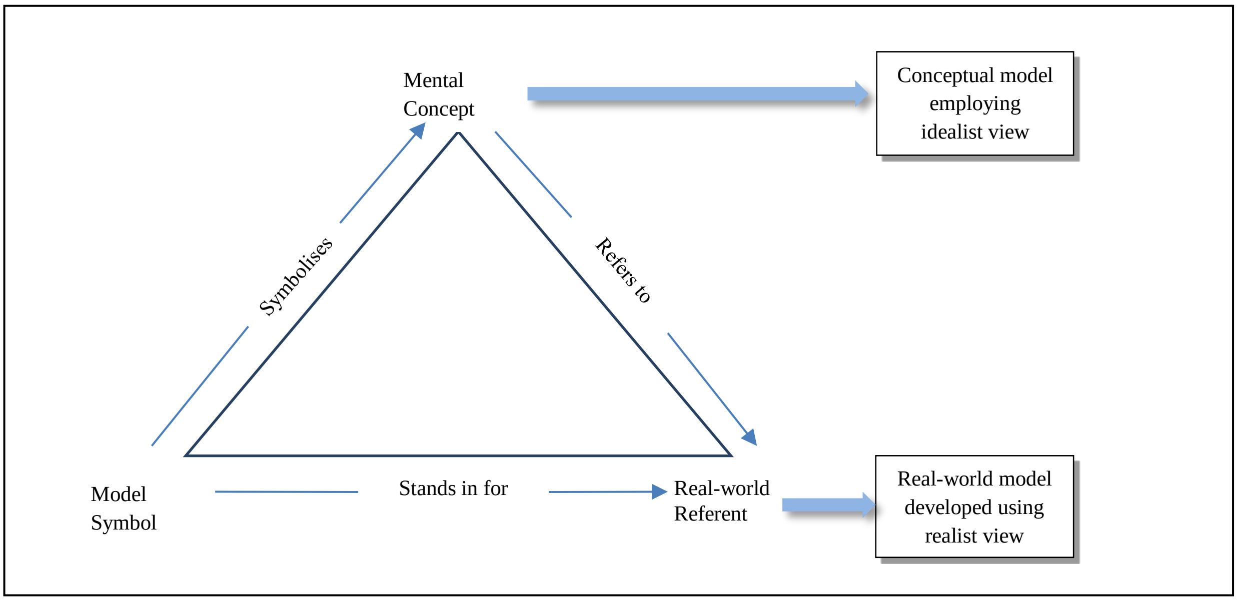


Figure 2.3: Model Relationship adapted from the Semiotic Triangle (Ogden et al., 1946)

This Real-world approach to semantics is described by Partridge, *et al*., (2013) as direct-domain-semantics a position that is concept-free and therefore not affected by the interpretation and consistency issues that surround the use of 'concepts'.

Descriptivism vs Revisionism

As stated by Strawson (1964b) if an ontology is developed that is based on existing documents and systems; then such an ontology that has been inferred from those artefacts can be termed a descriptive ontology. Consequently, a descriptive ontology captures common sense and social norms from natural language usage and human cognition. Citing Strawson (1964b) Partridge *et al.* (2013) state that there is a choice in philosophical ontology between a commitment to 'descriptive' or 'revisionary' metaphysics. Descriptive metaphysics seeks to describe reality via the conceptual schemes of human thinking and will therefore reflect the accepted picture of the world. However, Partridge *et al.* (2013) criticise this approach as it relies on current assumptions relating to the general nature of reality. Citing Lewis (1986) , Partridge *et al.* (2013) assert that committing to the revisionary approach will result in an effort to improve existent philosophical theories. As a consequence of this design aim, revisionary ontologies attempt to capture the intrinsic nature of the world and will in many cases impose restrictions on the type of entities that are admitted, i.e. only admitting spatio-temporal entities (Oberle *et al.,* 2007).

Endurantism vs Perdurantism and the Criteria of Identity

Endurantists hold the view that continuants (objects that extend in dimensional space) endure, i.e. they pass through time as a whole object and therefore are completely present at each point in time they occupy. Consequently, as they are wholly present at each point in time, they cannot have temporal parts (Stell and West, 2004). Three dimensional objects are always viewed relative to the present and therefore to make an assertion relating to a continuant at different times requires that such an assertion also contains a form of temporal indexing (Stell and West, 2004). Due to their adherence to the belief in continuant objects, endurantists are described as three dimensionalists. Opposing the endurantist view, perdurantists (four dimensionalists) assert that objects extend over four dimensions, three of space and one of time (Sider, 2003). Perdurantists assert that all non-present past and future objects exist, but are remote from the present time. As objects are extended in time by accumulating different temporal parts, so that, at any time they are present, they are only partially present, in the sense that some of their temporal parts (e.g., their previous and future states) may be not present (Sider, 1996). As individuals are extended in time as well as space they can possess both temporal and spatial parts. As a spatio-temporal extent is not wholly present at a point in time, an object at a point in time is a temporal part of the whole four dimensional extent, thus change can be expressed through the whole-part relationships that exist between the spatiotemporal parts that make up the complete spatiotemporal whole (Sider, 1996). For example, Figure 2.4 is a space-time map which depicts the company, ACME Company Limited as a 4D spatio-temporal extent that extends through space-time. ACME’s extension has temporal parts named ‘Business Activity 1’ representing the company’s paper manufacturing state and ‘Business Activity 2’ representing the mobile phone manufacturing state. Both these temporal parts (states) are extensions that are physically part of ACME’s overall spatio-temporal extension.

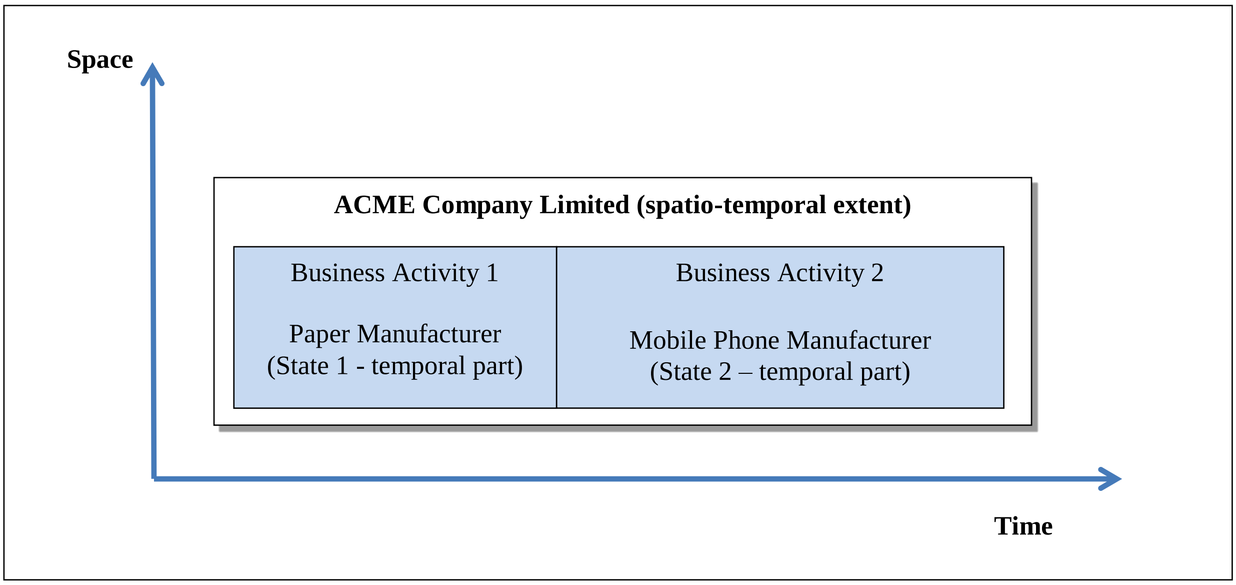

Space
ACME Company Limited (spatio-temporal extent)
Business Activity 1
Paper Manufacturer
(State 1 - temporal part)
Business Activity 2
Mobile Phone Manufacturer
(State 2 – temporal part)
Time

Figure 2.4: Example space-time map (de Cesare, Foy and Partridge, 2013)

In contrast to figure 2.4, figure 2.5 depicts temporal parts that overlap, illustrating the extensionalist criteria of identity; that if two individuals have the same spatio-temporal extent (i.e. they are in the same space at the same time) then they meet the extensionalist criteria of identity and are therefore the same thing (Partridge, 2005).

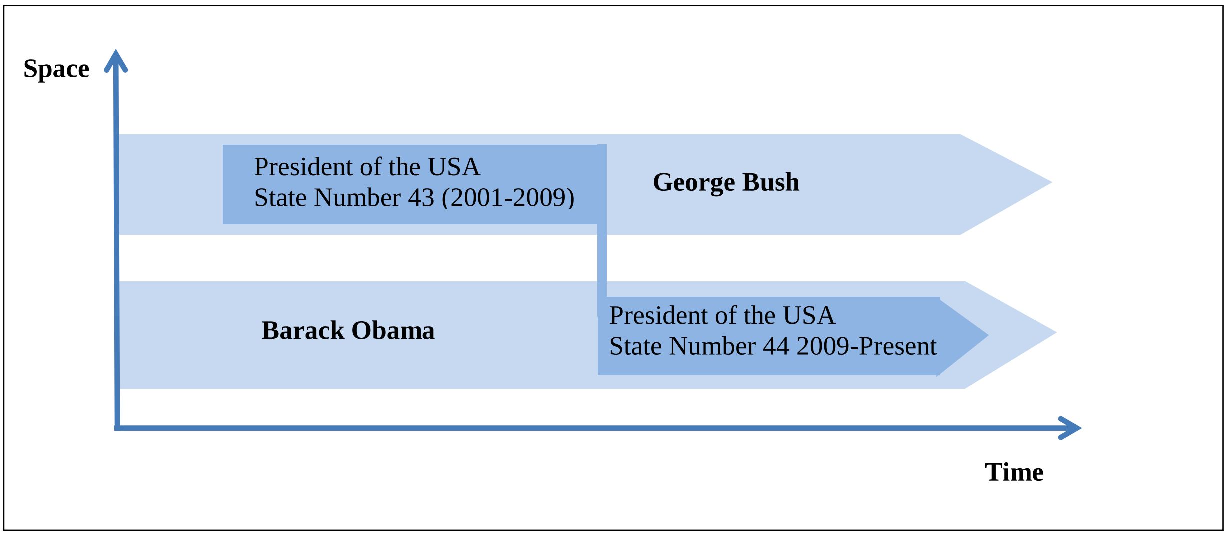


Figure 2.5: Time Space Map President of the USA (Partridge, Mitchell and de Cesare, 2013)

Perdurantists assert that objects persist by having different temporal parts at different times. Figure 2.5 depicts a space-time map of the four-dimensional President of the USA Object. In this map, George Bush and Barak Obama have simple straight time-lines. The President of the USA is a socially constructed object that is a composite of temporal parts of George Bush and Barak Obama. A temporal part (time-slice) of George Bush's full tempo-spatial extent (time-line) is his Presidency of the USA. Similarly, a temporal part of Barak Obama's is his Presidency. The fusion of these two (forty third and forty forth) presidencies together with previous and future presidency temporal parts forms the President of the USA object (Partridge, Mitchell and de Cesare, 2013).

Multiplicative vs. Reductionist

As co-located entities are assumed to have incompatible essential properties, a multiplicative ontology allows two or more entities to be co-located within the same spatio-temporal coordinates (Oberle *et al.*, 2007). Reductionist ontologies assert that a spatio-temporal location can contain only one object (Oberle *et al.*, 2007). Therefore as depicted in Figure 2.5, the President of the USA and Barack Obama are during the period they overlap - the same object.

Time and Presentism vs Eternalism

On the subject of time, McTaggart (1908) introduced two distinct approaches via which events in time can be ordered; he termed these the 'A' and 'B' series. The 'A' series theorises that each event in time can be ordered by examining its properties of future, present or past. The 'B' series hypothesises that all events in time are relative, for example event 'X' is before event 'Y' and that as these positions are relative they are fixed forever. McTaggart (1908) concludes that the 'A' series must be contradictory as each event possesses the incompatible properties of past, present and future. Furthermore, as the 'B' series cannot exist without the 'A' series, McTaggart concludes that time itself, including both the 'A' and 'B' series, are unreal and therefore that temporal order is illusory.

Having considered that time may be illusory, this brief outline moves on to the debate relating to the presentism versus eternalism. Presentists commit to the theory that only objects that are temporally present exist, consequently they commit to the belief that all things that have existed or will exist are unreal (Sider, 1996). In contrast, eternalism is a version of the non-presentist view which commits to the ontological theory that in addition to the objects that presently exist, past and future objects also exist (Sider, 1996). And just as it is not possible to perceive an existing object at an extreme physical distance, past and future objects are not in the present space-time location. Consequently, from the eternalist ontological view, an object's temporal proximity has no bearing on the existence of such an object (Markosian, 2014).

Actualism vs Possibilism

Actualists assert that only what is real exists, while possibilists also admit possible worlds into their view of reality. Partridge *et al.* (2013) assert that possible world semantics provides the facility to model objects that exist in worlds other than that of the 'real world'. According to Look (2013), the theories relating to possible worlds have been explored by several leading philosophers and mathematicians, including Leibnitz who first explored the subject. Theories related to possible worlds such as Modal Logic (Kripke, 1959) and Modal Realism (Lewis, 1986) have been developed in more recent years. Supporting possible world semantics within an ontology provides the facility to include objects from simulations such as from 'what if' scenarios or fiction. It also provides that facility to clearly disambiguate objects from a possible world from the real world such as the 'Wonka Bar', a fictional chocolate bar manufactured by the fictional Willy Wonka Company, both of these object belong to the possible world described by author Roald Dahl (2007) and the real 'Wonka Bar' chocolate bar, which is manufactured in a real factory by the real Willy Wonka Company, a brand owned by Nestlé USA Inc.

Sets

On the subject of sets, Smith and Ceusters (2010) state that employing set thematic theory within a foundational ontology is erroneous due to the premise that sets are timeless abstract mathematical objects and that employing sets to stand in for real world entities presents a problem of how to relate the abstract mathematical set with a concrete collection of objects. Furthermore, that by employing sets in this manner, the mathematical set-forming operator has the effect of isolating the referents from time and space and therefore by implication that, nothing concrete can exist. Partridge *et al.* (2013) provide a contrary argument that sets are not timeless abstract mathematical objects, that they are in fact real, and provide the following thought experiment to support this assertion. Consider a number of objects such as: a paper-weight, a lamp and a laptop computer that are formed into a set by having in common the same physical location, i.e. one particular desk. It is possible to conclude that located on this one particular desk are both the objects that constitute the set and the set itself (Partridge, Mitchell and de Cesare, 2013).

Abstract Objects

Within contemporary philosophy there is an ongoing debate relating to the subjects of realism, materialism and physicalism and there is no agreement on a standard account of how to distinguish between abstract and concrete objects (Rosen, 2014). Philosophers who commit to abstract objects are often termed platonists whilst philosophers who deny their existence are termed nominalists (Rosen, 2014). As abstract objects are defined as those objects that have no spatial extent and have no causal powers, Partridge, *et al*. (2013) assert that as nothing can be known about them or their existence, ontological commitment to abstract objects should be avoided.

Higher Order

Higher Order Logics (HOL) and the ontologies based on these logics provide the facility for types (or sets) to also be members of types. An ontology that does not support this structural relationship is limited to the first order. Limiting the expressivity of an ontology to the decidable fragments of First Order Logic (FOL) has practical advantages such as being tractable to machine inference (over variables). However, these limits mean that such ontologies cannot effectively model a number the fundamental structures found in the real world, such as classification systems. Therefore, an ontology committing to support higher order elements will provide the facilities to more accurately model the world; however this will be at the cost of losing machine reasoning (inference engine) support.

Well-Founded vs Non-Well –Founded

In a similar vein to the previous paragraph relating to disallowing higher order objects, an ontology may adopt a well-founded set structure which restricts a type or class from being a member of itself. This avoids Russell’s Paradox (Russell, 1902), but restrict the ontology from asserting the foundational class object – the class of all classes (that is itself a class).

Ontological Commitment Summary

Due to the lack of consensus at the metaphysical level, several foundational ontologies have been developed each of which adopt differing philosophical views. Each of these philosophical stances outlined in the previous paragraphs represent ‘the tip of the iceberg’ regarding the mass of philosophical knowledge that has been produced over the millennia, however, such choices will result in foundational ontologies which differ significantly. Table 2.3 provides a brief overview of a number of foundational ontologies and their associated ontological commitments Some of the listed ontologies are composite, for example the Unified Foundational Ontology

(UFO) (Guizzardi, de Almeida Falbo and Guizzardi, 2008) aims to unify (or map) other foundational ontologies. Although this presents a problem when integrating ontologies based on different metaphysical theories, as long as such the theories are applied consistently, then there exists that possibility of translating from one foundation to another.

Ontology	Description
Cyc	Cyc, a proprietary ontology that consists of a foundation ontology and a number of domain ontologies termed micro-theories. Each micro-theory is free of contradictions; however the complete ontology is not (Cycorp Inc., 2015.).
BORO	The Business Objects Reference Ontology (BORO) provides a foundation, upper ontology together with a method for constructing domain ontologies. It adopts a metaphysical stance based on an extensionalist criteria of identity and consequently is a perdurantist (four-dimensional) ontology (Partridge, 2005).
J. Sowa's ontology	A general ontology based on the philosophical comprehensions of Sowa (1999).
BFO	The Basic Formal Ontology (BFO) consists of two related but separate ontologies SNAP relating to continuant entities such as three-dimensional enduring objects, and SPAN occurrent entities. Thus BFO incorporates both three-dimensionalist and four-dimensionalist perspectives (Grenon and Smith, 2004).
DOLCE	The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) has a cognitive bias and is therefore descriptive. As DOLCE restricts the use of universals to organise and characterise particulars, the ontology is limited to representing first order models (Gangemi *et al.*, 2002).
GFO	The General Formal Ontology (GFO), is a realist ontology which in the same way as BFO differentiates between endurants (objects) and perdurants (processes) (Herre, 2010).
UFO	The Unified Foundational Ontology (UFO), incorporates GFO, DOLCE and OntoClean (Gangemi *et al.*, 2002) in a single foundational ontology that is partitioned into three modules: UFO-A defines the core of UFO excluding terms related to perdurants and intentional social things, UFO-B is limited to terms related to perdurants and UFO-C contains intentional social and linguistic things (Guizzardi, de Almeida Falbo and Guizzardi, 2008).
UMBEL	Upper Mapping and Binding Exchange Layer (UMBEL) is a curated subset (the open version) of the Cyc ontology developed with the aim of providing mapping between formal and less formal ontologies such as DBpedia (Jain *et al.*, 2010).

Table 2.3: A Number of Foundational Ontologies

In summary, there are a number of the metaphysical theories that underpin foundational ontologies that are significantly different and therefore, to ensure that a coherent ontological architecture is developed, it is important that consistent philosophical commitments are made. A framework that provides a structure for these choices is provided by a foundational ontology and the ontological paradigm adopted. As asserted by Partridge *et al.* (2013) due to the increasing use of semantics within Information Systems there is an emerging need for understanding ontology in the philosophical sense.

2.4.4 The Ontological Paradigm

Science is based on a predominant paradigm which consists of an integrated set of theoretical and methodological beliefs (Kuhn, 1964), for example in the case of contemporary physics there are two such paradigms, relativity and quantum mechanical theories. For any new paradigm to be accepted, the beliefs that predicate such a new paradigm must appear superior to the existing or previous theories.

Kuhn (1964) coined the term ‘paradigm shift’ to describe this transition from one prevalent theory and set of beliefs to a new one. He also provides an example of such a shift by describing the change that occurred when the geocentric model of the solar system, based on Ptolemy's school, was replaced by the heliocentric model proposed by Copernicus. Kuhn (1964) also asserts that the actual paradigm shift occurred some time after the Copernicus model was first proposed, not taking place fully until the publication of Galileo Galilei manuscript concerning motion, Kepler theory on planetary orbits and the development of Newton’s theory of universal gravity. Therefore, the paradigm shift from the Aristotelian/Ptolemaic model was not instantaneous; rather it took place over the course of several decades and was opposed by many individuals and organisations that were committed to the geocentric model.

As asserted by Partridge (2005) the theory of relativity and the unified view of space-time developed by Einstein (1920) also led to such a paradigm shift within the scientific community. This new way of viewing reality as a space-time fabric was applied by Quine (1948) to resolve a number of philosophical problems related to the identity of physical bodies over time. This work resulted in a simplification of the separate patterns for space and time into a general pattern for space-time. The continuity of the identity of physical objects over time is based on the concept of four-dimensional extensions. This four dimensional approach was termed perdurantism by Sider (2001), who advocated the theory of existence based on an object having extension in this universe, through the three dimensions of space and the fourth dimension of time; a spatio-temporal extension. However, as with previous shifts, the accepted ways of seeing the world have not changed contemporary society or most engineering practice. Thus software design and modelling are still largely based on the 'Entity' paradigm, a simplified version of the Aristotelian substance paradigm, which for over two millennia has provided the predominant paradigm which views the physical world as being constructed from three dimensional entities that endure through time (Partridge, 2005). The following table provides an overview of some of the basic differences between the perdurantist and endurantist view.

Endurantism (3D)	**Perdurantism (4D)**
Objects have only the three spatial dimensions.	With the exception of events, Objects have three spatial and one temporal dimension and are viewed as spatio-temporal extents. Events deemed to be three dimensional object (as they have no temporal extent) in a 4D universe.
Objects are deemed to endure that are wholly present at any point in time during the course of their lifetime.	At any given time a 4D object is only partially present.
Objects are viewed from the present. The default is that statements are true now.	Objects from the past, present, and future all exist.
Objects do not have temporal parts.	Objects extend in time as well as space and therefore have both temporal parts and spatial parts.
Different objects may coincide at a point in time, i.e., occupy the same 3D extension (non-extensionalism).	When two objects occupy the same location in time and space they are the same thing (the extensionalist criterion of identity).
Time and space are treated separately. All statements need to be time indexed.	Time and space are one fabric.
Understand change in terms of things.	Understand things in terms of change.

Table 2.4: Metaphysical Differences-Endurantist vs Perdurantist Ontologies (Al-Debei et al., 2012)

The Object (4D) Paradigm

The Object Paradigm (Partridge, 2005) adopts perdurantism as described by Sider (2001) and therefore identifies an object (or thing) through its spatio-temporal extension. The Object Paradigm is the basis for the Business Object Reference Ontology (BORO) which will be discussed in the following sections. However, to avoid confusion related to OO programming – often also described in terms of the Object Paradigm - the term 4D Paradigm is used throughout the remainder of the this text.

Identity Over Time

The 4D Paradigm provides a solution to the problem of identity over time, which is a subject that has been a focus of philosophical study and debate for millennia. To illustrate the nature of this problem, it is useful to examine Theseus's paradox, a thought experiment undertaken by Plutarch that raises the question as to whether an object, in this case Theseus's ship, that over the course of time had all its original timbers replaced, remains the same object or not (Cohen, 2004). The 4D Paradigm can address such a problem by maintaining a reference to the original entity through temporal mereology and temporal parts. Therefore, a physical object can maintain a coherent identify whilst many or all of its (spatio-temporal) parts are substituted over the course of time. Consequently, adopting 4D Paradigm has many advantages for integration as it provides a clear and unambiguous means of establishing identity over time. For example a company ‘ACME’, may have many states based on changing name, address, company officers and business activity. By employing the 4D Paradigm, these become a series of states (temporal parts) of the company (whole spatio temporal extent). This allows the integration of both current and historic information. For example, it would allow information pertaining to previous business activities or pervious company officers to be integrated in a coherent manner.

2.5 The Business Objects Reference Ontology (BORO)

The aim of this section is to present an overview of the Business Objects Reference Ontology (BORO), a foundational ontology which is adopted as the foundational ontology that will be employed throughout this design science investigation.

BORO is a perdurantist (four-dimensional) ontology that has a metaphysical grounding in the 4D (Object) Paradigm. Developed by Partridge (2005), BORO has been employed to as the foundation for a number of major ontologies that include the International Defence Enterprise Architecture Specification for exchange Group (IDEAS Group, 2011) and the U.S. Department of Defense Architecture Framework (DoDAF Architectures Framework Working Group, 2009). BORO was also employed as the foundation for ISO 15926 which has been used extensively within the field of oil and gas (West, Partridge and Lycett, 2006). The BORO foundational ontology defines the basic ontic categories and a number of patterns such as *super-subclass, types-instance, named-by* etc. BORO is higher order extensional ontology which employs physical existence as the criterion for identity.

In terms of being applicable to semantic integration the ontology has a number of distinct advantages that are detailed in the following Section 2.5.2.

The explicit metaphysical choices which form the BORO ontological architecture are presented in Table 2.5.

Feature	Description (Ontological Commitments)
Realist	It takes a mind-independent real world view.
Revisionary	It changes the way we look at the world.
Completeness categories	Based upon extensional criteria of identity.
Perdurantist	It adopts a four dimensional view – objects are spatio-temporal extents extended in time and space.
Eternalist (ontologically)	Past, present and future objects all exist.
Presentist (epistemically)	Any implementation based on the ontology exists in the present and we cannot know the future. The histories of spatio-temporal extents' past events and states are accessible to the system.
Possible worlds	Supports the theory of possible worlds.
Extensionalist	A clear way of establishing identity and identity over time.

Table 2.5: BORO Ontological Architecture (Partridge, Mitchell and de Cesare, 2013)

2.5.1 Rationale for Selecting The BORO Foundational Ontology

BORO has a precise and simple method for establishing the criteria for identity based on the analysis of individual objects, if one of more objects occupy the same space at the same time, then they are the same object (as depicted on figure 2.5, the president of the USA, Obama example). This extensionalist method of establishing identity can also be applied to Type objects; if they have the same member objects then they are the same. However, this method requires recursion until individual element objects spatiotemporal extents can be compared. Furthermore, as the ontology stores the history of an object in the form of its previous states, the full life cycle of the object can be modelled. Therefore structural and dynamic elements can be combined into a single model.

BORO is a pure 4D ontology – other Foundational Ontologies such as the BFO (Grenon and Smith, 2004) employ separate foundations for static continuant material entities and occurrent (processes and event) objects. Having a single object that represents a spatio temporal extent simplifies the relationship between states, events and objects to that of a mereology.

BORO therefore provides:

a) A simple set of foundation meta types (established through the partition of objects type): Elements, Types and tuples (relationships) and a straight forward means of categorising data being integrated.

b) A clear means of establishing and maintaining the identity of individuals, types (classes) and relationship objects that is based on explicit metaphysical choices (as presented in table 2.5).

c) The separation of names and the entities they name into separate structures. Thus removing the name referent confusion that can occur.

d) Perdurantist objects have both temporal parts (states) and physical parts (elements) through which the ontology can maintain identity over time, and it can identify elements with a degree of precision that is not possible using names alone.

e) The identity of an object is based on its place in the ontological structure, not its name.

f) The ability to model higher order structures (types of types etc.) providing the maximum flexibility when modelling source data and how they can be combined structured and integrated.

g) A notation based on BORO UML which can be developed using industry standard design tools.

2.5.2 The BORO Foundation

The BORO ontology consists of the foundation (ontic categories) and a number of patterns. Table 2.6 details the ontic categories of the ontology and the discriminant that forms the distinction between these categories that is their criterion of identity.

Object	Description
Objects	The root object of BORO is ‘Objects’. This meta-type is partitioned to form the ontic categories Elements, Types and tuples.
Elements	The Elements Object represents a Type that has all Elements as instances. This is an ontic category, with its criterion of identity. Elements are those things which have spatio-temporal extent and as BORO is a 4D eternalist ontology it encompasses physical things that exist in the past, present and future. Element identity is established by 4D extent; if two Elements occupy the same space and time, they are the same. An element's spatial extent may tend towards zero (e.g. a point in space), as may their temporal extent. An element with no temporal extent is an event). Elements are differentiated from Types as they cannot have member instances.
Types	The Types object has all Type objects as instances. The Types object is also a Type and is therefore reflexive (a member of itself) and non-well founded. Types, therefore, is an object that contains all Type objects. The member Type objects are differentiated from Elements by that fact that they can have member instances. A Type objects criteria for identify is based on the spatio-temporal extent of the member objects. BORO semantics does not restrict Types having members that are also Types and therefore the BORO is a higher order ontology.
Tuples	The tuples object is an instance of meta-types and has all tuples as instances. This is its criterion of identity. Tuples are used to establish relationships between objects. The identity of a tuple object is based on the identity of the related objects that are present in each of the tuple’s places. Unlike many modelling and implementation systems, BORO is not limited to binary relations; any number of places can be defined. Furthermore, tuples are a first class object and can also occupy a tuple place. Tuple Types and higher order objects such as Types of Types of tuple can also be formed. This provides the ability to model relationships in a natural way rather than distorting the relationship to fit the model.

Table 2.6: BORO Ontic Categories (Partridge, 2005)

From these foundational objects the rest of the BORO is constructed. BORO has a number of common constructs ‘patterns’ that are used extensively within the foundation such as an individual object being a member of a specific Type, a Type being a subtype of another Type etc. As is described in the next section, these patterns can be specialised to develop domain ontologies, for example to define the Type Companies (that has all companies as member instances), it is possible to assert that the Companies Type is itself a member of the foundation Types (Type that has all Types as members).

2.5.3 Ontology Design Patterns

BORO provides a range of ontological patterns that can be specialised and adapted to form domain ontologies. These ontological patterns, will, as is discussed in the following chapters related to Design Science, be employed to instantiate the prototype graph based warehouse. These patterns encompass aspects of 4D ontologies (along with others) that may provide greater levels of flexibility and reusability when evolving information systems (de Cesare, Foy and Partridge, 2013). Therefore, developing ontologies based on patterns together with a recognised framework may also help improve the quality of the overall results. Fowler (2002) asserts, in relation to software development that design patterns capture and synthase the essence of good design and are gleaned from field practice. Consequently, Fowler (2002) suggests that patterns are discovered rather than invented. Patterns can also be applied to ontology development, therefore rather than starting from a blank sheet of paper, a domain level ontology design can be constructed from the designs drawn from a pattern library – in this case the BORO foundation. The domain ontology design process therefore involves the analysis of the domain (universe of discourse) and the selection, adaption and combination of the relevant patterns drawn from this foundational ontology.

2.5.4 Foundational Ontology Patterns

The following is a non-exhaustive list of a number of the main patterns.

Pattern	Description
Super-Sub Type	A super-sub type is a set theoretic basic relationship, the pattern establishers a relationship between two types that asserts that one type is a specialisation of another type.
Instance Of	Asserts that an Element, tuple or Type is a member of a Type
Power Type Instance	Asserts that a Type is the Power Type of a Type (set of all possible sets).
Power Types	A Powertype is defined as the set of all subsets that can be created from a set of either individuals or sets. PowerTypes, are represented within the Ontology however, they are never fully instantiated rather they provide a method of modelling higher order objects – such as Types containing instances of Types (second order). A Powertype can also have a Powertype instance. This allows third order, forth order (ad infinitum) objects to be modelled. Thus PowerTypes can be employed to model the structures that form classification systems.
Whole-Part	The whole-Part tuple asserts that one (part) Element is a part of another (whole) Element. Object semantics supports both abstract concepts of; whole-part relation where the whole can either be a proper part or an improper part, i.e. a part of itself.
Temporal Objects	Object semantics physicalises time, thus a time period becomes a physical Element – a spatio-temporal object. The time dimension can be measured as a time period, for example a day has a spatial dimension that is all of space between the start and end events of the time period.
Event Temporal Objects	Under 4D semantics the distinction between a time period and an event is that events do not persist through time and are defined as a slice of a four-dimensional extension with zero thickness along the time dimension. Consequently the distinction between physical bodies such as a period of time and an event is that the former, as it persists through time, is a four-dimensional object and the latter, as it does not persist through time is a three-dimensional object.
Temporal Whole-Parts	As object semantics physicalizes time we can apply mereology in the same manner as the spatial whole-Part relationships. Therefore the Temporal Whole-Part tuple can assert that one (part) Element is a part of another (whole) Element.
States	A state is a temporal part of an individual that persists through time. States (and elements in general) are bounded by events. A state can have further temporal parts such as sub-states and events.
The happens–at (whole–part) tuples	The happens–at (a time) pattern provides a relationship between an event object and the spatiotemporal object that it is a part of. Object semantics proposes that an event a three dimensional object

Table 2.7: BORO Patterns (Partridge, 2005)

Pattern	Description
The happens–to (whole–part) tuple	An event - can be associated with the object that the event happens to by the happens–to' tuple. The event that is associated with object by the whole part relation. This is a temporal mereological relationship; the event is a temporal part of the object that it effects. In object semantics, the 'happens–to' tuple is an extension (a composite of the two member extensions) and so an object within the scope of the paradigm that forms a connection between the extension of the event that is a part of the extension of the object that the event happens to.
Names	Within BORO 'Names' are physicalised; they are considered to have spatio-temporal extension and as they are defined by the fact they name something (one element within the ontology) they are a derived type. To provide clarity within the ontology, 'names' have a separate structure from the things they name. The BORO also provides representational, symbolic based patterns. A name will belong to a Name Space which holds all names related to a particular naming authority or domain. As the OP adopts (Strawson, 1964a) theory of utterances – where each utterance of a name is an individual event and so has an extent. Therefore, a name is a type (set) of the utterances of the same name rather than individuals.

Table 2.7 Continued: BORO Patterns

Thus, by employing the common patterns of the foundation, each of the domain ontologies that are developed (including Types and instance level objects) is integrated with, and grounded by the foundation. Furthermore, as will be described in the following chapters that detail the design science iterations, the same domain patterns that are developed at design-time, are also loaded to the graph database (run-time environment) to become the 'schema' that is common to all datasets that are integrated. Therefore, the common foundation and domain patterns are essential to the achieving semantic data integration.

2.6 **Foundational Ontology Driven Semantic Data Integration**

The literature identifies a number of approaches to the solving the semantic data heterogeneity problem. Common to all such solutions, is the process of unifying data based upon common semantics. This process involves interpreting and matching evidence from:

a) relations, names, types, attributes and constraints gleaned from the data source model;

b) instance-level evidence available such as field values;

c) other external evidence such as documentation (Noy, 2004).

There remains, however, the problem of establishing the canonical semantics through which such integration can be undertaken. A foundational ontology can solve this problem by providing a philosophically grounded domain independent system of formal categories (Guizzardi, de Almeida Falbo and Guizzardi, 2008). To fulfil the pragmatics of instantiating information systems, such foundational ontologies can also be reflected in the form of meta-model artefacts that contains a set of the most generic objects and relationships. This in turn can be extended (specialised) to represent objects and relationships specific to a domain of interest. Thus, to effect semantic integration the domain specific objects and relationships can be interpreted and translated to conform to the foundation and are thus semantically aligned, or as described by Rector (2003) semantically normalised. In the content of this research, semantic 'normalisation' relates to the process of ensuring a modelled element is categorised as only one foundational object (element, type or tuple) and that it is assigned the correct position and relations within the ontological structure.

2.6.1 Semantic Extraction

Input data to a semantic integration process may be structured in many forms such as fixed record or delimited tabular files, RDF, RDFS, OWL etc. and may consist of both model (schema) level and or instance level data. In addition, as described by Ahmad and Odeh (2012) process models such as Business Process Modelling Notation models may also serve as a source data to the semantic extraction process.

To overcome the difficulties related to integrating models that employ different modelling structures Kappel *et al* (2006) describe a process of 'Lifting' which transforms a model, such as the simple row and column structures found within tabular data, into an ontology. This is similar in nature to the process described by William, Johannesson and Bubenko (1996) who introduced a process which extracted a conceptual schema from a relational schema. In a research paper published on the subject, Myroshnichenko and Murphy (2009) present a method to automatically map well-formed ER schemas into the semantically equivalent ontologies represented in OWL. Table 2.8 describes the ER elements and the equivalent OWL ontology objects.

ER Object	Ontology Object (OWL)
Entity	Class
Strong Entity	Class
Weak Entity	Base class and additional class encapsulating equivalents of the partial key attributes and the key attribute of the owner entity
Attribute	Datatype property
Single-valued Attribute (nullable)	Functional datatype property
Single-valued Attribute (not nullable)	Functional datatype property with min constraint set to one
Multi-valued Attribute (nullable)	Datatype property
Multi-valued Attribute (not nullable)	Datatype property with minimum constraint set to one
Key Attribute	Functional datatype property with minimum constraint set to one
Composite Attribute	Class with properties corresponding to components of the composite attribute
Binary Relationship without Attributes	Pair of inverse object properties
Binary Relationship with Attributes	Class with datatype properties corresponding to the relationship's attributes and two pairs of inverse object properties associating the participating entity classes and the relationship class
Ternary Relationship	Class with three pairs of inverse object properties associating the participating entity classes and the relationship class
Participating Entity Role Name	Name of the appropriate object property in the pair of inverse object properties manifesting a relationship without attributes
Min Cardinality Constraint	Minimum cardinality restriction
Max Cardinality Constraint	Maximum cardinality restriction

Table 2.8: Mapping ER Schemas to OWL Ontologies (Myroshnichenko and Murphy, 2009)

Although very effective at mapping the syntax, such approaches do not deal with the fundamental differences between the modelling paradigms; i.e. ER closed world UNA versus OWL open world semantics.

Thus the first stage in semantic integration process begins with 'lifting' the source data and its associated model so they conform to an ontological representation in terms of structure and semantics. This can be considered a semantic extraction process which interprets the instance level data, the model to which it conforms and other evidence available such as background information. This process results in a domain ontology. The second stage of the process involves the transformation to the paradigm of the target foundation ontology i.e. semantically grounding the modelled elements by classifying them as one of the foundational objects

2.6.2 Semantic Transformation

The initial semantic extraction produced a domain ontology that conforms to endurantist semantics i.e., Entities Types, Entitles, Attributes Types and Attributes. Therefore, before integration, these semantics need to be transformed to align with the foundational ontology, for example, in the translation from 3D to 4D semantics. The BORO foundation provides a view of reality and the patterns that can be employed perform this translation, for example, translating an 'Entity' from a three dimensional extension to a four dimensional spatio-temporal extent (this process is described in detail within chapter four of this thesis). Therefore the foundational ontology provides the equivalent of a canonical data model (Saltor, Castellanos and Garcia-Solaco, 1991) that can be employed to develop and integrate other domain models providing the semantics that are common to all data sets that will be integrated. Thus the translation process results in a new domain ontology that extends the ontic categories and patterns of the foundation. Through this process, domain ontologies are developed to represent the entities and relationships that are represented by the data.

2.6.3 Semantic Integration – Ontology Matching

Once a number of such models have been created in an ontologically consistent form the semantic matching process can be undertaken. Ontology matching has been an active research area with a number of techniques being applied that result in varying levels of accuracy. In terms of undertaking such matching, according to Kalfoglou and Schorlemmer (2003), there are a number of categories of solution. Table 2.9 provides an overview of these solutions.

Category	Description
Frameworks	A combination of tools, they provide a methodological approach to mapping. Some are also based on theoretical work.
Methods and tools	Tools, either stand-alone or embedded in ontology development environments, and methods used in ontology mapping.
Translators	Translators are normally employed at the early stages of an ontology mapping project to assist rather than undertake the complete processes
Mediators	Mediators provide useful information input to the mapping programs. Normally employed at the early stages of an ontology mapping project to assist rather than undertake the complete processes
Techniques	This is similar to methods and tools, but not so elaborated or directly connected with mapping.

Table 2.9: Semantic Integration Solutions (Kalfoglou and Schorlemmer, 2003)

Shvaiko and Euzenat (2013) describe ontology integration as a matching operation and provide an example which is depicted in the following Figure 2.7.

The matching operation determines an Alignment Ontology for a pair of ontologies input Ontologies (1 and 2) by finding an alignment between these ontologies. This process can be supplemented by i) using an input Alignment Ontology which is extended; ii) the use of

matching parameters such as thresholds and weights; and (iii) external resources, such as common knowledge and domain specific thesauri (Shvaiko and Euzenat, 2013). This alignment process can employ one or more of the following methods: terminological, structural pattern, extensional (based on a reference to individual instance) or based on annotation such as background information (Shvaiko and Euzenat, 2013). Automatic techniques are applicable when matching very large ontologies or schemas, as for example found within the biological sciences. A manual process is feasible when the ontologies to be matched are on a smaller scale and a high level of precision is required (Shvaiko and Euzenat, 2013).

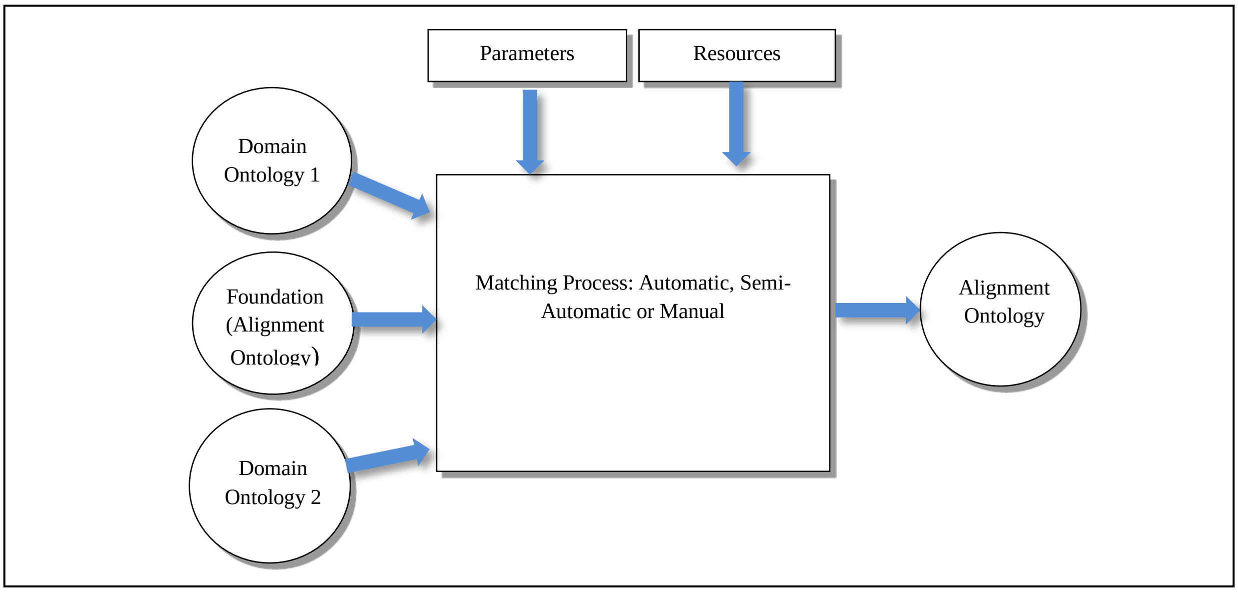

Parameters
Resources
Domain Ontology 1
Foundation (Alignment Ontology)
Domain Ontology 2
Matching Process: Automatic, Semi-Automatic or Manual
Alignment Ontology

Figure 2.7: The Ontology Matching Process (Shvaiko and Euzenat, 2013).

2.6.4 Semantic Integration – Loading Data

In addition to containing a terminological model (such as a class hierarchy), a dataset may also contains instance level data, such information needs to be extracted from the source dataset and transformed and integrated with canonical ontology and other related instance level models objects that may exist in the combined ontology. Through this process the integration of individual elements takes place. This can be considered as integration within vertical and horizontal plains. Firstly the vertical relationships between an individual element and the domain ontology (and hence the foundation ontology) must be established which consists of establishing the individual's relationships (such as type instance, set member, part of), then establishing the 'horizontal' relationships that are deemed to hold between individual domain level objects (e.g. an individual company being located at a particular geographic location). Foundational ontological patterns can then be applied to simplify this process. This can be a complex transformations that requires both one-to-many and many-to-one transformations.

2.6.5 System Realisation

The results of a semantic data integration process will normally be taken forward to some form of system realisation of which there are generally understood to be two primary types of solution available namely; a distributed database or centralised warehouse database where the datasets are combined. The database itself may be in the form of a RDBMS, triple store or other system that provides a means of persisting data to storage. Figure 2.8 depicts the distributed database architecture approach, which typically employs mediation via a message bus or a hub and spoke which is used to connect each of the distributed database members.

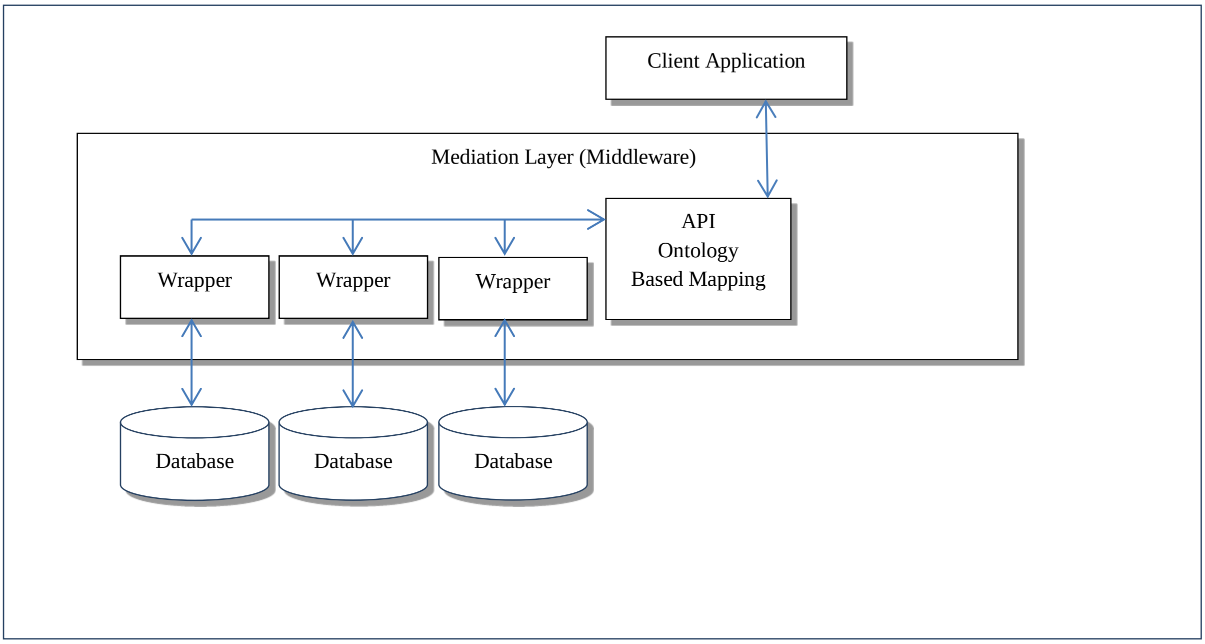


Figure 2.8: Ontology Mediated Integration

Each connected database has its own ‘wrapper’ which transforms inbound updates and outbound data requests based on the map produced in the matching process. Such solutions are typical of Enterprise Application Integration (EAI) middleware architecture. The operation of a distributed ontology mediated system involves the use of mappings that describe the relationship between the terms of the ontology and their representation in the each of the data sources. One example of such a solution is OntoQF (Munir, Odeh and McClatchey, 2012) a system that provides query formulation services to assist a user in information search and retrieval processes by providing a means of generating RDBMS queries. This approach employs ontology to supplement a relational database schema with one or more semantic domain models. OntoQF has been applied in the field of medical knowledge engineering to assist clinical researchers in generating relational database queries by providing methods that interpret and transform OWL-DL expressions into relational expressions based on the Relational Algebra (RA). This enables users to formulate RA queries without having specific knowledge of the information structure and access mechanisms of the underlying RDBMS data source. OntoQF achieves this integration without recourse to changes or replication of RDBMS based data sources. The solution thus enables a user to formulate a class description employing OWL-DL statement constructs. These constructs are translated into the corresponding executable relational query. This process employs ontology to database schema maps that provide information relating to ontology properties (object and datatype), database name, table names, column names and primary and foreign keys.

According to Munir *et at.*, (2012) In addition to the translation of OWL-DL to SQL and the enrichment of the RDBM schema with semantic knowledge, another merit of this approach is that no OWL-DL based interpretation of RDBMS stored (bulk data) is required and hence such data do not need to be stored or processed as ontology instances, thus avoiding the scalability issues.

The alternative to such mediated approaches is to employ a warehouse based approach which involves the Extraction Transformation and Loading (ETL) of data from each of the source datasets to a warehouse database system.

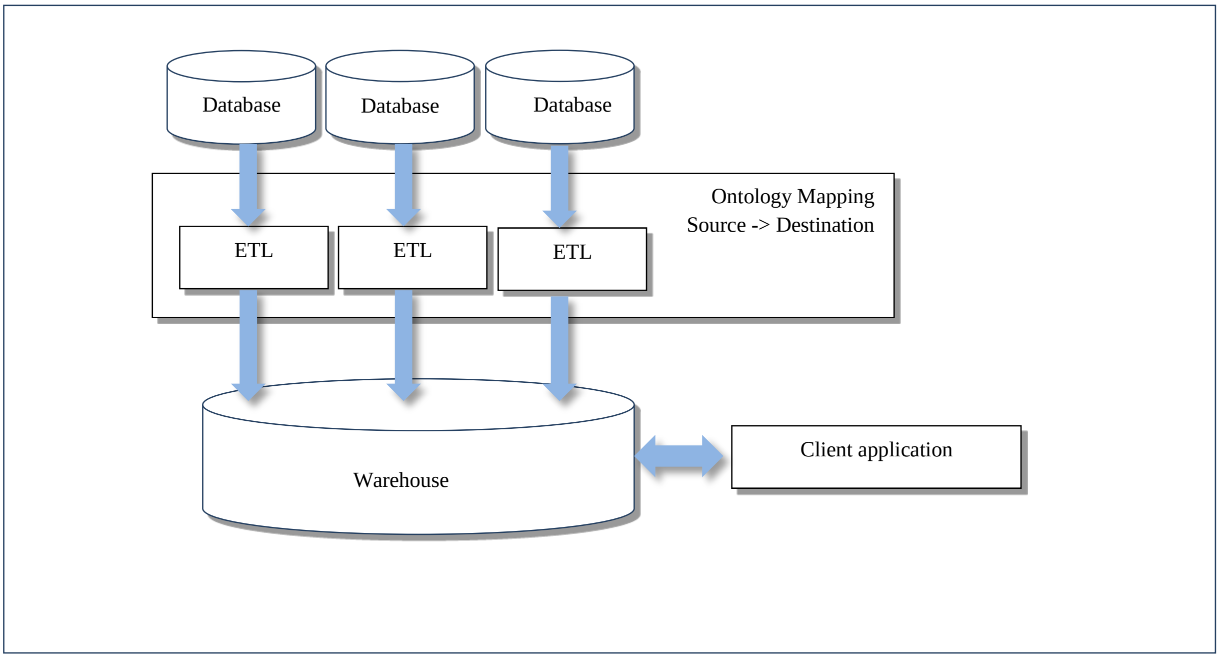


Figure 2.6: Centralised warehouse based integration

In relation to the implementation of such warehouse systems, the ability to employ new types of persistence structures has come about through the major technological shift occurring within the field of computing and the availability of new architectures based on NoSQL software that arose within Facebook, Google and Amazon who found that existing RDBMS technology was not capable of handling their 'big data' processing requirements (Strozzi, 2010). NoSQL, also known as Polyglot Persistence, are systems that provide new ways of persisting information which in many cases do not have fixed schemas and are therefore capable of storing information without imposing restrictions on the structure or the values of the key-value pairs, documents or nodes/edges stored (Fowler and Sadalage, 2012). There are a wide range of such NoSQL systems available each of which are designed to solve different problems, however, in the context of this research a Graph Database is selected as it provides that ability to store the ontology (model) and instance level objects (data) in its modelled graph form without the need for extensive model translation (from graph to tabular form).

Graph Database

The specific software selected is Neo4J (Neo Technology, 2015) an open-source NoSQL graph database implemented in Java and Scala that supports the property graph model. The model contains three primary elements; nodes (vertices), relationships and properties. Nodes are connected via directed edges (termed relationships) both of which can have a set of associated properties in the form of key-value pairs. The properties associated with both nodes and edges can be indexed to provide a means of navigating the graph to start a traversal. A traversal is the primary means of querying the property graph database (Neo Technology, 2015) .

Although a number of other Graph databases were considered for the project, Neo4J was selected as it supports: large scale graphs (i.e. a combination of both in-memory and disk based storage), transactions (ACID compliant), bulk load facility, has a well-defined declarative graph query language (Cypher), and has an Application Programmers Interface (API) that supports a language the researcher is familiar with (Java). Furthermore an open source version is available.

Graph Database Query Processing

Neo4J (Neo Technology, 2015) provides a declarative language for describing select, insert, update or delete operations that can be performed on a graph database. Graph queries operate in a similar manner to SPARQL to match a specific pattern, then to retrieve the relevant node properties. Neo4J also provides a range of common graph algorithms.

Alternative OWL Based Technology

There is a choice between rule based reasoners that scale well for instance level knowledge (ABox) but are limited to simple DL fragments, or tableau reasoners that scale well for complex terminological knowledge (TBox) but are limited in their ability to process large instance level knowledge (ABox) (Bock *et al.*, 2008). The former could employ RDBMS in most cases, the later due to being memory constrained is not suitable for projects (such as this) that intend to integrate large datasets.

2.7 Literature Review Summary

The following section provides an overview and summary of the literature review findings and identifies a number of weaknesses that are inherent to current approaches to semantic integration.

The problem of semantic data integration has been well researched by the database and Semantic Web community (Doan, Noy and Halevy, 2004; Kalfoglou and Schorlemmer 2003; Saltor, Castellanos and Garcia-Solaco, 1991).

Semantic data integration is difficult to achieve as evidenced by the existence of information silos - islands of applications that were not designed to interoperate or integrate (Arsanjani, 2002). Information silos arise due to the tendency for organisations to implement new information systems to support immediate business requirement rather than trying to ‘bolt-on’ new functionality to centralised enterprise systems. This results in data that are distributed amongst disparate incompatible packaged/bespoke applications (Katasonov and Lattunen, 2014).

Incompatibility arises due to the differing abstractions of reality that are inherent to information systems designed for different purposes by different people (Kent, 1978) and the differing semantics of the modelling tools and runtime environments.

Integrating external data from the Semantic Web and Open Data sources will exacerbate the problem of semantic data integration by introducing a vast range of data together with the implicit and explicit models that underlie such data.

The problem of semantic data integration requires a solution that will enable the semantics of the original model to be extracted (recovered) (Roddick, 1995), and to be translated to a common form (semantically normalised) (Rector, 2003). A foundational ontology can provide a grounded view through which to unify these models (Pease and Niles, 2002). However, there is a range of meta-ontology (metaphysical) theories that provide a criterion for each of the ontological commitments, which are principally the things believed to exist within the context of a particular theory (Bricker, 2014). As asserted by Partridge *et al.* (2013) due to the increasing use of semantics within Information Systems there is an emerging need for understanding ontology in the philosophical sense. Table 2.10 provides a summary of the problems of semantic integration and the weaknesses in current approaches. Therefore to address these semantic integration problems, there is a need to research the new approaches – such as 4D approach to semantic integration and how it can be applied with an emerging high performance scalable graph database technology.

Weakness	Description
Lack of grounding	Many current models employed within information systems have no form of grounding in a more fundamental theory (Cregan, 2007). Thus the ontological commitments underlying the model are unknown. On examination of many Linked Open Data ontologies, they are often ungrounded – or rely or reference other ungrounded ontologies.
Model Strata and translations distortion	The strata of models formed by the software design process and the requirement to translate the high level models of reality that are created at the initial design time into a series of tabular statures that are focused of the execution environment can lose direct traceability to the initial model. This is analogues to the oft cited, within IS practice OO-RDBMS impedance mismatch (Ireland *et al.*, 2009).
Over simplification to fit a model of reality to a tractable theory	The need to simplify the abstraction of reality to it can fit neatly into a FOL theory, thus ignoring the fact that reality is not so simple and higher order objects exist (Bailey, 2011).
Integrating models which are founded on different semantics	There are many automatic translation techniques for translating RDBMS schema and data to an OWL 'ontology'. However, there is a lack of recognition of the semantic differences that underlie the differing modelling constructs.
Dividing models into static and dynamic types	The separation of static and dynamic aspects of reality into different structural and process models leads to the development of incompatible abstractions together with exotic relations such as trans-ontological relations that are employed to bridge these static and dynamic worlds.
Naming and meaning confusion	That there is often naming and meaning confusion, as described by Frege (1948). The objects place in reality (and the ontology) define its meaning.
Establishing identify	Many modelling and information systems use ephemeral means of establishing an objects identity which do not function well over time.
Employing techniques that do not scale	Many of the software tools such as OWL tableau calculus based reasoners, as they are constrained by memory, cannot scale to inference over ontologies containing large scale instance population (Bock *et al.*, 2008). The alternatively is to use simplified semantics and rule based reasoning - that could in many cases employ standard RDBMS techniques.

Table 2.10: Summary of Existing Semantic Integration Weaknesses

2.8 Research Direction

Within this chapter, literature has been reviewed which has examined and elaborated the problem of semantic integration and provided an introduction to the subject of ontology and foundational ontologies together with the diversity of metaphysical theories that underlie such ontologically models. The BORO perdurantist foundational ontology has been presented and discussed together with the foundational patterns that are inherent to the ontology. The review has examined literature related to the use of ontologies within semantic integration. The way that ontology has diverged into two forms; the information systems ontology and ontology in the philosophical form was described. The information systems ontology is considered to be an engineered artefact which is developed for a specific purpose (Gruber, 1993), whereas philosophical ontology are routed in metaphysical theories that reflect a fundamental view of the world. The theories that underlie ontologies were outlined together with the differing forms of foundational ontologies that they produce; realist, revisionary, cognitive (bias) etc., were discussed. Finally a number of weaknesses have been identified that apply to current approaches

Therefore a justification has been presented for the assertion that foundational ontologies can make a contribution to solving the problem of semantic data integration and that, consequently, there is a need to better understand their application within an IS semantic integration context.

CHAPTER 3. RESEARCH DESIGN

3.1 Introduction

It is appropriate that prior to embarking on a research project, that the researcher is aware of the valid research methods and how such methods may be applied to guide the research envisaged. Therefore, as a precursor to any course of research, it is necessary to develop the knowledge and understanding necessary to make informed decisions regarding the selection and application of such methods. This chapter presents literature relating to design science, its application within Information Systems (IS), the types of artefacts such research can produce and the evaluation criteria that are relevant to such artefacts. The chapter then provides the reader with details of the research design for this study and the ways in which design science has been applied to the research problem. Thirdly, a plan is presented that will be employed to guide the activities undertaken during the course of this research project.

It may be possible to conduct IS research that yields valid results using other methods and frameworks; however, the researcher considered the guidance provided by a recognised design science framework was valuable in ensuring the research was conducted with rigour and employed a methodology that would be familiar to other researchers in the field.

3.2 Design Science Research within Information Systems

March and Smith, (1995) assert that human purposes can be met through the implementation of Information Systems (IS), which provide the means of capturing and processing information. The implementation of IS involves the instantiation of systems that form a complex organisations of hardware, software, procedures, information and people (March and Smith, 1995). Furthermore, Hevner, March and Park, (2004) state that within IS research there are two dominant paradigms that can be used to characterise such research, behavioural science and design science. Behavioural IS research is primarily focused on a form of human or organisational behavioural investigation and, as stated by Geerts (2011), follows a well-defined path of: problem definition, literature review, hypothesis development, data collection, analyses, results and discussion. Such research can be categorised as descriptive research; i.e. explaining and understanding the status-quo whereas design science, through the development of new artefacts that serve human purposes, offers ways of improving a state of affairs and can therefore it can be classified as prescriptive research (March and Smith, 1995). Consequently, when the subject of a research investigation relates to engineered rather than natural phenomena, design science offers a similar well-defined path which consists of the established methods for undertaking such research. The early work of Simon (Simon, 1996) provided much of the foundation for design science, which he described as the science of the artificial; an analogue to natural science. Design science is intrinsically a problem-solving paradigm that employs creativity and trial-and-error research which aims to develop new and innovative artefacts (Hevner, March and Park, 2004). It is through this problem solving process of artefact design, implementation and evaluation that new knowledge is acquired (Hevner, March and Park, 2004).

Therefore, when it is theorised that existent engineered artefacts and processes through which they are produced can be improved in some way, design science offers a valid research methodology through which to plan and execute a course of investigation to test the theory. Design science research is primarily conducted via the execution of empirical research that is undertaken through a number of cycles composed of discrete design, build and evaluation stages, the latter of which enables a judgement to be made as to whether the core aim of the research has been achieved (Simon, 1996).

As asserted by Hevner *et al*., (2004) design science research methodology when applied to information systems consists of a combination of descriptive methods drawn from behavioural science and prescriptive methods drawn from design science. Thus Hevner *et al.* (2004) state that such research provides facets of design science that relate to the utility of engineered artefacts within the context of the environment they may inhabit, i.e. the impact on people and organisations that may employ them, and the facets that relate to the learning process inherent in the design process itself.

Hevner *et al*., (2004) assert that for any course of investigation to be considered valid information system design science research (and not routine design), it must meet the fundamental requirements of: having a question that is framed in terms of a problem that is both significant and of relevance to business; that it is conducted with rigour; and that the research includes a core design-build-evaluate cycle which encompasses multiple iterations. Rigour is attained by referencing and applying relevant theories from the knowledge base. The relevance of the research can be assessed by judging whether the artefact satisfies an actual business need. The process through which the research is conducted is described by Hevner *et al*., (2004) within a research framework which is depicted in Figure 3.1.

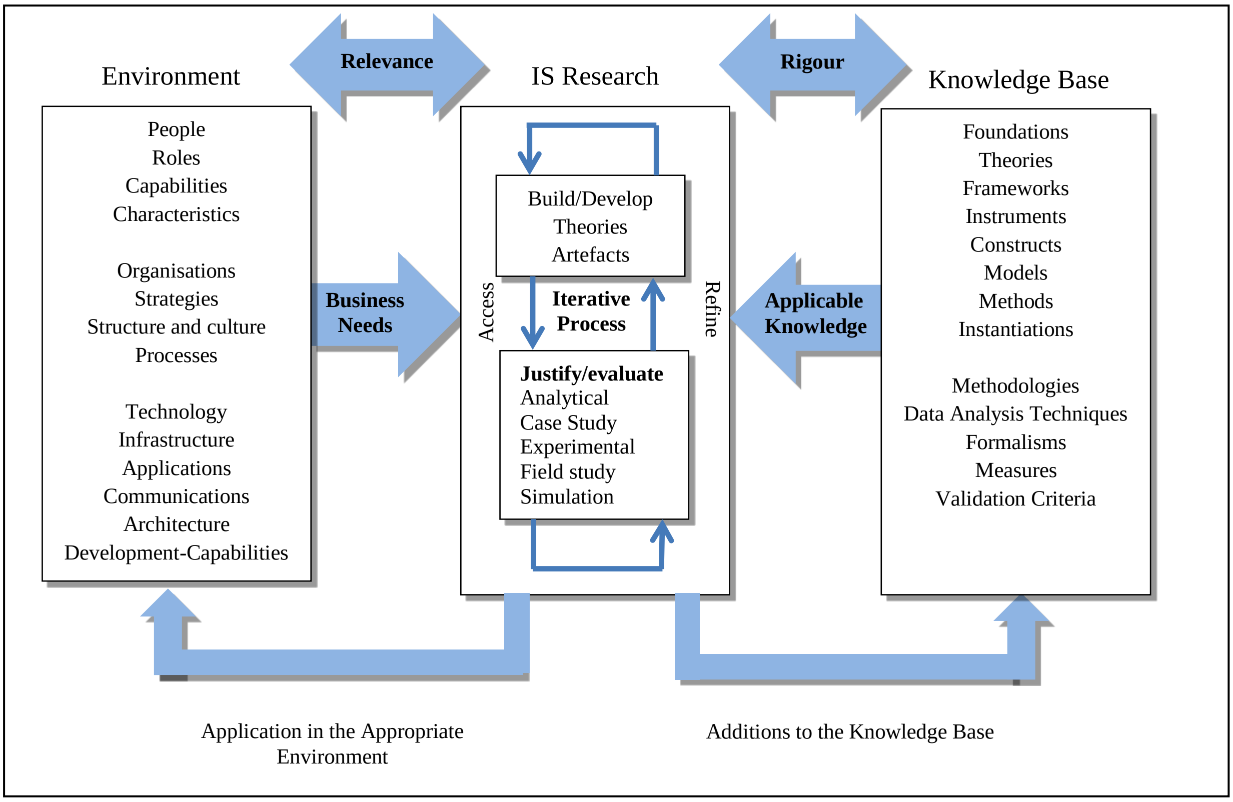
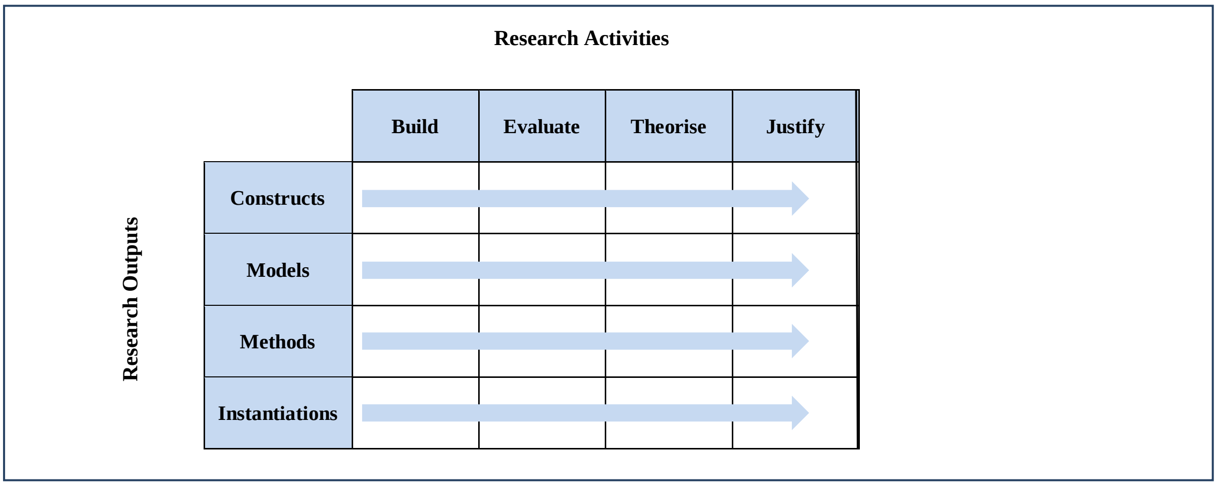


Figure 3.1: Information Systems DR Framework (Hevner, March and Park, 2004)

March and Smith (1995) state that when design science is conducted within the field of IS, four major activities and four related research output artefact types can be identified. These activities and outputs are depicted as matrix dimensions within Figure 3.2.



	Build	Evaluate	Theorise	Justify
Constructs				
Models				
Methods				
Instantiations				

Figure 3.2: A Design Science Research Framework (March and Smith, 1995)

3.2.1 Research Activities

March and Smith (1995) state that to ensure that design science is effective and of relevance, both design science and natural science activities need to be undertaken during the course of the research. The research activities they identify are detailed in Table 3.1.

Activity	Description
Build	The build DS research activity refers to that act of instantiating the artefacts that will be the subject of the research investigation. These artefacts can include constructs, models, or instantiations and the methods through which these artefacts are developed. Within this thesis, the term 'implementation' is also employed to refer to this DS build activity.
Evaluate	The evaluation activity refers to the assessment and evaluation of the DS artefacts produced by the research. This is performed against a set of predefined metrics. Evaluation is an essential activity of DS as it enables the researcher to establish the degree to which the artefact being assessed meets the aim of the research.
Theorise	It is important to develop theories that elaborate the reasons why an artefact resulting from the research performs in a particular way or has particular properties.
Justify	Through evidencing the results of the artefact evaluation and explaining or citing theories that explain such results, the researcher can justify the judgement as to the degree the research met the initial aim and the value of the finding of the research to the corpora of knowledge related to the domain being investigated.

Table 3.1: Research Activities (March and Smith, 1995)

3.2.2 Research Outputs

The research artefacts identified by March and Smith (1995) are detailed in Table 3.2.

Feature	Description
Constructs	According to Hevner *et al.* (2004) constructs are constituted by the specialised concepts that form a vocabulary that describe the knowledge within a domain and therefore can be employed top model domain problems and solutions.
Models	Models are abstractions of the real world situation that describe the design problem and solution space. March and Smith (1995) assert that a such models form a set of propositions or statements that express the relationships that hold among constructs.
Methods	Methods describe process stages that are necessary to solve a particular problem and relate the constructs and the models previously defined. Methods can be considered tools that can be created by design science and applied by the researcher (Newell and Simon, 1976). March and Smith (1995) state that methods by asserting that a method is "a set of steps (an algorithm or guideline) used to perform a task. Methods are based on a set of underlying constructs (language) and a representation (model) of the solution space. … Although they may not be explicitly articulated, representations of tasks and results are intrinsic to methods. Methods can be tied to particular models in that the steps take parts of the model as input. Further, methods are often used to translate from one model or representation to another in the course of solving a problem."
Instantiations	Newell and Simon (1976) state that instantiations are significant artefacts within the field of DS research as they provide a better insight into the problem domain and consequently offer the prospect of the research providing a better solution. Furthermore March and Smith (1995) assert that such instantiations can be employed to ascertain the feasibility and effectiveness of the models, methods and constructs by facilitating actual rather than theoretical evaluation.

Table 3.2: Research Artefacts (March and Smith, 1995)

3.3 The Overarching Methodology

Vaishnavi and Kuechler (2004) build on the previous work of Hevner, *et al.* (2004) to synthesise a methodology specifically designed for executing design science research within IT that encompasses both the methods through which the research can be undertaken and the resultant artefacts assessed and evaluated. This overarching methodology consists of a number of stages that can be employed to guide the design and execution of a design science project. This methodology is depicted in Figure 3.3.

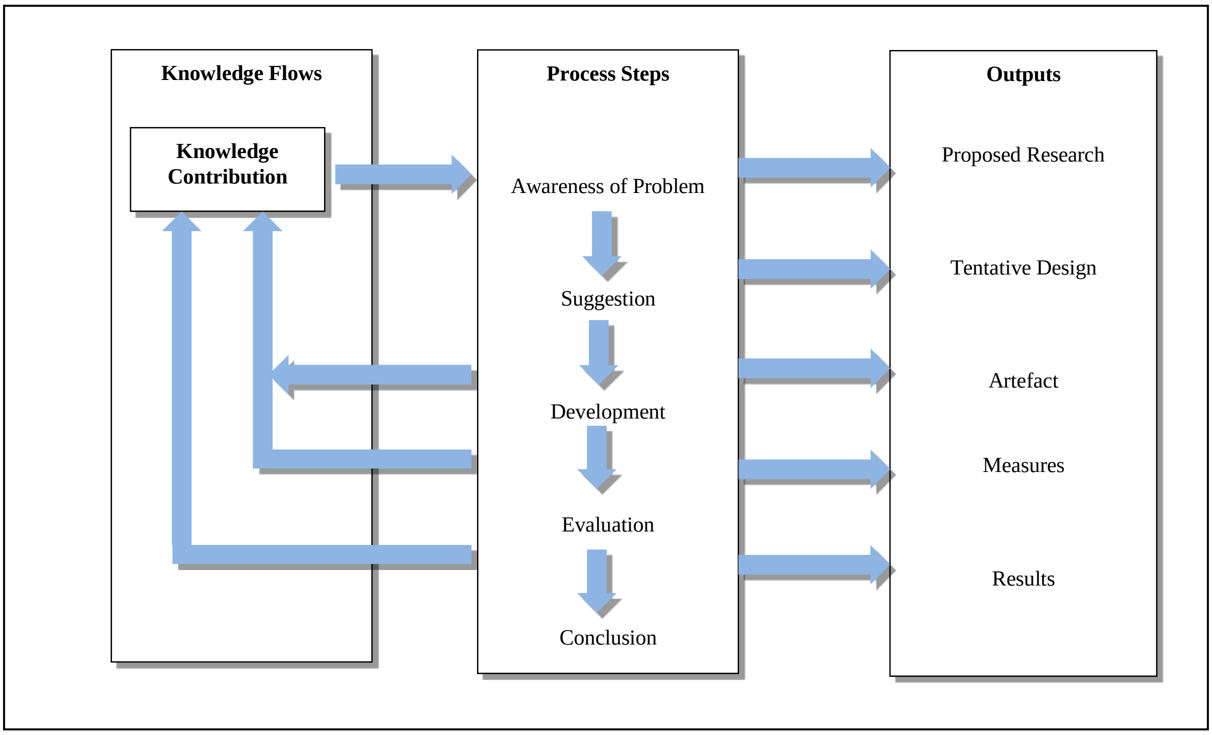


Figure 3.3: Design Science Process (Vaishnavi and Kuechler, 2004)

3.3.1 Problem Awareness Stage

The research problem is defined during the design science initial stage, the 'Relevance Cycle', and is motivated by a desire to improve the environment by the introduction of new and innovative artefacts (Simon, 1996). The research problem in this study is semantic data integration as discussed within Chapter two.

3.3.2 Suggestion Stage

The ‘suggestion’ stage in the process defined by the design science methodology consists of introducing an identified research problem and how it might be solved. In our case, the problem space is semantic data integration and the proposed solution is to use 4D foundational ontology and graph database.

3.3.3 Development and Evaluation Stages

The development and evaluation of artefacts in the form of process designs and the instantiation of a prototype system are carried out by during the course of the core empirical design-develop-evaluation research activity that is central to the design science methodology. These activities are presented in detail in the following section that describes each of the three interactions of this cycle.

3.3.4 Conclusion Stage

Intermediate conclusions are drawn at the following the evaluation process at the end of each of the three core empirical design-develop-evaluation iterations. However, final conclusions are presented in details within chapter seven.

3.4 The Design Science Core Design-Build- Evaluation Cycle

At the core of the design science methodology is the design-build-evaluate iterative cycle through which the research is conducted. This section describes these activities. The cycle is executed in an iterative manner with each iteration providing knowledge that informs subsequent iterations. Hevner *et al* (2004) assert that it is this core cycle that forms the interface between design science and behavioural science with truth being provided by relevant theories utilising design science information system artefacts. The core process commences with a theory and proceeds through the course of three problem-solving cycles that build and evaluate artefacts with the aim of developing knowledge relevant to subject being investigated. It is through this process that valid information systems research is achieved.

3.4.1 Design

According to Simon (1996) who states, in relation to design:

> “Engineers are not the only professional designers. Everyone who devises a course of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artefacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences. Schools of engineering, as well as schools of architecture, business, education, law, and medicine, are all centrally concerned with the process of design” (Simon, 1996, p.111).

Within industry, information systems design practice is constrained by timescales and financial resources, and is normally undertaken in a pragmatic manner that avoids risk by utilising proven methods and architectures. Whereas within information system design science research, design can be undertaken in the context of an exploration of new and innovative ways through which to solve information systems related problems. Therefore within design science the artefacts being developed are being instantiated, not for immediate commercial gain or to solve an immediate technical problem, but rather to serve the purpose of answering a research question. Thus, design science research through the act of design can explore new possibilities that are outside of current design practices used within a problem setting (Schon, 1992). Although the primary purpose of the development of such artefacts is the furtherance of academic knowledge, design science research must also meet the requirement of addressing an important problem that is of relevance to business. It is through this focus on business solutions and the objective of improving IS systems that design science offers the means of addressing the relevancy gap that has often beset academic research within the field of information systems (Walls, Widmeyer and El Sawy, 1992).

In addition to the design activities related to the artefacts that are the subject of the investigation, the experiment through which such artefacts are designed, validated and evaluated must also be undertaken. This process is a fundamental component of the empirical design-build-evaluate research cycle that is at the core of the DS research project.

3.4.2 Instantiation (Build)

As stated previously, the build design science research activity refers to that act of instantiating the artefacts that will be the subject of the research investigation. These artefacts can include constructs, models, or instantiations and the methods through which these artefacts are developed.

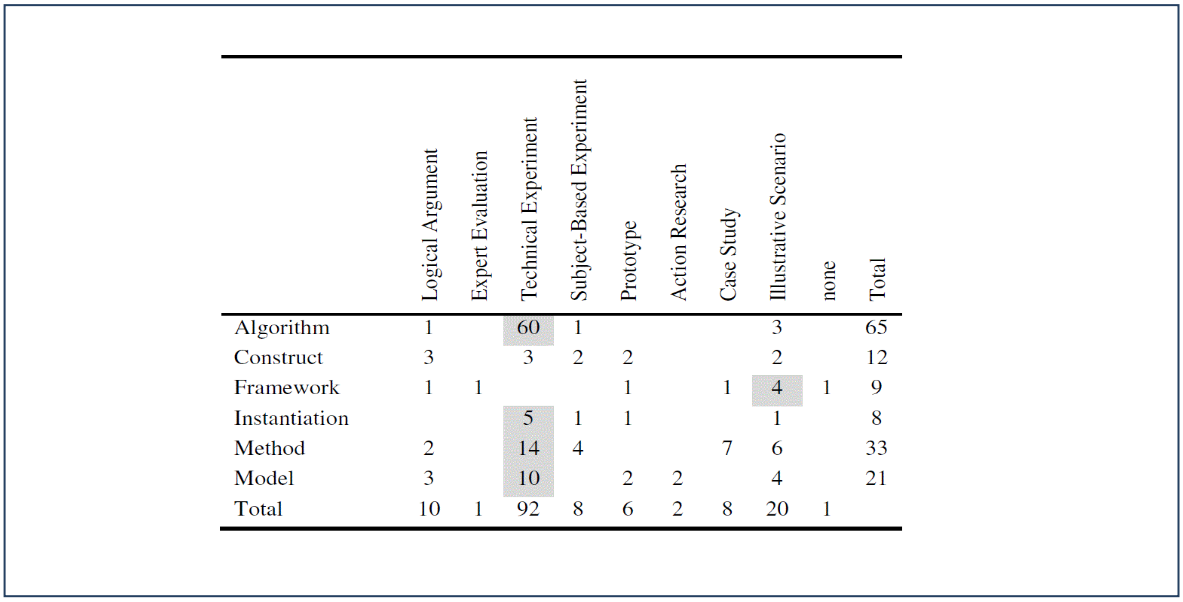
3.4.3 Evaluation

The evaluation activity is a core activity of design science as it establishes the degree to which the research artefact being assessed meets the aim of the research. Therefore, to ensure the project complies with rigor essential to valid research, clearly defined methods are required through which the researcher can measure and assess the progress of the research project. Hevner *et al.* (2004) state that the evaluation method and metrics employed should be carefully matched with the artefact that is the subject of the investigation. Their classification of evaluation methods and the guidance as to the appropriateness of each method when assessing an artefact is provided in table 3.3.

Evaluation Method	**Application**
Observational	Case study: Study the artefact in depth in a business environment
	Field Study: Monitor the use of artefact in multiple projects
Analytical	Static Analysis: Examine structure of artefact for static qualities (i.e. complexity
	Architecture Analysis: Study fit of artefact into a technical IS architecture
	Optimisation: Demonstrate inherent optimal properties of artefact or provide optimality bounds on artefact behaviour
	Dynamic Analysis: Study the artefact in use for dynamic qualities (e.g., performance)
Experimental	Controlled Experiment: Study artefact in a controlled environment for qualities (e.g., usability)
	Simulation: Execute artefact with artificial data
Testing	Functional (Black Box) Testing: Execute artefact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Execute artefact interfaces to discover failures and identify defects
Descriptive	Informed Argument: Use information from the knowledge base (e.g. relevant research to build a convincing argument for the artefact's utility
	Scenarios: Construct detailed scenarios around the artefact to demonstrate its utility

Table 3.3: Design Evaluation Methods (Hevner, March and Park, 2004)

Peffers *et al*. (2007) published a seminal paper on the subject of design science evaluation within the field of Information Technology (IT). This research constituted a literature based study that analysed research papers published within a number of respected scientific IT journals that could generally be described as design science research. Their paper provides a taxonomy which covers common design science terms together with a matrix which details the relationships between design science artefacts and the evaluation methods that are applied. A summary of their finding are detailed in the figure 3.4 in matrix form.



	Logical Argument	Expert Evaluation	Technical Experiment	Subject-Based Experiment	Prototype	Action Research	Case Study	Illustrative Scenario	none	Total
Algorithm	1		60	1				3		65
Construct	3		3	2	2			2		12
Framework	1	1			1		1	4	1	9
Instantiation			5	1	1			1		8
Method	2		14	4			7	6		33
Model	3		10		2	2		4		21
Total	10	1	92	8	6	2	8	20	1	

Figure 3.4: Types of Evaluation (Peffers et al., 2007)

The quantitative data relating to the methods used to evaluate types of artefacts within design science design science provided by the Peffers *et al.* (2007) study validates and supplements the earlier work of Hevner *et al*. (2004).

3.5 The Adopted Design Science Research Methodology

3.5.1 Introduction

The design science process as proposed by Vaishnavi and Kuechler, (2004) is adopted as the overarching method for this research project. The rationale for this decision is that Vaishnavi and Kuechler, (2004) provide methodology specifically designed for executing design science research within IT that encompasses both the methods through which the research can be undertaken and the resultant artefacts evaluated. This overarching methodology consists of a number of stages that can be employed to guide the design and execution of a research project. Each of the design science process stages and how they are constituted for this research are detailed in this section.

This design science research project is focussed on the assessment of the effectiveness of employing a perdurantist (4D) foundational ontology to form a Graph Database resident 'schema' through which to semantically integrate a number of datasets and is therefore highly relevant to a business problem. The primary artefact developed is the 4D-SETL framework and the means of evaluation is via technical experiment and illustrative scenario.

The research problem was originated as a theory that 4D ontologies, when combined with a graph database, could provide a partial solution to the problem of semantic data integration. The problem of semantic data integration was defined through reference to previous literature (detailed in chapter two). This literature based research activity forms part of the design science 'Rigor Cycle' as described by (Hevner, March and Park, 2004).

3.5.2 Overview

A diagrammatic view of how each of the main stages of the overarching design science research methodology relate to the content and structure of the thesis is presented in Figure 3.5.

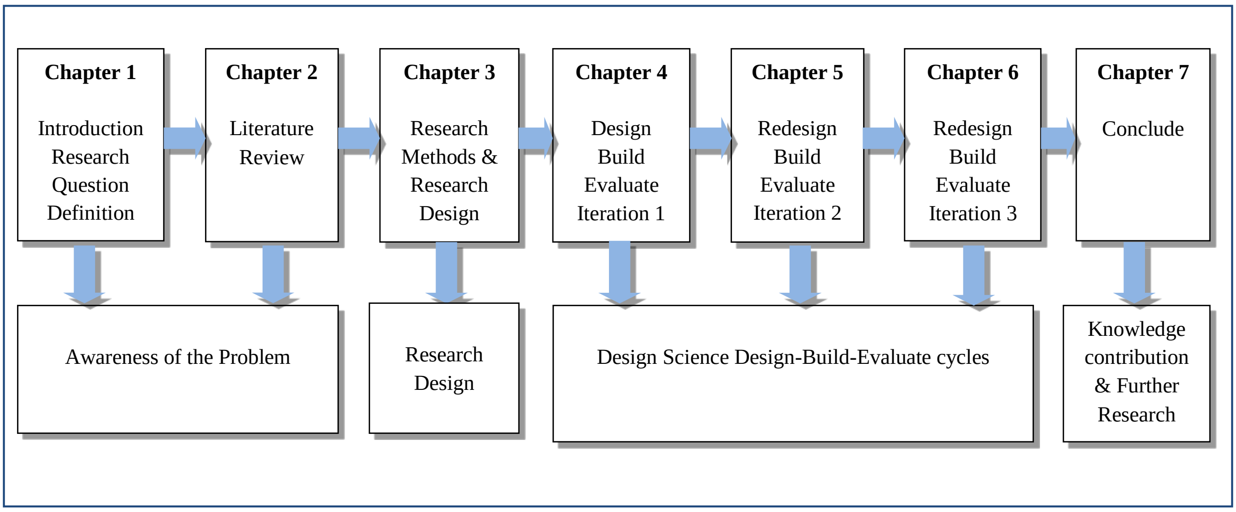

Chapter 1
Introduction Research Question Definition
Chapter 2
Literature Review
Chapter 3
Research Methods & Research Design
Chapter 4
Design Build Evaluate Iteration 1
Chapter 5
Redesign Build Evaluate Iteration 2
Chapter 6
Redesign Build Evaluate Iteration 3
Chapter 7
Conclude
Awareness of the Problem
Research Design
Design Science Design-Build-Evaluate cycles
Knowledge contribution & Further Research

Figure 3.5: Framework adaptation for this project

3.5.3 The Initial Stages

The research problem, semantic data integration was introduced in Chapter one and the tentative ideas developed as to how 4D foundational ontology and a graph database could be employed to address this problem. The literature was consulted relating to these subjects to provide knowledge and understanding of the previous research undertaken and the state-of-the-art within semantic data integration. This ensures the rigour necessary to confirm that the artefacts and methods that are the subject of the proposed research are in fact innovative and will thus, on completion of the research, through publication, aim to make a contribution to the corpora of knowledge related to ODISE: Ontology-Driven Information Systems Engineering.

The objective of the second stage of the research project is therefore to set the scene for the research by providing its context within the rise of the Semantic Web and Open Data. Then to gain an understanding of the previous research related to foundational ontologies, semantic data integration and state-of-the-art graph databases. The objectives of the second stage were fulfilled through a review of literature relating to these topics. The review also helped inform the decision making related to the selection of technologies that would be employed within the initial design, implementation and evaluation stage. The knowledge developed during the literature review formed the tentative ideas and plans were made concrete through the development of the initial 4D-SETL framework and its application and assessment that are undertaken during the course of the first iteration of the design-build-evaluate cycle.

3.5.4 The Core Design-Build-Evaluate Cycle

The Design-Build-Evaluate cycle is at the core of a design science project. Through the iterations of the cycle, artefacts are produced and improved until they demonstrate the utility necessary to achieve the research aim (Simon, 1996). Figure 3.6 depict the trend of increasing artefact quality and knowledge over the course of the three iterations. At the start of each iteration it is essential to provide an argument for the construction of the artefacts and, on completion of the design and build activities, to provide a thorough evaluation (Iivari, 2007).

The first, second and third (final) iteration of the design, build and evaluate cycle are described in detail within chapter four, though to six respectively.

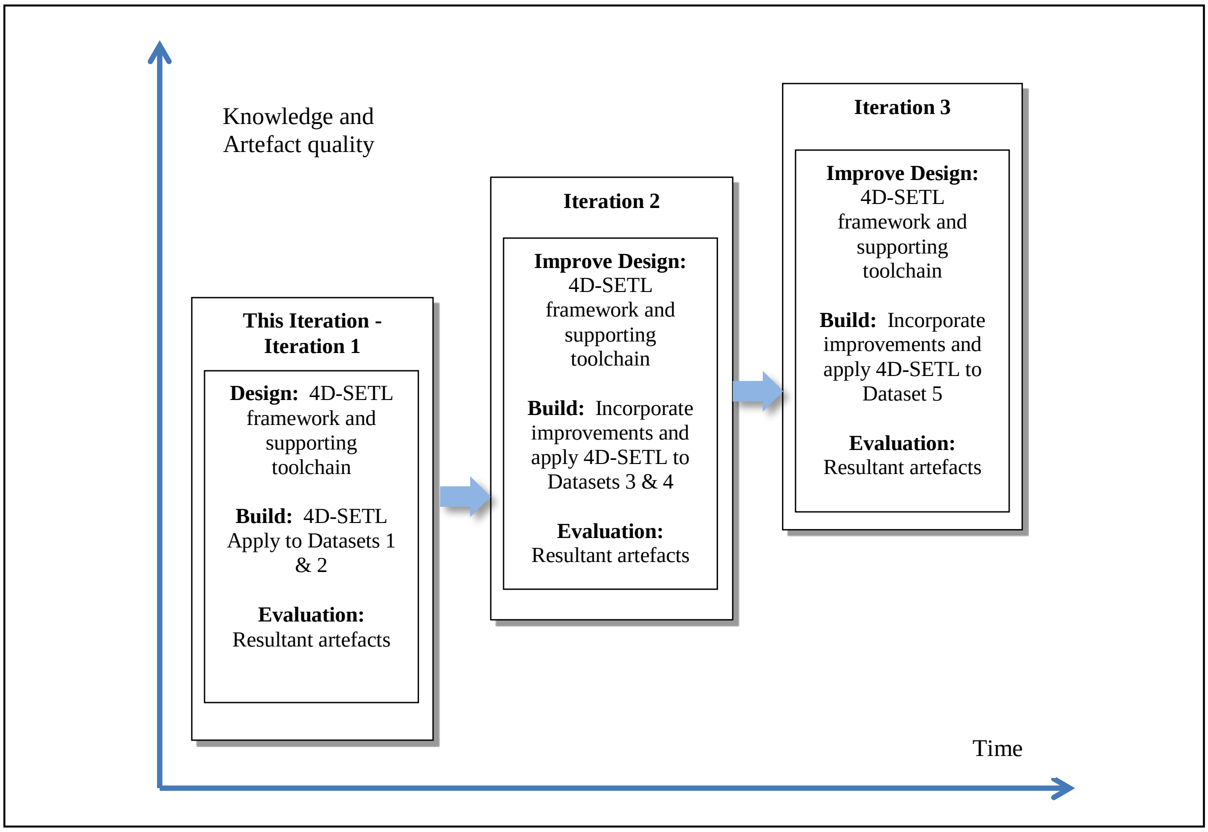

Knowledge and
Artefact quality
This Iteration -
Iteration 1
Design: 4D-SETL framework and supporting toolchain
Build: 4D-SETL Apply to Datasets 1 & 2
Evaluation: Resultant artefacts
Iteration 2
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Datasets 3 & 4
Evaluation: Resultant artefacts
Iteration 3
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Dataset 5
Evaluation: Resultant artefacts
Time

Figure 3.6: Research Design, Build, Evaluate Cycles

During the course of the first iteration, the initial 4D-SETL framework is designed and developed and a software system instantiated to support the framework and to realise the artefacts that are used to access the effectiveness of the framework. The 4D-SETL is further developed, improved and evaluated over the course of the subsequent iterations. The 4D-SETL framework adopts the processing stages of Extract-Transform-Load (ETL) found in many warehouse system implementations. The effectiveness of the system then assessed through the following prescribed design science evaluation methods.

Technical Experiment: The artefacts resulting from the execution of the design-build design science methodology include instantiated technical system artefacts such as prototype data warehouse populated with an ontological model. These artefacts will be subject to technical experiment to assess their qualities in terms of basic performance.

Illustrative Scenario: In addition to technical experiment illustrative scenarios will also be employed to describe the effectiveness of the artefact in relation to current information system design and implementation techniques.

The core design-build-evaluate cycle is repeated over the course of two further iterations through which the framework is improved and tested with a number of large scale datasets of differing complexity.

Ontology Evaluation:

A number of methodologies have been proposed to enable ontologies to be evaluated. Two of such methods are OntoClean (Guarino and Welty, 2002) and the method proposed by Gruber (1995). OntoClean provides guidance as to the selection of ontological elements with the aim of providing a developer with guidance as to the correct decisions that should be made relating to the selection of correct ontology model elements. The framework also provides developers with ant-patterns i.e. the error most often made by inexperienced developers. The meta-ontology described by Ontoclean was the basis of the DOLCE foundational ontology. Gruber’s evaluation criteria are based on the criteria of: clarity, coherence, extendibility minimal encoding bias and minimal ontological commitment.

As the research work of this described by this thesis employs a foundational ontology based on concepts that are to an extent incompatible with those espoused by DOLCE, the criteria described by Gruber will be employed to evaluate the domain ontology artefacts produced by the design science stages of this project.

3.5.5 Thesis Summary and Concluding Stage

The last stage of the research presents the conclusion of the project and an overall assessment of the project aim, objectives and artefacts produced during the course of the three design-build-evaluation iterations

3.6 Summary

As described within this chapter, the design science overarching method is adopted from Vaishnavi and Kuechler (2004) and therefore it is executed over the course of a number of stages that include: problem awareness, research design, design science design-build-evaluate (iterative cycle) and conclusion. The core design-build-evaluate cycle is executed over the course of three iterations. Each of the iterations is used to design, build and evaluate a set of artefacts aimed at the exploiting and evaluating the use of a perdurantist foundational ontology and graph database for semantic integration. This is achieved through the development of the 4D-SETL framework.

In the first iteration the 4D-SETL framework is developed and applied to integrate two experimental datasets and the results then evaluated. Through the course of the second and third iteration the methodology, tools and resultant semantic warehouse system are improved. The artefacts resulting from the design science are classified and evaluated according to the design science method adopted. The evaluation is based on the effectiveness of the method to semantically Extract, Transform and Load (ETL) a number of experimental datasets within a data warehouse system instantiated using a graph database. Evaluation is based on technical experiment and illustrative scenario. Table 3.4 provides an overview of the three iterations and the activities and evaluation methods employed.

3.6.1 Activity vs Output Overview

Iteration	Purpose	Activities	Output Artefacts	Evaluation
1	The aims of this iteration are firstly to develop the 4D-SETL framework. Secondly to apply the framework to integrate the first two datasets and to uncover the general form and structure of integration patterns assess the effectiveness of the solution.	Design: 4D-SETL framework.	4D-SETL framework.	The 4D-SETL framework is evaluated through the resultant artefacts via: a) Technical experiment. b) Illustrative scenario. c) Ontology evaluation
		Design: Prototype warehouse application and a tool chain to support 4D-SETL.	Prototype Data warehouse populated with the foundation ontology (Graph database). Tool-chain.	
		Apply 4D-SETL to establish the temporal and geographic ontologies (including instance level data load).	A data warehouse populated with the foundation and two domain ontologies including instance data.	
2	The aim is to improve the 4D-SETL framework and associated software and to further test the effectiveness of by integrating larger scale more complex datasets.	4D-SETL framework improvement.	Improved 4D-SETL framework.	
		Improve prototype warehouse application and 4D-SETL supporting tool chain.	Improved framework and software application (BORO UML – Graph DB translation).	
		Apply 4D-SETL to establish the SIC 2007 and Companies House ontologies (including instance level data load).	Warehouse application populated with foundation and four domain ontologies including instance data.	
3	The aim is to improve the 4D-SETL framework and associated software and to further test the effectiveness of by integrating a larger scale more complex dataset.	4D-SETL framework improvement	Improved 4D-SETL framework.	
		Build: 4D-SETL framework improvement	Improved framework, warehouse software application (Ontology Data Loader API).	
		Apply 4D-SETL Apply 4D-SETL to establish the Directors ontology (including instance level data load).	Warehouse application populated with foundation and five domain ontologies including instance data.	

Table 3.4: Research Activity vs Output Overview

3.6.2 Evaluation Overview

Table 3.5 provides details of the how the effectiveness of the framework is evaluated based on its ability to address identified weaknesses in the current approaches.

Weakness	Description	Artefact Effectiveness Evaluation
Lack of grounding	Many current models employed within information systems have no form of grounding in a more fundamental theory (Cregan, 2007). Thus the ontological commitments that underlie such models are unknown. On examination of many Linked Open Data ontologies, they are often ungrounded – or rely or reference other ungrounded ontologies.	**Artefacts:** 4D-SETL BORO Foundation, domain ontologies. **Evaluation:** Illustrative scenario. **Effectiveness:** The ability to semantically interpret and model each of the experimental data sets and the domains represented in a consistent manner that is grounded by the foundational ontology.
Model Strata and translations distortion	The strata of models formed by the software design process and the requirement to translate the high level models of reality that are created at the initial design time into a series of tabular statures that are focused of the execution environment can lose direct traceability to the initial model. This is analogues to the oft cited, within IS practice OO-RDBMS impedance mismatch (Ireland *et al.*, 2009).	**Artefacts:** 4D-SETL ETL of the BORO Foundation and instance level objects. **Evaluation:** Illustrative scenario. **Effectiveness:** The ability of the 4D-SETL to reflect the ontological models within a graph database with minimal translation.
Over simplification to fit a model of reality to a tractable theory	The need to simplify the abstraction of reality to it can fit neatly into a FOL theory, thus ignoring the fact that reality is not so simple and higher order objects exist (Bailey, 2011).	**Artefacts:** SIC 2007 domain ontology. **Evaluation:** Illustrative scenario. **Effectiveness**: Ability of the 4D-SETL to model and semantically integrate higher order elements such as taxonomic classification which is the basis of the SIC 2007 categories (an ontology consisting of only types and types of types – no instances).
Integrating models which are founded on different semantics	There are many automatic translation techniques for translating RDBMS schema and data to an OWL ‘ontology’. However, there is a lack of recognition of the semantic differences that underlie the differing modelling constructs.	**Artefacts:** Domain ontologies **Evaluation**: Illustrative scenario. **Effectiveness**: the ability to produce domain ontologies that meet the objective criteria proposed by Gruber (1995) for judging the quality of such models.
Dividing models into static and dynamic types	The separation of static and dynamic aspects of reality into different structural and process models leads to the development of incompatible abstractions together with exotic relations such as trans-ontological relations that are employed to bridge these static and dynamic worlds.	**Artefacts:** All five domain ontologies **Evaluation:** Illustrative scenario. **Effectiveness**: Ability to model structure and change (states) within the same model.

Table 3.5: Artefact Evaluation through Addressing Specific Weaknesses

Weakness	Description	Artefact Effectiveness Evaluation
Naming and meaning confusion	That there is often naming and meaning confusion, as described by Frege (1948). The objects place in reality (and the ontology) define its meaning.	**Artefacts:** Domain ontologies - Naming pattern. **Evaluation**: Illustrative scenario. **Effectiveness**: Providing clarity through the separation of names and the things the name into separate but related structures.
Establishing identify	Many modelling and information systems use ephemeral means of establishing an objects identity which do not function well over time.	**Artefacts:** Domain ontologies – states and happens-at, happens-to patterns. **Evaluation**: Illustrative scenario The effectiveness of 4D-SETL to cope with change over time: by in providing: i) a clear means of establishing and maintaining the identity of domain objects as their constituent spatiotemporal parts unfolded over time, (enabling process and static data to be combined within a single model), ii) Enabling new datasets to be added without recourse to amendment of the existing models or data. Evaluation: Illustrative scenario.
Employing techniques that do not scale	Many of the software tools such as OWL tableau calculus based reasoners, as they are constrained by memory, cannot scale to inference over ontologies containing large scale instance population (Bock *et al.*, 2008).	**Artefacts:** Geographic, Company and Officer Domain ontologies. Graph database queries (graph traversals) **Evaluation:** Technical Experiment. **Effectiveness**: The ability to support the loading of foundational and domain ontologies which contain a large number of individual elements (instance data) and the ability to perform effective data retrieval via graph database queries (graph traversals).

Table 3.5 Continued: Artefact Evaluation through Addressing Specific Weaknesses

Table 3.7 provides an outline of the evaluation criteria that are employed to assess the 4D-SETL domain ontologies following the final iteration of the design science cycle.

Criteria	Description
Clarity	Ontology should effectively communicate the intended meaning of defined terms. Ontologies are also not limited to definitions in the logic sense. To specify a conceptualization one needs to state axioms that do constrain the possible interpretations for the defined terms. Concepts should be independent of social or computational context. Formalism is a means to this end. When a definition can be stated in logical axioms, it should be. Where possible, a complete definition (a predicate defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). All definitions should be documented with natural language. Clarity entails ontological definitions that are context independent.

Table 3.7: Ontology Evaluation Criteria (Gruber, 1995)

Criteria	Description
Coherence	An ontology should be coherent: that is, it should sanction inferences that are consistent with the definitions. Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples.
Extendibility	It should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialize the ontology monotonically.
Minimal encoding bias	The ontology should be specified at the knowledge level without depending on a particular symbol-level encoding.
Minimal ontological commitment	An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities. An ontology should make as few claims as possible about the world being modelled,

Table 3.7 Continued: Ontology Evaluation Criteria (Gruber, 1995)

3.6.3 Overarching Integration Plan

The overall plan regarding the application of the 4D-SETL framework is to load the foundation ontology then to integrate each of the domain ontologies and their associated instances (datasets). The load order and ontology source is provided in Table 3.6.

Iteration	Load Order	Ontology	Ontology Source
1	1	Foundation Ontology	BORO UML Foundation Ontology Model
	2	Temporal Ontology	BORO UML Ontology Model
	3	Temporal Ontology Instances (Calendar)	Dataset programmatically generated
	4	Geographic Ontology	BORO UML Model (extracted from dataset)
	5	Geographic Ontology Instances	Postcode Dataset (2.5 M records)
2	6	Standard Industry Codes Ontology (Taxonomic Ranks)	BORO UML Model
	7	Standard Industry Codes Ontology (Taxonomic Classes)	SIC 2007 Dataset (of types and types of types)
	8	Companies House Ontology	BORO UML Model (extracted from dataset)
	9	Companies House Ontology Instances	Companies House Dataset (3.5 million)
3	10	Company Officers Ontology	BORO UML Model (extracted from dataset)
	11	Company Officers Ontology Instances	Directors Dataset (12.5 million records)

Table 3.6: Ontology Load Order and Source

The rationale behind this ordering of the integration of the different datasets is that the foundation ontology is common to all domain ontologies and the temporal and geographic (geo-spatial) ontology entities are common to both the Company and Company Officers datasets. Figure 4.2 depicts these relationships in diagrammatic form.

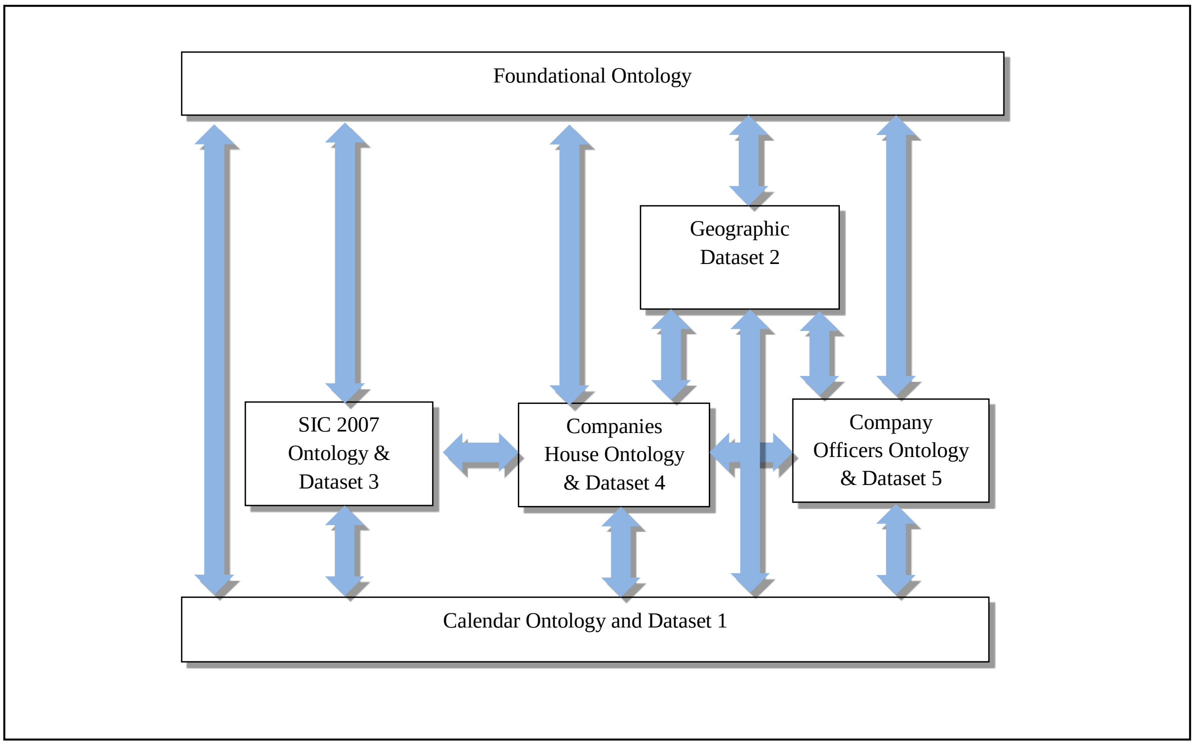

Foundational Ontology
Geographic Dataset 2
SIC 2007 Ontology & Dataset 3
Companies House Ontology & Dataset 4
Company Officers Ontology & Dataset 5
Calendar Ontology and Dataset 1

Figure 3.7: Dataset Relationship Overview

This ordering may benefit from further elaboration, therefore the following example is provided. The semantic data is held within the warehouse system in graph form. Therefore to add a domain ontology element requires that it relates to the elements of the foundation, for example to assert that the new domain object is a Type, requires that the relationship Types Instance be asserted between the new object to added to the warehouse and the foundation Types instance (the Type of Types). The same is true of instance level ontology objects; they must be integrated with the graph that forms the domain model. Therefore the load order detailed in table 3.5 has been established as the logical order to load and integrate each ontology.

CHAPTER 4. DESIGN SCIENCE FIRST ITERATION- THE INITIAL DEVELOPMENT AND APPLICATION OF THE 4D-SETL FRAMEWORK

4.1 Introduction

This chapter describes the execution of the first iteration of the research design-build-evaluate cycle. The aim of this iteration is to explore the exploitation of a 4D foundational ontology and a graph database to perform semantic data integration and to assess the *effectiveness* of the approach. This aim is realised through a number of objectives, the primary of which is the development of the 4D Semantic Extract Transform and Load (4D-SETL) framework which employs a 4D foundational ontology and a graph database to perform the integration. Following the design and development of 4D-SETL, the framework is applied to Extract, Transform and Load (ETL) two experimental datasets. The two main artefacts resulting from the first iteration are the 4D-SETL framework and a prototype data warehouse system populated with the foundation and two domain ontologies that include a large number of instance level elements. The effectiveness of the 4D-SETL, and thus the effectiveness of exploiting foundational perdurantist ontology and graph database for semantic integration, is evaluated via technical experiment and illustrative scenario.

This chapter is structured as follows. The aim and objectives are outlined in the next section (4.2) which also provides details of the research design for the design science iteration. This consists of a list of the design-build-evaluation related activities, the research artefacts produced and the experimental datasets employed. Section 4.3 provides details and design rationale for the 4D-SETL framework and explains the stages involved. The following two sections (4.4 and 4.5) describe the first and second application of 4D-SETL framework, respectively; and how the 4D-SETL is employed to semantically integrate the two experimental datasets. Section 4.6 provides an overview of the prototype system instantiation and the tool-chain that is developed during the course of the design science iteration to support the framework. Section 4.7 details the outcomes of the iteration through the evaluation of 4D-SETL assesses the effectiveness of the use of 4D (perdurantist) ontology to semantically integrate data. Finally section 4.8 discusses the learning and the deficiencies that will be addressed in the subsequent research iteration.

4.2 Design Science Research - Iteration One

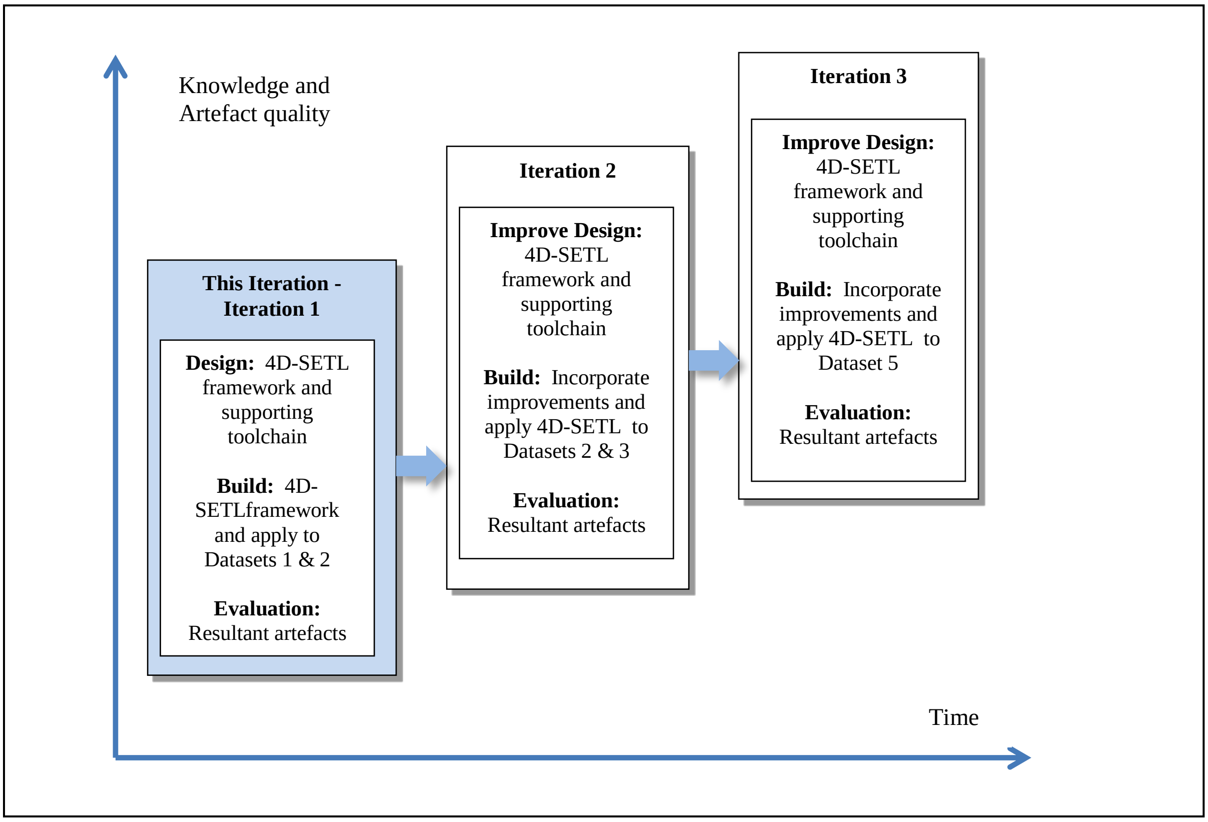

Knowledge and
Artefact quality
This Iteration -
Iteration 1
Design: 4D-SETL framework and supporting toolchain
Build: 4D-SETLframework and apply to Datasets 1 & 2
Evaluation: Resultant artefacts
Iteration 2
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Datasets 2 & 3
Evaluation: Resultant artefacts
Iteration 3
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Dataset 5
Evaluation: Resultant artefacts
Time

Figure 4.1: Research Implementation Cycle Iteration One

The activities within this iteration serve to explore the research problem and are planned and executed in accordance with adopted design science research methodology. A diagrammatic overview of the first iteration and how it relates to the subsequent iterations is provided in Figure 4.1.

4.2.1 Aim and Objectives of Iteration One

The objectives of this first design science iteration are realised through activities related to the design-build-evaluate of the 4D-SETL framework. These objectives serve the main aim of the research which is to evaluate the *effeteness* of exploiting both a 4D ontology and a graph database to facilitate semantic integration. The stage, activity reference and objectives are listed in Table 4.1.

Stage	Activity	Objective	Section Reference
Design	1	Design the 4D-SETL framework. Document the ontological architecture including the foundation ontic categories and foundational patterns that will be employed within 4D-SETL	4.3
	2	Design the tool chain to support the framework	4.4
Build	3	The First Application of 4D-SETL – Temporal Ontology	4.5
	4	The Second Application of 4D-SETL – Geographic Ontology	4.6
Evaluation:	5	a) Technical experiment heuristic testing of the warehouse artefacts b) Illustrative scenario: describing the artefact qualities and advantages.	4.7

Table 4.1: Stages and Activities and Within this Iteration

4.2.2 Experimental Data Sources for Iteration One

The first iteration utilises two experimental data sources that were chosen as many of their instance elements are common to the datasets integrated in subsequent iterations. The first experimental dataset is composed of temporal locations in the form of a calendar; the second experimental dataset contains geo-spatial locations in the form of a postcodes and geographic co-ordinates. The geographic dataset is in the form of tabular instance level ‘data’ and descriptive natural language ‘model’ of each of the data source elements.

4.3 The Design of the 4D-SETL framework

This section describes the development of the 4D-SETL framework and its first application. The requirements of this first iteration were drawn from the initial research idea and although the desired result was known at the outset, the methods and tools that would be employed to implement the 4D-SETL framework and how it could be applied were largely unknown. Therefore, the first iteration of design-build-evaluation cycle was initiated without a clear view as to the tools and techniques that would be appropriate to use. Therefore, design decisions are based on creative inspiration and experience rather than grounded theories, which is an acceptable form of design when researching ‘the cutting-edge’ of the domain, (Hevner *et al.* (2004).

The 4D-SETL framework builds on the work of Partridge (2005) and adopts the foundational Business Objects Reference Ontology (BORO), a perdurantist (4D) ontology together with the REV-ENG reverse engineering technique (Partridge, 2005). REV-ENG was previously employed to semantically interpret and recover existent datasets from legacy systems. The BORO foundation provides the semantics that are common to all domain and instance objects. REV-ENG provides the means of interpreting the elements of a dataset in terms of the foundation and therefore of developing domain ontologies. 4D-SETL extends the methods and ontology developed by Partridge (2005) to provide the means of semantically integrating data within a graph database. The approach aims to produce semantically rich and highly expressive ontologies in contrast to those that only adopt Description Logics (DL) based First Order Logical (FOL) theories (e.g. Semantic Web ontologies). This choice is motivated by the research aim which is to exploit a 4D ontology that is grounded in a perdurantist, realist view of reality with a clear way of defining identity that may prove more suitable for modelling and integrating data related to a range of domains. Liberated from the expressivity constraints imposed, by for example FOL, the designer is free to model the world in a more accurate manner by replicating structures found in reality such as higher order objects and how the states of such objects unfold over time. Therefore, rather than employing inference services based on general purpose reasoners (inference engines), the resultant system is targeted towards the use of declarative, imperative and functional programming techniques for the purposes of processing data and querying the warehouse.

4.3.1 Applying the 4D-SETL framework

The process by which a perdurantist foundational ontology can drive semantic integration is described in this section.

It is theorised that 4D ontologies will prove an appropriate solution to the problem of semantic data integration as they provide an unambiguous view of identity over time. The particular foundational ontology selected (BORO) also provides the facility to differentiate between the model representation of objects of reality and the model symbols that name such objects. The separation of names from the objects that they name has many advantages, for example it can diminish the confusion between reference and referent that can occur. Furthermore, it provides a clear distinction between language and physical structure. For example, the city of Liverpool is part of the UK, the name ‘Liverpool’ is not part of the name ‘UK’. Therefore, it is the two geo-political land areas and the whole-part relationship of the objects within the ontological model that defines the meaning, rather than each of the objects names.

Rather than being design time artefacts, within the 4D-SETL framework ontologies are employed as integral components of the executable application providing the structure and coherence necessary for both integration and the long term persistence of the ontologies (foundation, domain and instance level) within a data warehouse instantiation that employs a graph database, a high performance, and scalable emerging software technology. As the persistence method employs a graph database, new instances, classes and relationships can be added to the ontology at run time. This is in contrast to Object Oriented software or RDBMS systems that are constrained by (normally) lacking the ability to allow schema or class updates at run time. Therefore, when loading ontologies to the warehouse, such ontologies can represent types, tuple objects (relationships) or instance level objects. This ability will be demonstrated within the second iteration, described within chapter five, when a data source representing the Standard Industry Classification system that consists of taxonomic ranks and classes is Extracted, Transformed and Loaded to the warehouse.

As previously discussed in Chapter two, and as stated by Noy (2004), that in order to achieve semantic integration it is necessary to study the available evidence to discover the semantics of a dataset to be integrated and then to establish the canonical semantics through which integration can be undertaken. The 4D-SETL framework fulfils this criterion by collecting together information such as schema and background information within a worksheet. BORO and REV-ENG fulfils the second criterion by providing the grounded domain independent system of categories together with a means of consistently interpreting the elements of the dataset and its schema. The patterns drawn from BORO are also used to instantiate the coherent model structures within the prototype warehouse (graph database) system and are reflected to form ontic categories (meta-meta-model), ontology patterns (meta-model) and instance level objects (model) artefacts. Figure 4.2 depicts a simplified view of the process of translating the semantic structures of tabular data to a graph based ontological representation. The table columns are converted to Type nodes and the contents of a record are converted to nodes that represent member instances. Within this example, there are also a number of relationships that are asserted between instance nodes to establish directorship and location relationships. These mirror the relationships patterns established at the Type level. These relationships are asserted with tuple instances that consist of a node and two (or more) place edges. Relationships (tuples) are first class objects and can are therefore be related to type (tuple-type) level objects. The instance level relationships reflect the type level relationships of the higher domain ontology level and are related by being instance, type members of the domain tuple type (this relationship is not shown in the figure).

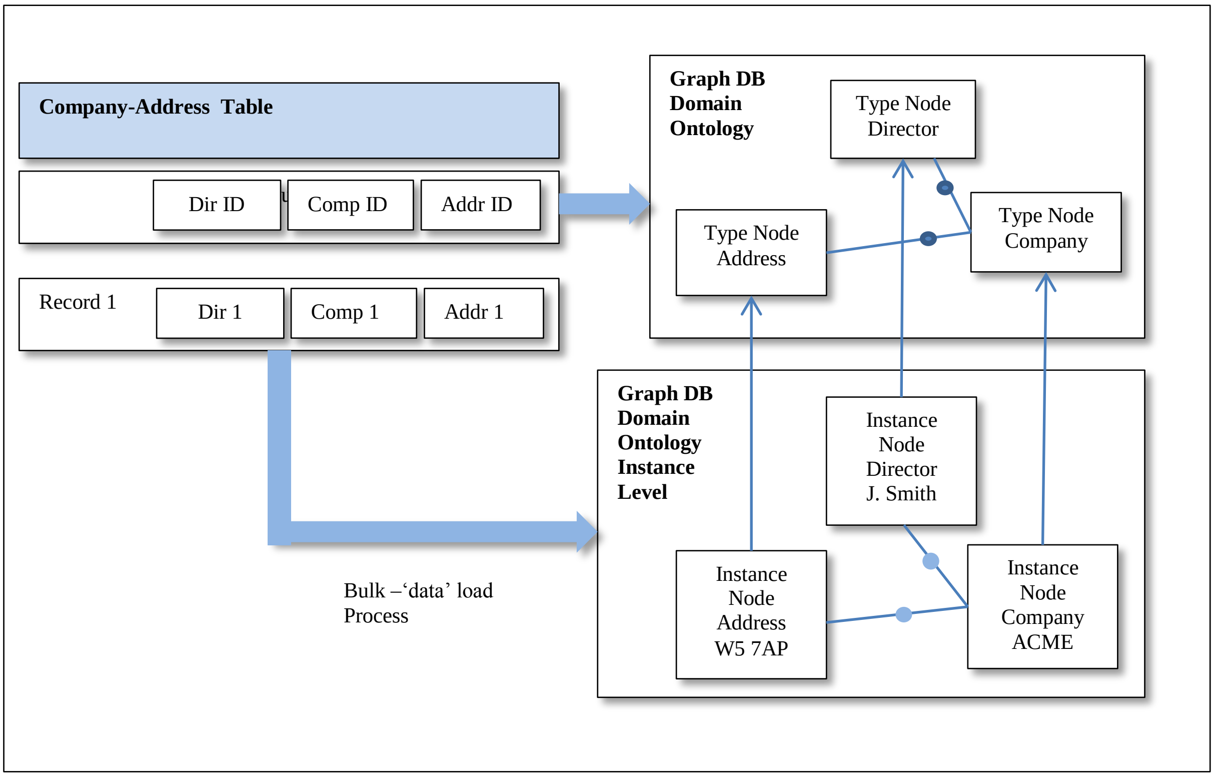

Company-Address Table
Dir ID
Comp ID
Addr ID
Record 1
Dir 1
Comp 1
Addr 1
Bulk –'data' load Process
Graph DB Domain Ontology
Type Node Director
Type Node Address
Type Node Company
Graph DB Domain Ontology Instance Level
Instance Node Director J. Smith
Instance Node Address W5 7AP
Instance Node Company ACME

Figure 4.2: Overview 4D-SETL Extract Translate and Load Process

4.3.2 4D-SETL Design Time Ontology Representation

The first stage of the 4D-SETL framework is to examine and analyse the datasets that will be semantically integrated and to document the elements that will form the domain ontological models within a worksheet (spreadsheet). Therefore, a record is created for each element and relationship in terms of the dataset, 3D Semantics and finally 4D semantics. Once this work has been completed, a domain ontology is developed based on this information. The knowledge representation language chosen for developing ontologies is the BORO profile of Unified Modelling Language (UML). Thus, industry standard tools are initially employed to develop the domain ontologies. However, the model semantics differ from that as usually employed within a software engineering context. Thus, to effect semantic integration, the domain specific objects and relationships are developed that conform to the foundation. This process has been described by Rector (2003) as semantic normalisation. However, the terminology adopted in this framework for this process is extraction and transformation.

4.3.3 Run Time Ontology

As depicted in Figure 4.3 through the 4D-SETL framework, the design time BORO UML ontological models are transformed to become an integral part of a the graph database system that can be utilised as an integral component of an information system that can be queried via a graph traversal and that can be undated through ACID compliant transactions. The ontology is represented as a set of graph nodes and edges within a graph database.

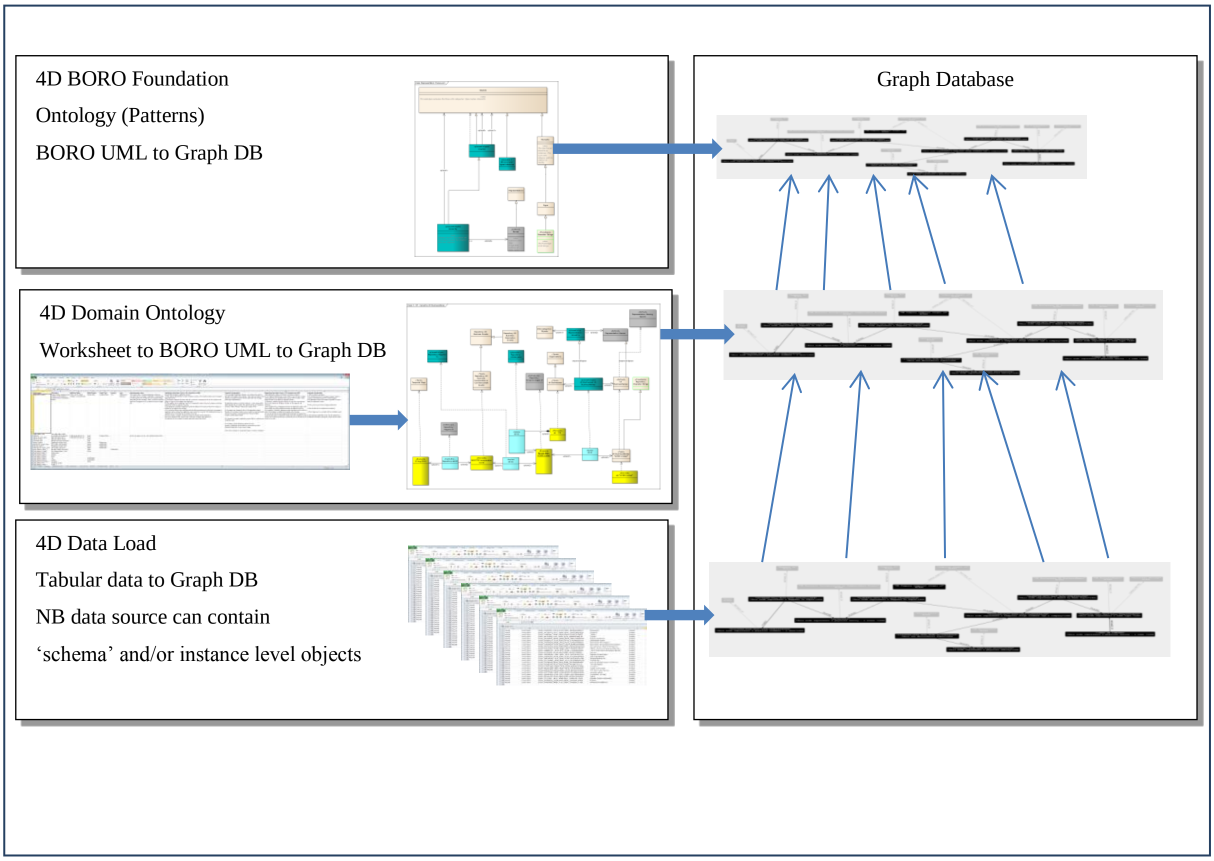


Figure 4.3: Ontological Architecture

4.3.4 4D Foundational Ontic Categories

As described in Chapter two, Section 2.7, at the highest level of the ontological architecture, the Business Object Reference Ontology (BORO) Foundation provides the fundamental ontic categories of Elements, Types and tuples.

4.3.5 4D Foundation Patterns

Tuple Types and tuple Type instances are employed to establish the general relationship patterns which are defined by BORO. The Table 2.7 provides a non-exhaustive list of a number of the main patterns.

During the course of the framework and prototype instantiation, and its application, the patterns used or discovered are documented as they will form part of the research output.

4.3.6 Ontology Design Patterns

Developing ontologies based on patterns, together with a recognised framework, can help improve the quality of the overall results. Fowler (2002) states, in relation to software design, that patterns capture the essence of good designs and suggests that they are discovered through observations of what is happening in practice, rather than being the subject of a good idea. In a similar way to software design, rather than starting from a blank sheet of paper, a domain level ontology design can be constructed from the design patterns drawn from a pattern library – in this case the BORO formational ontology.. Table 4.2 provides an outline of the phases in the process.

Phases	Description
1	Gather background and authoritative information related to the data which will be used as the basis for the ontology.
2	Analyse the concept to decide what each field represents in reality (3D) view
3	Convert to a 4D view (add temporal extent to 3D object)
4	Analyse what the data represents using the Rev-Eng process (see Figure4.7) – Element, Type or Tuple.
5	Analyse what pattern to apply (see Figure 4.8) such as: A Name that names a state i.e. the state of being President of the USA. An l Element such as Barak Obama – that has constituent states A physical location –relating an Element via a Located-at relationship (tuple) - Events that create and dissolve states and their temporal location happens-at/happens-to. An extensive description of these patterns is described by partridge (2005) .

Table 4.2: Ontology Design Phases

The ontology design process involves the analysis of the domain (universe of discourse) and the selection, adaption and combination of the relevant patterns drawn from the foundation. The following provides a non-exhaustive list of the patterns available from within BORO

BORO Name Pattern

The *Name* pattern is core to the BORO foundation and will be elaborated within the following paragraphs. The BORO (4D) foundational ontology considers ‘names’ to be a record of an utterance event (Partridge, 2005), which is more complete that that employed in the majority of information systems and adopts the philosophical stance expounded by Strawson (1964a):

> “The distinction between identifying reference and uniquely existential assertion is something quite undeniable. The sense in which the existence of something answering to a definite description used for the purpose of identifying reference, and its distinguishability by an audience from anything else, is presupposed and not asserted in an utterance containing such an expression, so used, stands absolutely firm, whether or not one opts for the view that radical failure of the presupposition would deprive the statement of a truth-value. It remains a decisive objection to the theory of Descriptions … that … it amounts to a denial of these undeniable distinctions” (Strawson, 1964a, p.85).

Commenting on Strawson’s (1964a) view, Snowdon (2009) asserts that there is no choice other than to apply common-sense to the nature of meaning and reference and that such fundamental concepts do not require validation by the empiricists, or those of science. Thus an instance of a character string (name or sign) encoded in electronic format within a dataset is deemed to represent a record of an ‘utterance’ event. Furthermore, an utterance event will exist in both time and space and therefore is an element.

The components of the BORO name pattern will be outlined in the following paragraphs.

- **A Name as a Defined Type:** The foundation objects that represent a '*Name'* (from the real or possible world) are a '*Defined Type'.* As by definition, a *'Name'* can only be a '*Name'* if it employed to name something. Therefore '*Names'* are defined by their function and consequently are considered a '*Defined Type'* (Partridge, 2005).

- **Name as a Type:** As within reality there can be, and usually are, many instances of a '*Name'*, for example a name such as 'ACME Limited' will exist in many paper based and electronic systems, BORO models a *'Name'* not as an individual *Element* but as a '*Type'* (or class) of *'Names',* which contains as member utterances of that particular name. Therefore, *Names* are modelled as a sub-type of *Elements Powertype* and *Representations*. The object *'Names'* within the foundation ontology is a second order object (a *Type* that contains *Types*). As previously described, each *'Name'* is represented by a *Type* (rather than an individual) that contains all the individual character *Strings* (sign objects) that are employed to *'Name'* an object (Partridge, 2005).

- **A Name as a Physical Object:** '*Names'* are related to the object they name by the '*named-by'* tuple (relationship) which establishes this relationship. By adopting BORO, an ontological commitment is made to the assertion that an instance '*Name String'* is a physical object in the form of spatio-temporal extent (*Element*). A '*Name String'* can exist in many forms: printed paper; the patterns formed by magnetic poles on a storage disk; electrical charges within a memory or processor chip; or an utterance in the form of sound waves (Partridge, 2005).

- **Name Spaces:** Each '*Name'* Type can also be a member of a '*Name Space'* instance which is controlled by a naming authority that has responsibility for naming objects, such as The UK Post Office (postcode areas) Companies House (company registration numbers) which provides unique identifiers. The '*Name Spaces Type'* is a third order '*Type'* that contains '*Name Space Types* that in turn contain '*Name Types'*(Partridge, 2005).

- **Names Model and Reality:** Within the BORO foundational ontology, names are as much objects as the objects they refer to in reality. Consequently, '*Names'* are model objects which are members of both the model and reality (domain), partly outside the object model and partly inside. Therefore, reality (the domain) and the ontology (the object model) are not distinct; rather they overlap (Partridge, 2005).

Figure 4.5 depicts a BORO UML model of the name pattern, which is established by a foundation pattern. This pattern is extended to the domain for the specific purposes of modelling the Company Record Office (*UK CROs*) reference numbers (names), a sub-type of *Names* and also an instance of a *Naming Spaces* that name *UK Businesses*. A relationship established though the tuple subtype type *UK Business UK CROs*

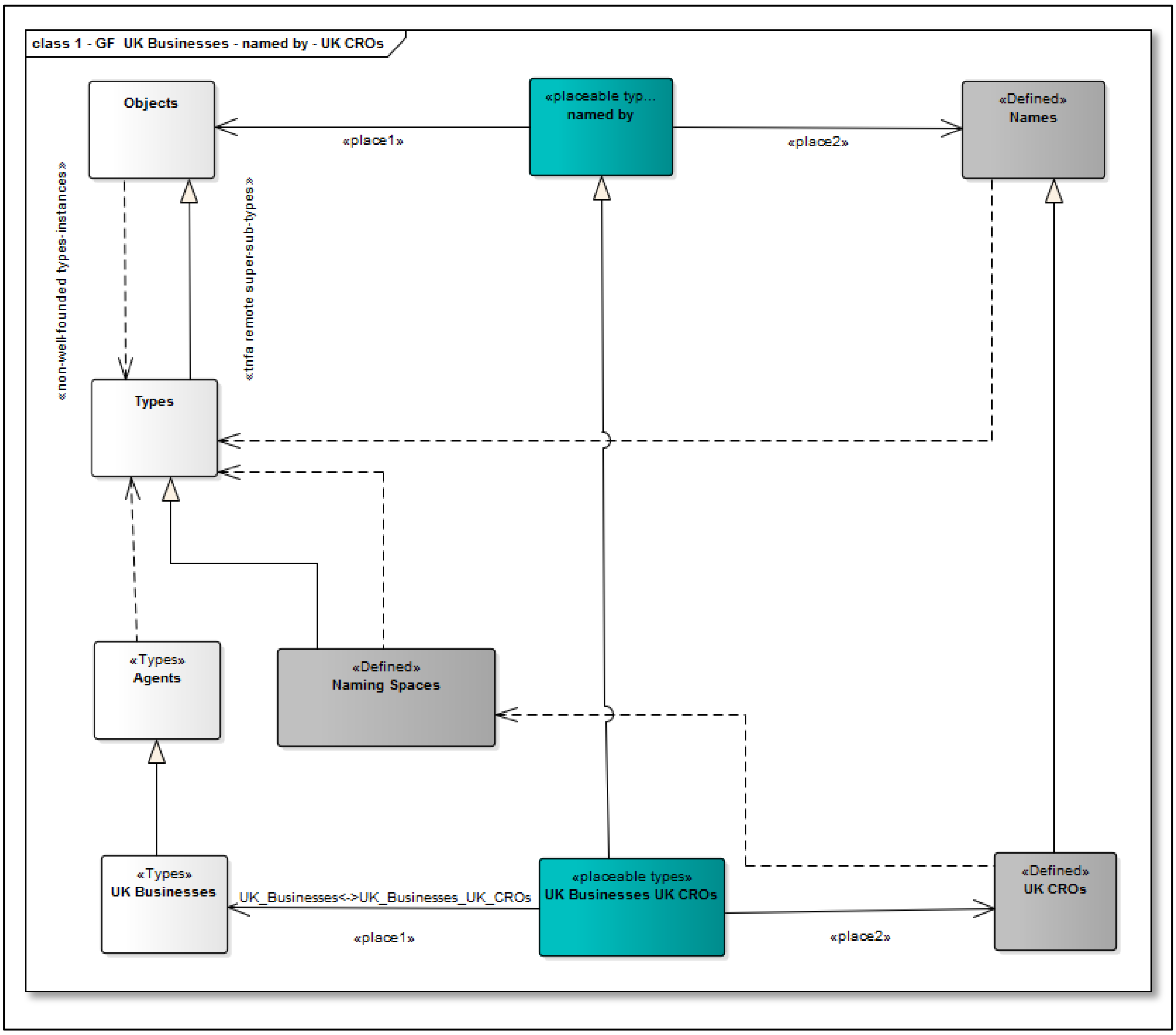

class 1 - GF UK Businesses - named by - UK CROs
Objects
«placeable typ... named by
«Defined» Names
«place1»
«place2»
« non-well-founded types-instances »
«tnfa remote super-sub-types»
Types
«Types» Agents
«Defined» Naming Spaces
«Types» UK Businesses
UK_Businesses<->UK_Businesses_UK_CROs
«placeable types» UK Businesses UK CROs
«Defined» UK CROs
«place1»
«place2»

Figure 4.4: Name Pattern (Partridge, 2005).

Exemplar Names – Integration Points

There can be any number of utterances that can be used to name an objects therefore, as previously described; a *'Name'* is a Type (class) of strings that can exist in a number of physical forms, including that of an electronic record. For the purposes of 4D-SETL, the time of the utterance is the time of the publication of the dataset. In addition, the authority of the publisher must be considered, for example the publisher may be a naming authority (or closely related). These features are employed during the 4D-SETL analysis and integration process to ascertain if a character string from a data source can be designated as being an

‘*Exemplar Name’*. If this is true, the name can be employed as an integration point –and datasets can be joined through this name. It should be noted that it is the elements that are named that are integrated rather than the names.

Model Object Names: Within the 4D-SETL model, object names are employed that are unique for each model element (node or relationship edge). The foundation and domain ontology object names that are generated by the ontology UML design tool have their own unique names. These names are reused within the warehouse (following the enhancement implemented for the second iteration). When an object is added to the warehouse that is programmatically generated by the load process, a unique name (sixteen hexadecimal characters) is generated. This ensures consistency as references within the graph database are independent of the node and edge index scheme of the graph database (Neo4J). This is important as both node and edge identifiers can be reused when an item is deleted. Human readable labels are also included in the model to help with debugging and query development.

Temporal Location Pattern

As a 4D spatio-temporal extent - extends in both time and space, it will have both temporal and spatial coordinates. In terms of a temporal location, it will be contained, for example, within the year 2015. As the foundational ontology asserts that the year 2015 represents all of space during the 12 month period, a spatio-temporal extent may be located within this period, or overlap with it. The location *happens-in* pattern has an associated creation and dissolution events. These events are 3D as they have a zero temporal extension; however they are physical objects that are also part of a slice of space-time.

Physical Location Pattern

A spatio-temporal extent may have some form of coordinates to specify its (relative) physical location. The *located-at* pattern provides the facility to associate s temporal part of the spatio-temporal extent at such a location.

Temporal Part Role Pattern

A spatio-temporal such as a company or person will be constituted from temporal parts. The role of being a company director is a temporal part of both the company and the person who is fulfilling the director role. In a similar manner, a company may have a business activity of mobile phone manufacturing; therefore, this state will be part of the Type defined by the Standard Industrial Classification system that has all such companies as members.

4.3.7 Semantic Data Integration Stages

The 4D-SETL semantic data integration processes is conducted over the course of three stages: Extract, Transform, and Load (to a warehouse graph database). These stages are applied to each of the datasets to be integrated in turn. Figure 4.5 provides an overview of the process.

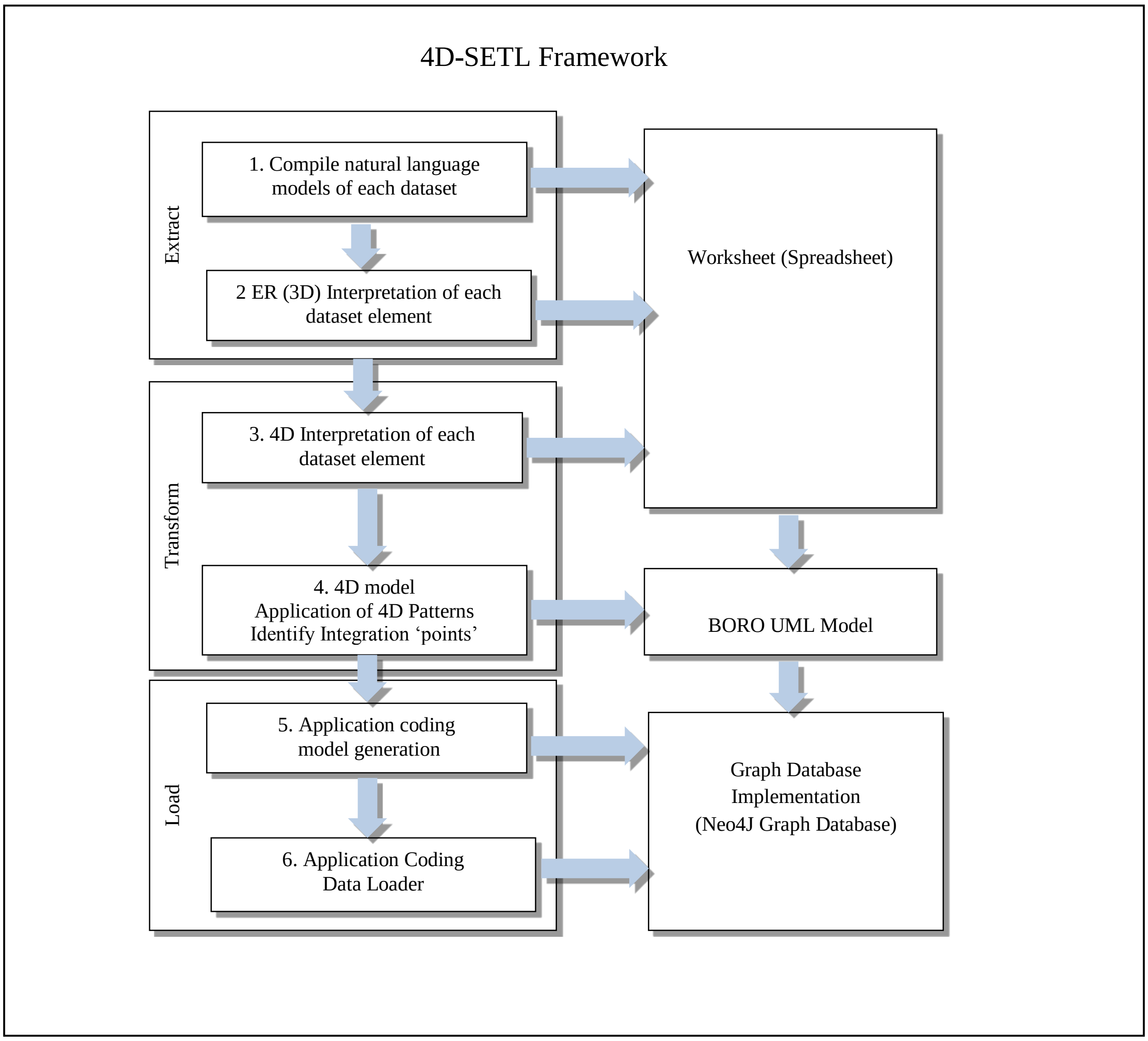

4D-SETL Framework
Extract
1. Compile natural language models of each dataset
2 ER (3D) Interpretation of each dataset element
Transform
3. 4D Interpretation of each dataset element
4. 4D model
Application of 4D Patterns
Identify Integration 'points'
Load
5. Application coding model generation
6. Application Coding Data Loader
Worksheet (Spreadsheet)
BORO UML Model
Graph Database
Implementation
(Neo4J Graph Database)

Figure 4.5: The Initial 4D-SETL Framework Stages and the Resultant Artefacts

First Stage: Extraction

The input data to the integration process may be structured in many forms such as RDBMS export, a delimited tabular file or an ontology. To overcome the difficulties related to integrating models that employ different structures, Kappel *et al*. (2006) describe a process of 'lifting' which transforms a model, such as the row and column structures found within tabular data, into an ontology. This is similar in nature to the process described by William *et* *al*. (1996) who introduced a process through which a conceptual schema could be generated from a relational schema.

Thus, the first stage in the 4D-SETL integration process begins with ‘lifting’ the source data so that they conform to an ontological representation in terms of structure. This can be considered a semantic extraction process which interprets the a) the instance level data and to what objects in reality they relate, b) the model to which such data conforms and c) other evidence available such as background information. The end result of this process is a domain ontology that conforms, to the best of the designer’s knowledge, with to the original 3D paradigm that was used to create the model.

The Office of National Statistics Postcode Directory (ONSPD) experimental datasets employed for the purposes of this research design cycle are sourced from government and other organisations in delimited, Comma-Separated Values (CSV) format. Normally such files are the result of some form of export process from a relational database system. Whilst the data is syntactically structured (in terms of rows and columns), the semantics of such datasets are implicit and therefore cannot be readily integrated with other datasets.

Within the Semantic Web field there has been much work related to the subject of translating ER Models to ontologies. However, as described by Sheth and Larson (1990), in relation to RDBMS integration, that typically the schemas alone contain insufficient semantics to enable the information to be consistently interpreted. Therefore 4D-SETL also references any background information that can be found relating to the data that is the subject of the integration.

However, from the tabular data there are various ways the original semantics can be recovered. The rules defined by Chen (1976) can be used for interpreting tabular Entity Relational (ER) in natural language form:

a) A Table: Common Noun – Entity Types

   Therefore, the Table Name can be interpreted as the Name of the Entity Type. Types are typically represented by a table. With more complex datasets, there may be a need to analyse several tables and the relationships between them. However, in the case of the experimental datasets which are employed within this project only single tables are available, and therefore the challenge is to semantically interpret and integrate this information.

b) A Row: Proper Noun – Individual

   Normally this is implicit – the index of the table.

   Each individual entity within the dataset is represented by a row (also described as a tuple or record) within a table.

c) A Column: Transitive Verb - Attribute Types:

   Therefore, the column Name can be interpreted as the Attribute Type Name.

   Attribute types are structured as columns within the tabular model.

d) A Field: Intransitive Verb – Attributes:

   Individual attributes are structured as fields within the tabular model and can be inferred to be the Name of an individual attribute.

The next step in the 4D-SETL framework is to detail what the tabular model represents (stands in for) in the real world. This is usually quite straight forward at this stage; it is either the name of something or a type of scalar value. For example, an address table with an attribute column called Postcode and a field containing the string ‘W13 9NP’. We have an

Entity Type Address, a Transitive Verb Postcode and an Intransitive attribute ‘W13 9NP’ - a Postcode instance.

From this can be inferred that there is an object – W13 9NP which is physical location and there is an object string ‘W13 9PY’ that names the physical object.

Second Stage: Transformation 3D Real World to 4D Real World Semantics

The initial semantic extraction produced a domain ontology that employed the same semantics as used during the original schema was created, i.e. Entities Types, Attributes. This is transformed to a perdurantist (4D) representation based on BORO - Types, Elements, tuples and Power Types etc. The BORO foundation provides a philosophically grounded view of reality, the patterns that can be employed to represent the ontology and a framework for its development. For example, translating an ‘Entity’ from a three dimensional extension to a four dimensional spatiotemporal extent (this process is described in detail later in this chapter during the application framework).

Therefore, the foundational ontology provides the equivalent of a canonical data model that can be employed to develop more specific domain models (Saltor, Castellanos and Garcia-Solaco, 1991) and provides the semantics that are common to all data sets that will be integrated.

Within this stage, the dataset referents are re-interpreted under the 4D (Object) Paradigm and the conceptual patterns for the entities are translated into 4D objects. For example, the physical object referenced by the string ‘W13 9NP’ is translated to a 4D object. This is achieved by converting the three-dimensional physical extension to four-dimensional extension - one that has both physical and temporal extension.

Thus, the ‘converted’ object is now considered to have both temporal and spatial extension and can be composed of other spatiotemporal parts (states). For example the surface area of the Earth (temporal extension, past plus future which is approximately 8 Billion years) named W13 9PY, now has a named state, a socially constructed state which has lasted approximately fifty years. Obviously, the underlying age of this particular surface area is of no particular use in the model, it is however useful to the conceptual realisation that what we are naming in our 4D ontology is an object that is extended in three spatial and one temporal extension. The next step is to identify which foundational category an object belongs. Figure 4.6 depicts the BORO classification process in flow chart form.

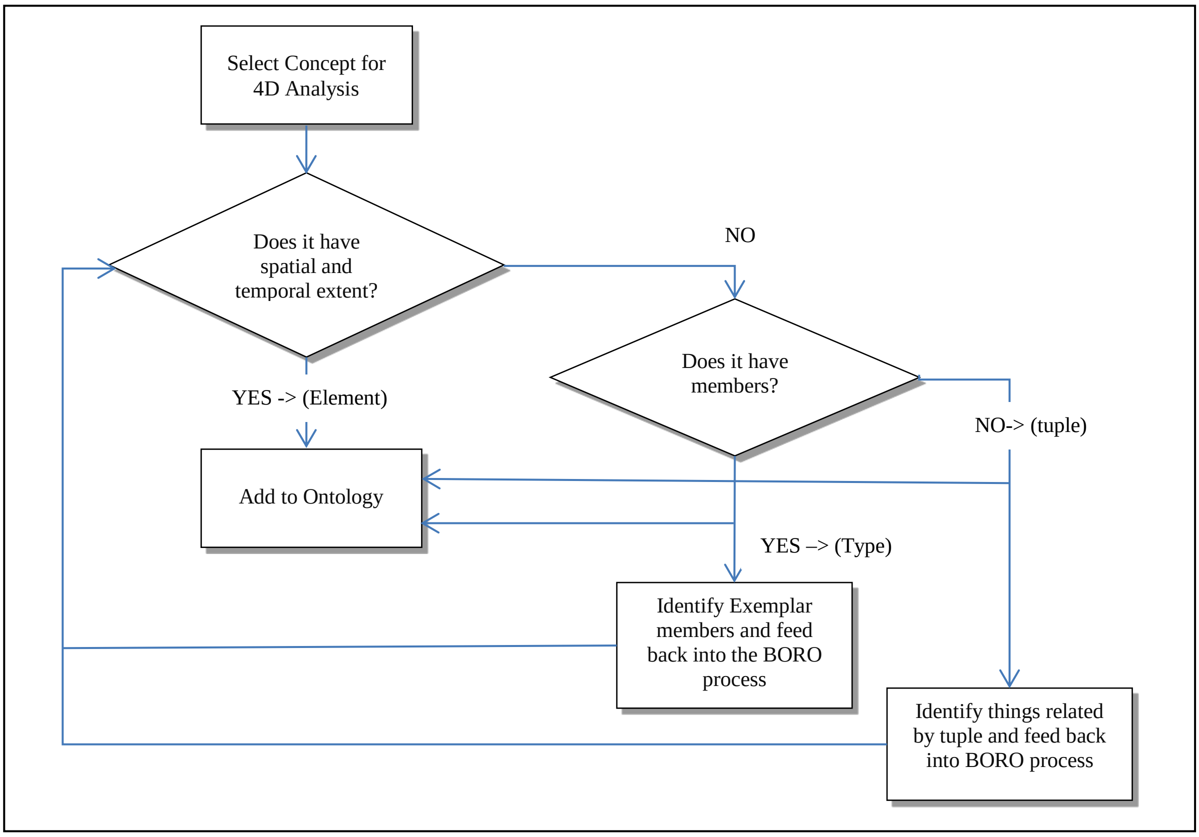

Select Concept for 4D Analysis
Does it have spatial and temporal extent?
NO
Does it have members?
YES -> (Element)
NO-> (tuple)
Add to Ontology
YES –> (Type)
Identify Exemplar members and feed back into the BORO process
Identify things related by tuple and feed back into BORO process

Figure 4.6: BORO REV-ENG Process (Partridge, 2005)

Each of the foundational ontological categories has clear criteria of identity which are: Elements, Types or tuples. Elements are identified by their 4D extension which exists in both space and time. Elements can also be equated to individuals i.e. objects that do not have member instances. Types are identified by their member instances and tuples are identified by the objects in each of the tuple places. In the example being described, the physical area W13 9NP persists through time and is therefore classified as an Element. Therefore, the object W13 9NP physical area has been classified and added to the ontology as a physical Element. However, the process is not complete; there is still the matter of the string ‘W13 NP’ which is a Name (string), as a name utterance is considered by BORO to have temporal and spatial extension it is also classified using REV-ENG as an Element. To clarify, take the name ‘ACME Limited’ – written on paper, engraved on a brass plate at the company’s registered office, or the characters encoded as patterns of magnetised metallic particles on the surface of a hard disk drive, all these instances would exist in space-time as 4D Elements.

The next step in the process positions the objects that have been translated to 4D within the overall ontological model. This is achieved by identifying a pattern that is applicable and establishing the relationships with the foundation and other domain level ontology objects based on a particular pattern. There are a number of foundational 4D patterns which can be identified and extended for use within the domain model. These include naming and structural patterns such as spatial and temporal mereological (whole-part) relations.

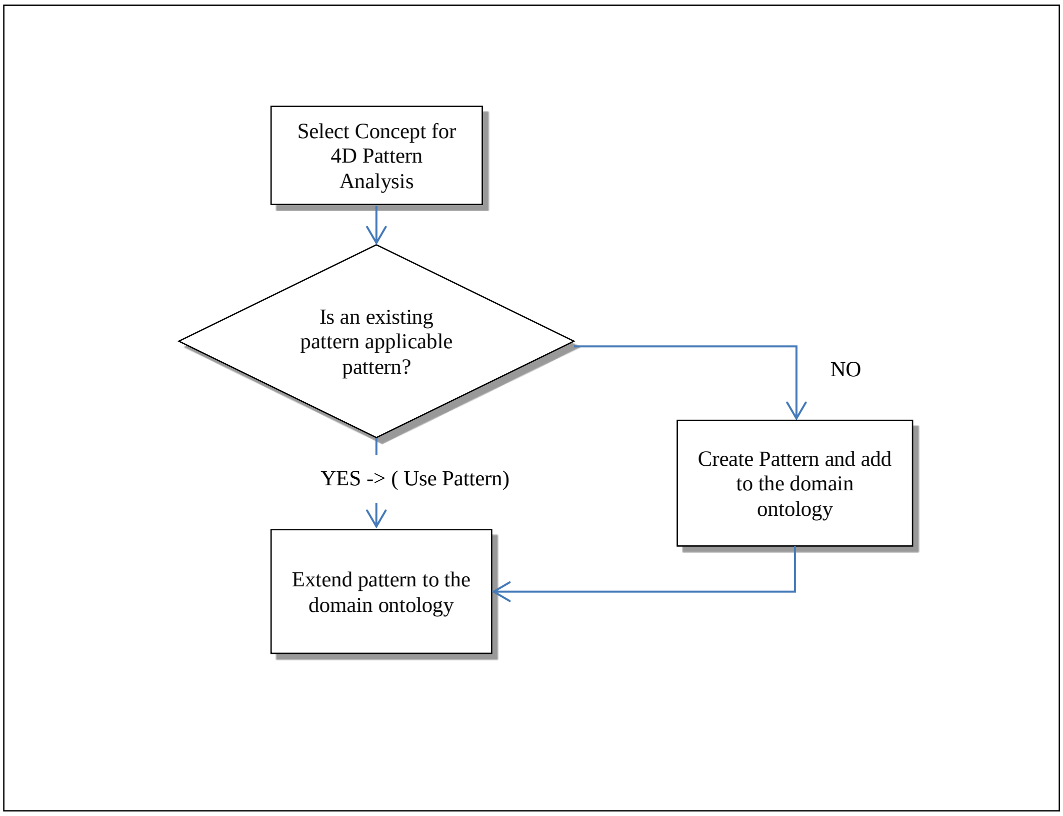

Select Concept for 4D Pattern Analysis
Is an existing pattern applicable pattern?
NO
Create Pattern and add to the domain ontology
YES -> (Use Pattern)
Extend pattern to the domain ontology

Figure 4.7: 4D Pattern Identification

The foundational patterns to apply in the example case are that of Type membership, asserting that a physical location is a member instance of the Elements Type and the string ‘W13 9NP’ is an element that is a component of the Name Pattern.

Following the documentation of all the dataset objects, a domain ontology model is developed using the Enterprise Architect design tool (Sparx Systems Pty Ltd., 2015). The ontology knowledge representation employs the BORO UML profile. Ontology development is achieved by extending and specialising the patterns of the foundation. Through this process, domain ontologies are developed to represent the entities and relationships that are represented by the data set. Within 4D-SETL pattern selection is a manual process, as is the extension of the foundation to form new patterns. Pattern instantiation is simplified via an 4D-SETL Application Programme Interface (API) in subsequent iterations of the DS.

4.3.8 4D-SETL Load BORO UML Domain Ontology (Model) Load Stage

Next the ontology is converted from BORO UML to a set of nodes and edges and loaded to the Graph Database. The structure maintains a direct representation of the BORO UML ontology (model) which will be established in the warehouse. This, in effect, becomes the graph schema that is used to provide the structure and coherence for the graph database. Each element of the domain ontology will be integrated with the foundational ontology. As the domain ontology is in a consistent perdurantist form, ontology matching process can be undertaken. Ontology matching in a manner described by Kalfoglou and Schorlemmer (2003) is undertaken. The results are stored in a warehouse based solution similar to that described by Shvaiko and Euzenat (2013) in Chapter Two. The matching operation determines an alignment of the ontologies in this case the domain ontology and the foundation.

4.3.9 4D-SETL Domain Ontology (Data) Load – Semantic Data Integration

A problem that must be overcome before successful semantic integration can be achieved is to identify when information from multiple source datasets refer to the same real-world object (Stoermer and Bouquet, 2009). In 4D-SETL, this is achieved this by firstly interpreting and transforming the data to an ontologically consistent form, including the data source exemplar names. When an exemplar name is also unique, in that is can refer to a unique entity, these names can be employed as keys *Unique Characters Name Types* – integration points through which to combine disparate datasets. As previously described, within BORO and consequently within 4D-SETL, a *Name* is represented as a *Type* that has member instance *Strings* that are records of utterance events. Obviously it is impractical to record all such *String* utterance events within the *Name Type;* however it is useful to record *Strings that* can be designated as an exemplar and unique identifier. If a *Name* originates from a naming authority, then it is an excellent candidate for such an *Unique Characters Name Types.*

Within the application of the 4D-SETL framework dataset analysis stage such exemplar names are identified. Furthermore, they are also employed to identify the integration points through which new data sets will be integrated, for example the creation and integration of a temporal part (state) that is a location state of a company such Exemplar Names are designated *Unique Characters Name Types* within BORO and within 4D-SETL they are indexed to enable the graph DB API to quickly locate the *Name Type* node identifier. This location state is a spatio-temporal part of the company and of the physical location. The physical location is *Named* by such an *Exemplar Name* derived from the geographic data set. For example there is a *Name Type* W13 NP that names a physical location. This has a *String* 'W13 9NP' as a member instance that is a record of the utterance event from ONS that is designated an *Exemplar Name*. The company dataset also has a *Name String* 'W13 9PY' to designate the location of the registered office – this is utterance event from Companies House. As both these *Name Strings* are utterance events that *Name* the same *Named* thing (a physical location) the company dataset *String'* W13 9PY' can be added as member of the *Name Type* and the other data related to the name can be added to the graph. It is very important to note that it is not the *Names* that are integrated – rather it is the temporal parts that are named that are the subject of the integration.

Therefore in the Figure 4.8, *Location State 2 is integrated - it is a temporal part that is common to both the ACME Company and Location B.* Figure 4.3 also depicts the relationship between the *Name Space*, *Name Type, Exemplar Name,* and *Physical Location*. The integration is achieved when Company *Location State 2* is integrated with *Physical Location 2.*

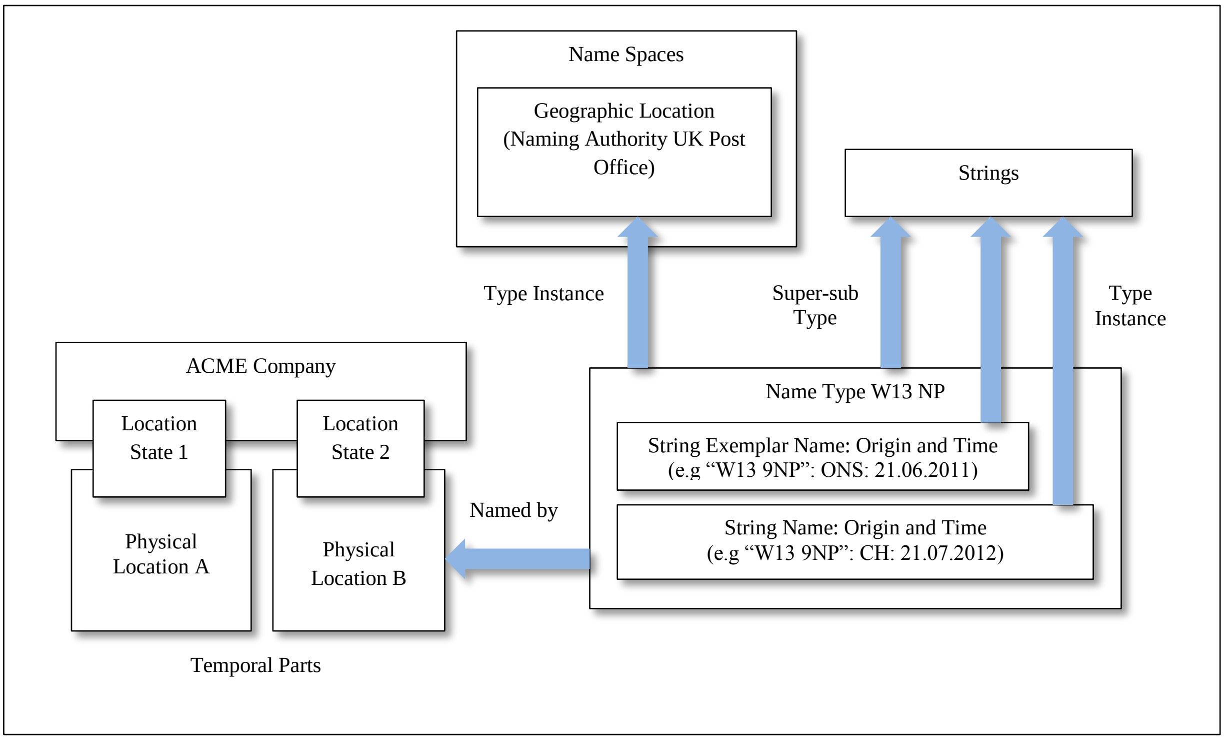

Name Spaces
Geographic Location (Naming Authority UK Post Office)
Strings
Type Instance
Super-sub Type
Type Instance
ACME Company
Location State 1
Location State 2
Name Type W13 NP
String Exemplar Name: Origin and Time (e.g "W13 9NP": ONS: 21.06.2011)
Named by
String Name: Origin and Time (e.g "W13 9NP": CH: 21.07.2012)
Physical Location A
Physical Location B
Temporal Parts

Figure 4.8: Name Space, Name Type and Exemplar- Physical Location Relationship

Domain Ontology Stitching

The first stage is the integration process involves ontologically ‘stitching’ the domain ontology objects into the BORO 4D foundation. This is undertaken manually in this first iteration by developing custom code to perform this action. During subsequent stages this process is automated by a process that employs the object references generated by the Enterprise Architect (Sparx Systems Pty Ltd., 2015) UML design tool to perform domain ontology stitching operation. This process involves locating the unique node identifiers for each of the foundation ontology objects then creating the edges to connect the domain ontology object to the foundation. For example to assert that a domain object is a Type requires a Types Instance edge from the domain to the Foundation Types object. The process uses index (namespaces) that are created for the foundation and each domain ontology to enable a search for each relevant node id during the load process.

Semantic Data Integration – Data Load

Individual fields are then extracted from the source dataset records, transformed to mirror the patterns of the domain ontology then loaded to the warehouse for persistent storage. It is during this process that the integration of individual elements takes place. This can be considered as integration within vertical and horizontal plains. Firstly, the vertical relationships between an individual element and the domain ontology (and hence the foundation ontology – as the domain ontology is linked to the foundation) must be established. This consists of creating the individual relationships (such as type instance/set member) and then the ‘horizontal’ relationships that are deemed to hold between individual domain level objects. This can be a complex transformation that requires both one-to-many and many-to-one transformations. If the individual field that is being processed is a component of a pattern – such as a name, then the whole Name pattern is created and loaded as a subgraph.

4.3.10 4D-SETL Framework Design Summary

The framework employs an approach through which the semantic extraction and transformation is achieved via the analysis of the structure and content of each experimental dataset, firstly to identify the semantics within the context of the Entity Paradigm (3D) and then to apply the BORO REV-ENG (Partridge, 2005) process to re-engineer the data so they conform to the 4D (Object) Paradigm. As 4D-SETL details the events that led to the existence of the object, the states that it is composed of and how the states unfolded over time, the model created documents the lifecycle of the objects that are to be integrated. These details are captured within a worksheet which is employed during the next stage of the stage of the process, which is the development of the domain ontology. During this stage, UML conforming to BORO semantics is employed to extend the foundational ontology patterns to model the required objects and relationships. BORO UML is used as a design document through which to develop the graph database model that is employed to integrate the instance level data. Finally, the domain ontology and dataset are loaded into the graph database. The domain is loaded first and integrated with the foundation ontology. Then, the bulk dataset is loaded and integrated with the domain level ontology.

The 4D-SETL framework has been developed and elaborated and now it can be employed to semantically integrate data based on the use of a foundational ontology that provides that common foundation to the domain and instance data integrated. In the next section the software tools and system components that are developed to support the framework will be described.

4.3.11 Design Science Build 4D-SETL

As discussed in Chapter Two, the results of a semantic data integration process will normally be taken forward to some form of system realisation of which there are two primary types; namely, distributed database and warehouse based. The warehouse based approach has been selected which involves the Extraction Transformation and Loading (ETL) of data from each of the source dataset systems to a single warehouse database system. This is based on a graph database system that enables both the ontology (model) and instance level objects (data) to both be stored in a form that reflect the ontological patterns and therefore this can be achieved without the need of extensive translation.

4.4 4D-SETL Tool Chain Design and Development

As there are no current tools to support the 4D-SETL development, the initial stage required the identification of software and techniques that would support the framework. As asserted by Kääriäinen *et al.* (2009), the development of complex information systems requires practitioners from a number of disciplines to collaborate and that some form of global shared development environment is available to support such cooperation. In the case of this project, although investigated through the work of a single researcher, there is still a need for a range of tools to be linked to produce an environment that will support the design, build and evaluation of artefacts. Within industry this range of software is referred to as a tool-chain. Currently there are no current Integrated Software Development (IDE) systems that support the development of graph database systems that employ 4D ontologies as schemas; therefore, a combination of custom software coding, commercial products and open source software are configured to provide this tool-chain. The tool-chain is established to enable the development and management of artefacts during execution of the 4D-SETL framework. The following sections provide outline details of each of the major tool-chain components.

Dataset Documentation

Dataset documentation is via a worksheet which is a convenient tabular structure to capture the details of the dataset and each of the constituent elements (see appendix C for an example worksheet).

Dataset Analysis and Editing

The VIM editor www.vim.org (2015) is used to edit and view the contents of very large dataset files. This is often necessary to ascertain why an import process failed. The editor can also be employed to generate test datasets by truncating and editing large files. Vim, (a contraction of Vi IMproved), is designed for use both from a command-line interface and as a standalone application in a graphical user interface mode.

4.4.1 Ontology Design and Documentation

At the time of writing no open software tools that provide a full range of UML design and support facilities are available. Therefore, a commercial design tool Enterprise Architect (Sparx Systems Pty Ltd., 2015) was configured to support BORO UML ontology modelling. Enterprise Architect is a mature software product which is based on open standards. The product also features an XMI export facilities which is employed in subsequent stages to automate the production of the graph database representation of BORO UML.

4.4.2 Software Design and Documentation

The Open Source - Eclipse Integrated Development Environment (IDE) (The Eclipse Foundation, 2015) was employed to develop custom software to support the project. This provides Java coding and facilities to support the development of the custom software. JUnit was also used extensively to support a Test Driven Development (TDD) methodology.

4.4.3 Warehouse - Graph Database

The graph database instantiation utilises the Neo4J software (Neo Technology, 2015) which is configured for the purpose of this research to function as a prototype data warehouse system. Using a graph based data warehouse enables the foundational, domain and instance level ontological models to be persisted without the usual translation to tabular form as required by RDBMS. The prototype warehouse system also provides a means of querying the combined data to assess the utility of the resultant artefacts and thus the effectiveness of the perdurantist 4D ontologies to underpin the 4D-SETL framework. Therefore, a technical experiment can be applied as deemed an applicable evaluation method by Hevner, *et al.* (2004). The Neo4J (Neo Technology, 2015) Graph database was employed to provide the warehouse system that enables the integrated data to be structured and stored. This software provides a Graph Database engine, the Cypher (Neo Technology, 2015) declarative query processing system and an Application Programmers Interface (API) through which to configure and load the graph database. The following Figure 4.9 provides an overview of the relationship between the tool-chain and the framework.

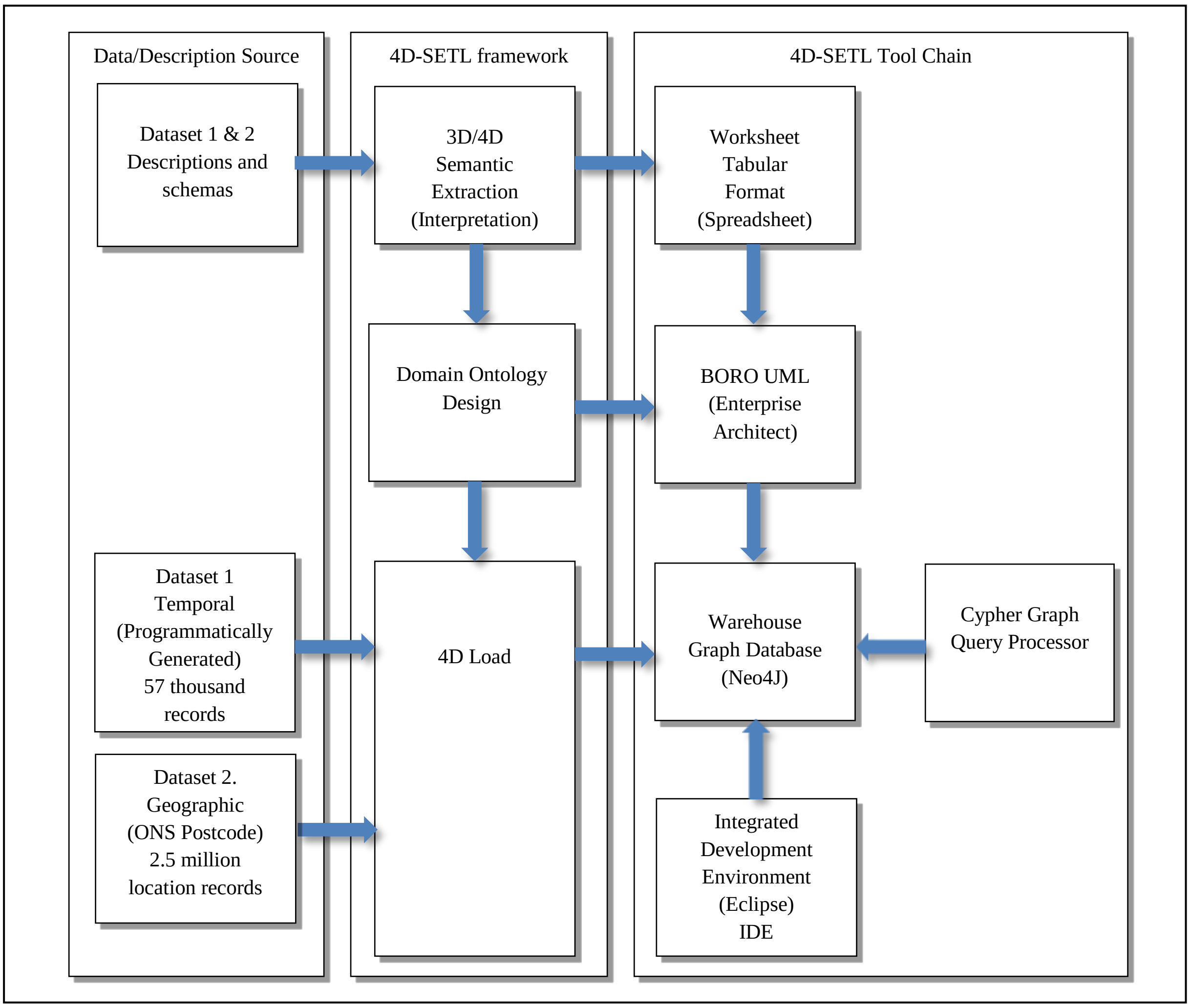

Data/Description Source
4D-SETL framework
4D-SETL Tool Chain
Dataset 1 & 2 Descriptions and schemas
3D/4D Semantic Extraction (Interpretation)
Worksheet Tabular Format (Spreadsheet)
Domain Ontology Design
BORO UML (Enterprise Architect)
Dataset 1 Temporal (Programmatically Generated) 57 thousand records
4D Load
Warehouse Graph Database (Neo4J)
Cypher Graph Query Processor
Dataset 2. Geographic (ONS Postcode) 2.5 million location records
Integrated Development Environment (Eclipse) IDE

Figure 4.9: 4D-SETL Framework - Tool Chain Relationships

4.4.4 Establishing the Ontological Architecture

The following paragraphs describe the ontological architecture of the system which is established during the initial design science iteration. part of the initial iteration.

4.4.5 Implementing the BORO 4D Foundation

At the highest level of the ontological architecture, BORO provides the fundamental ontic categories of Elements, Types and tuples. The BORO ontology is reflexive and it can thus be used to model its own foundation. Tables 4.3 and 4.4 list the BORO foundation ontic categories and ontological patterns respectively together with the UML, BORO UML and Noe4J property graph interpretation.

Object	Description	UML	BORO UML	Property Graph
Objects	This meta-type subsumes the three ontic categories: Types, tuples, Elements.	Class	Type	Node
Elements	This type has all elements as instances. This is an ontic category, with its criterion of identity. Members of Elements are instance of an Individual: something with spatio-temporal extent.	Class	Type	Node
Types	A specification of a Type. This type has all types as instances. This instance relationship is implied but implemented within the model. This is ontic category, with its criterion of identity. Types are non-well founded in that Types is a type – a member of itself.	Class	Type	Node
Tuples	This object is an instance of a Meta-types. It has all tuples as instances. This is its criterion of identity. It is a subtype of Placeable Types. Tuples thus represent relationships among/between objects.	Class	Type	Node

Table 4.3: BORO 4D Foundation Ontic Categories

Object	Description	UML Semantics	BORO UML Semantics	Property Graph Object
Super_SubTypes	Assert super-sub-type relationship	Generalisation	Super subType	Edge
Types_Instances	Asserts the axiom that an instance is a member of a type	Dependency	Types Instances	Edge
Non_Well_Founded_Types_Instances	Reflexive, an object can be a member of itself – for example the foundation object Types (class) is itself a Type (and therefore a member of itself).	Dependency	Non_well_founded_type_instances	Edge

Table 4.4: BORO Foundational Relationships (Partridge 1996)

The BORO foundational patterns form next layer of the ontological architecture, Table 4.5 details these objects.

Object	Description	UML	BORO UML	Property Graph
TupleType	a specification of a Type whose members are tuples	Class	Type	Node
Powertype	A specification of a Type that is the set of all subsets of a given Type. The member Types objects are not instantiated – rather this Type is used to structure the ontological elements that represent higher order elements (types of types)	Class	Type	Node
Structural Patterns (relationships)				
wholePart	a relationship between an Individual and one of its parts (structural)	Class	Tuple	Node + 2 Place Edges
powertypeInstance	a relationship between a Type and its Power Type (structural)	Class	Tuple	Node+ 2 Place Edges
Temporal				
temporalWholePart	A 'wholePart' asserts the spatial extent of the (whole) individual is co-extensive with the spatial extent of the (part) individual for a particular period of time.	Class	Tuple	Node + 2 Edges
TemporalWholePart-Type	A couple between two Individual Types where for each member of the whole set, there is a corresponding member of the part set for which a wholePart relationship exists, and conversely	Class	Tuple	Node + 2 Edges
startBoundary	The beginning of a temporalBoundary.	Individual	Individual	Node
StartBoundary Type	The beginning of a temporalBoundaryType.	Class	Type	Node
endBoundary	A temporal whole part couple that relates the temporal boundary to the whole.	Individual	Individual	Node
EndBoundaryType	A temporal whole part Type that relates the temporal boundary to the whole taken over a Type.	Class	Type	Node
WholePartType	A coupleType that asserts one Type (the part) has members that have a whole-part relation with a member of the other Type (whole).	Class	Type	Node
Names				
Name	A specification of a Name – including exemplar text	These objects form the naming pattern which will be described in details later in this document.		
Naming Space	a specification of a Type whose members are Names			
namedBy	a relationship between a Name and the thing it names			

Table 4.5: BORO 4D Foundation Patterns (Partridge 1996)

Other patterns				
Measure	The magnitude of some attribute of an individual.	Tagged value	Tagged value	Parameter
Measure Type	A category of Measures	Tagged value	Tagged value	Parameter type
Agent	Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability.	Class	Type	Node
Representation	A SignType where all the individual Signs are intended to signify the same Thing.	Class	Type	Node
Representation Scheme	A Representation Type that is a collection of Representations that are intended to be the preferred Representations in certain contexts.	Class	Type	Node
Physical Location				
Location	A point or extent in space that may be referred to physically or logically.	Individual	Individual	Node
LocationType	The powertype of Location	Class	Power Type	Node

Table 4.5: (Continued) BORO 4D Foundation Patterns (Partridge 1996)

4.4.6 Ontology Relationships

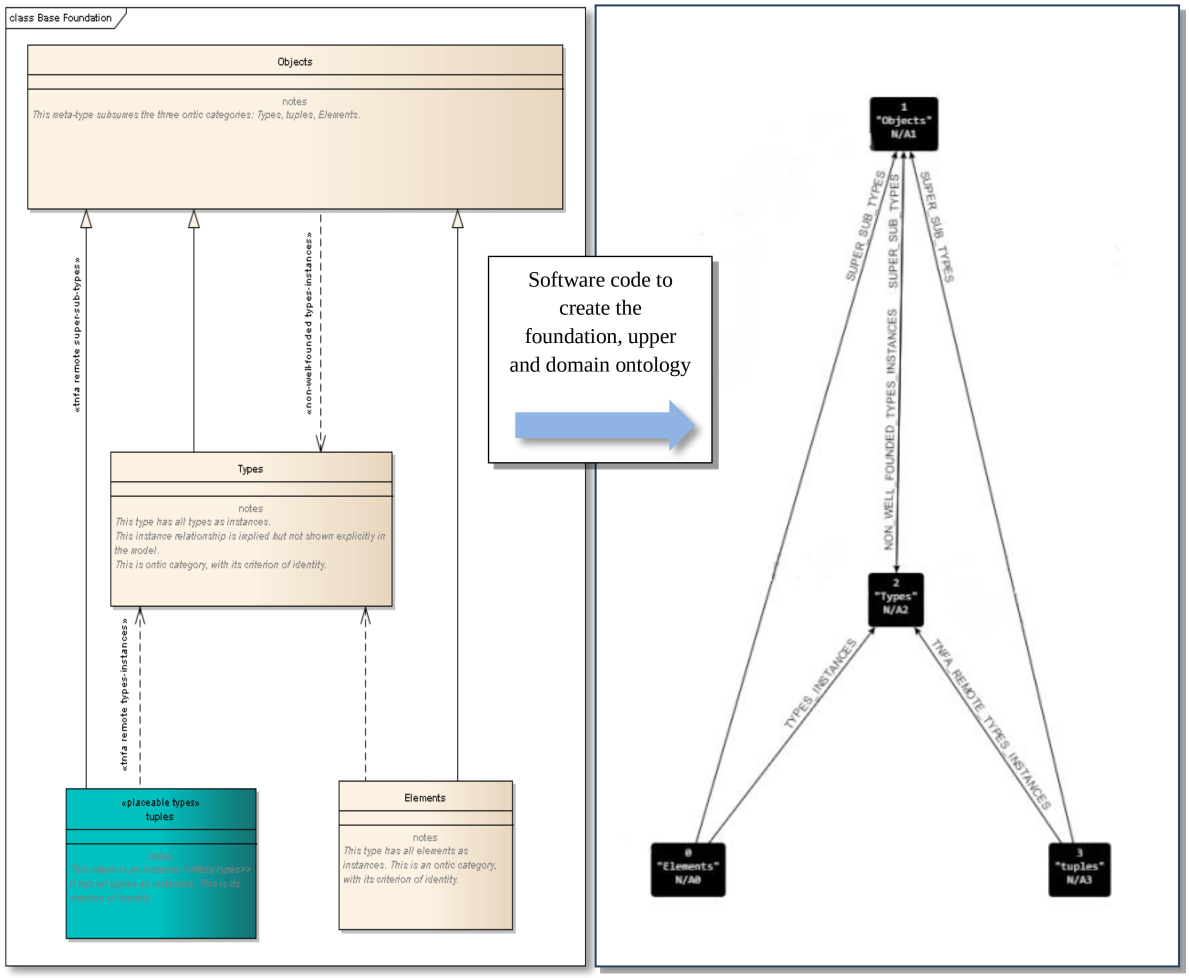
Relationships within the BORO ontology are represented by tuples, which are first class objects within the ontology, and can therefore, represent individual tuples. They can also be a member of a tuple type or higher order tuple type (power types). However, when employing relational objects, there is a problem as asserted by Bradley (1899) who describes the infinite regression which occurs when ‘things’ are associated by relationship ‘objects’. Bradley’s Regress is not a new argument and is related to Plato’s Third Man argument as described by Strang and Rees (1963). Therefore, within the model and graph implementation, relationship objects are asserted through the use of tuple objects. The regression is terminated by the use of ‘place’ binary relationship edges, the alternative i.e. asserting a ‘place’ relationship with another tuple that would lead to an infinite series of tuple objects.

There are a number of exceptions to the rule of using tuples to assert relationships within the framework, for example Non-Well-Founded-Type foundational relationships and the binary relationships Super-Subtype, Power-Type-Instance and Type-Instance are modelled as simple edges.

It is the assertion made though the Non-Well-Founded-Type edge (Types is reflexive) that Types is member of Types – i.e. it is itself a Type that has the effect of making BORO a non-well founded ontology. Therefore, it is possible to establish Types that can lead to a paradox; i.e. The Type of all Types that is not a member of itself (Russell, 1902). This is however unlikely to lead to a problem – as the ontological artefacts produced are being assessed using imperative programming techniques rather than machine reasoners.

4.4.7 BORO UML Model – Graph Database Translation

To instantiate the BORO foundation ontology within the 4D-SETL graph database the BORO UML model was manually transcribed through the development of software application code. Thus a graph database resident representation of the BORO UML model consisting of the patterns of nodes and edges was available to enable the application of the 4D-SETL framework (note this manual process was automated during the course of the second iteration).


class Base Foundation
Objects
notes
This meta-type subsumes the three ontic categories: Types, tuples, Elements.
«tnfa remote super-sub-types»
«non-well-founded types-instances»
Types
notes
This type has all types as instances.
This instance relationship is implied but not shown explicitly in the model.
This is ontic category, with its criterion of identity.
«tnfa remote types-instances»
«placeable types»
tuples
notes
This object is an instance <<Meta-types>> it has all tuples as instances. This is its criterion of identity
Elements
notes
This type has all elements as instances. This is an ontic category, with its criterion of identity.
Software code to create the foundation, upper and domain ontology
1 "Objects" N/A1
SUPER_SUB_TYPES
SUPER_SUB_TYPES
SUPER_SUB_TYPES
NON_WELL_FOUNDED_TYPES_INSTANCES
2 "Types" N/A2
TYPES_INSTANCES
TNFA_REMOTE_TYPES_INSTANCES
0 "Elements" N/A0
3 "tuples" N/A3

BORO Foundation Ontic Categories Modelled in UML Conforming to BORO Semantics

BORO Foundation Ontic Categories recreated within NEO4J Graph Database

Figure 4.10: Foundation BORO UML to Graph DB Translation

The architecture established via a Test Driven Development (TDD) (Fraser *et al.*, 2003) process which is employed to create and configure the research software artefacts. This process and major artefacts produced are depicted in Figure 4.11.

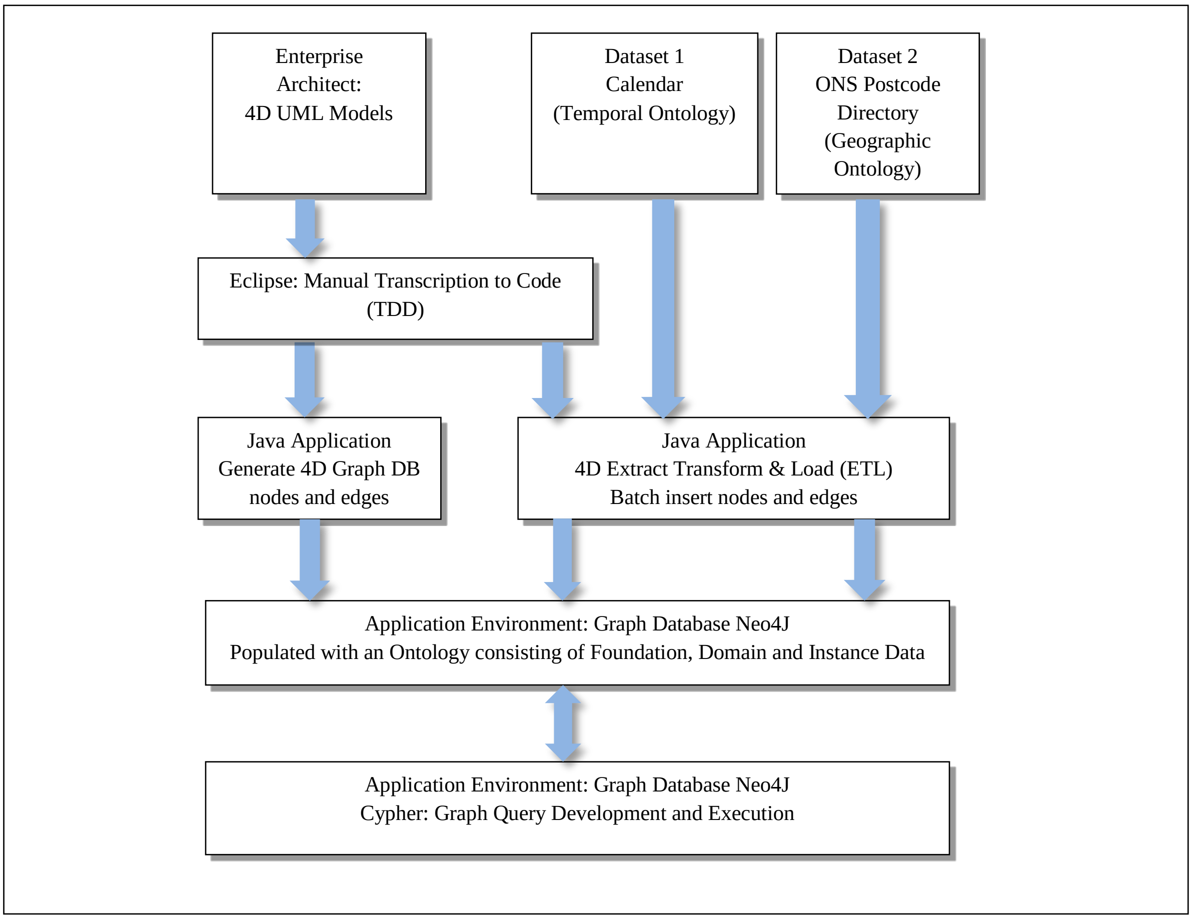

Enterprise Architect: 4D UML Models
Dataset 1 Calendar (Temporal Ontology)
Dataset 2 ONS Postcode Directory (Geographic Ontology)
Eclipse: Manual Transcription to Code (TDD)
Java Application Generate 4D Graph DB nodes and edges
Java Application 4D Extract Transform & Load (ETL) Batch insert nodes and edges
Application Environment: Graph Database Neo4J Populated with an Ontology consisting of Foundation, Domain and Instance Data
Application Environment: Graph Database Neo4J Cypher: Graph Query Development and Execution

Figure 4.11: Iteration One System Software Configuration

The TDD method is adopted as it provides the ability to quickly develop software code through which to explore the functionality of the graph database via the implementation of a number of artefacts. The process involved defining a test, such as ‘connect to the graph database’, then adding new ontology elements via the graph database API and finally developing a function that would enable the test to be passed. As the code was constantly changed through the course of the iteration, re-running the test cases following each change ensured that existing functionality was not adversely affected by a change. As this project is research focussed on the 4D-SETL framework rather than software development, the TDD did not attempt to produce production quality code. The following is a pseudo code listing of the graph database directory and configuration file generation.

```
Map config                                                //New Graph Database configuration (Map)
String graphDbFileName=filename                           //Set DB name – Directory name that will contain the new database
File graphDb = new File(graphDbFileName)                           //Create a new Graph Database
File configFile = new File(fileName)                               // Graph DB Configuration Parameters File
fw = new FileWriter( fileName )                                    // Generate Configuration and Store to disk
fw.append( "neostore.nodestore.db.mapped_memory=5G\n      // Set node store memory cache"
+ "neostore.relationshipstore.db.mapped_memory=5G\n"      //Set relationship memory cache
  + "neostore.propertystore.db.mapped_memory=1G\n"        // Set property (key-value) store cache
+ "neostore.propertystore.db.strings.mapped_memory=1G\n   // Set The size to allocate for memory mapping the string property store
 + "neostore.propertystore.db.arrays.mapped_memory=1G" )  // The size to allocate for memory mapping the relationship store
```

4.5 The First Application of 4D-SETL – Temporal Ontology

This section describes the design science research activities related to the first application of the 4D-SETL framework, applied to integrate the first dataset which consists of a Gregorian calendar. Although the processing and integration of each dataset is presented in the chronological order that it was undertaken, the reader may wish to start with the second application of the framework (section 4.19), as the 4D temporal ontology created to represent that calendar may be quite unfamiliar.

4.5.1 Experimental Data Source One – Calendar (Temporal Domain Ontology)

The first dataset to be interpreted is in the form of a calendar. The elements of which are interpreted to develop the 4D ontological model of temporal patterns that represent time periods and events. Within the temporal ontology days, months and years types are configured. Having a set sequence, the individual objects and relationships that populate the calendar ontology within the warehouse are programmatically generated. The 4D Paradigm (as described in the second chapter), and consequently the 4D-SETL framework, adopts a different view of time than our normal perception provides. This will be described and elaborated during the course of the section that details this integration activity.

Dataset Documentation: Temporal Data Source	
Information Source	Program generated (https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html)
Publisher:	Not applicable
Publication date:	Not applicable
Update frequency:	Not applicable
Scope	Range 1862-Present (approximately 56,210 days). This is based on the needs of the project. The Companies House register oldest company was incorporated in 1862.
Name:	Temporal Ontology
The scope and granularity of the calendar have been selected based on the integration needs of the project. The timespan that the calendar covers is therefore set to the period from the year 1862, when the oldest company in the Companies House dataset was registered to the present. The granularity is set to one day, as this is sufficient for the purposes of structuring and integrating business related events such as company incorporation. **Documentation:** The Gregorian calendar is employed relative to the UK and Greenwich Mean Time (GMT). The calendar does not include future dates, as this would require the ontology to encompass possible worlds, the method through which such object are modelled within the BORO and consequently 4D-SETL.	

Table 4.6: Temporal Data

4.5.2 4D-SETL – Extraction Stage

With regard to the customary endurantist view taken by IS designers, time is simply a value on linear scale. Objects (endurants) and processes (occurrants) are modelled in different forms such as static class diagrams, for the former, and process models for the later. As objects are considered to endure, change is deemed to occur through a change in the accidental attributes which are considered a part of the object. There are also attributes considered essential – which are immutable and do not change. Therefore, the time of the attribute change can be recorded to provide a historic record of how objects within a model of a system evolved over a period of time. Another method employed within IS, is to take a snapshot of the enduring objects and their attributes at set periods of time to record a history. With regard to an endurantist model of a calendar, there are names for instances or periods of time. For example, 21.06.2014 (day:month:year) is the name of a time period starting at 00:00:00 (hour:minute:second) on 21.06.2014 and ending at 24:00:00 on the same day. Therefore the 3D analysis of the calendar yields a set of names that represent the point instance or period values based on a geocentric linear scale. It can be noted that IS normally designate the start of a day slightly after the time 00:00:00 to ensure their time based calculations function correctly.

4.5.3 4D-SETL – Translation Stage

The BORO and, consequently, the 4D-SETL framework adopts a radically different view of time than our normal intuition provides (Partridge, 2005). The extraction process will start with the name that represents a point in time that has been discovered from the extraction stage and translate this into a 4D model.

Taking ’21.06.2014’ from the previous example, this is a string, a physical entity that is a record of an utterance event, and therefore we can apply the *name pattern* (see Figure 4.4). Now that we have established that the element is a *Name* (of a particular day), what this *Name* names, can be analysed. Within the 4D Paradigm any period of time, such as today, is a temporal stage (a slice) of the whole of space-time. For example, the 4D spatio-temporal extent that ‘21.06.2014’ names, consists of all space-time contained within the temporal stage that starts at zero hours and lasts until midnight on 21.06.2014. As this object extends in both time and space, it is a physical body, a 4D element (see Figure 4.6). The two events that are named 00:00:00 (start) and 24:00:00 (end) are also parts of the day 4D element. These start and end events are also physical objects. These objects encompass all of space but have no temporal extension. The events that are employed as the basis of the measurement of time, be they geocentric events such as sunrise or atomic oscillations, are socially constructed and are relative – there is not fixed time.

Therefore, following the translation process, a 4D model has been developed that consists of 4D elements that names (in the form of a physical ‘strings’) and time periods. Event objects have been identified that have a temporal extent of zero thickness that represent start and end events. All other time periods and events follow this same pattern. Table 4.6 details the objects identified.

Object	**3D semantics**	**4D semantics**	**BORO UML**	**Graph Database**
Space-time	No interpretation Space and time exist apart.	Space-time (the universe over eternity). Within the model additional universes may be added to the space-time Type to represent possible worlds	4D Spatio Temporal Extent (Element)	Node
Year	A period demarcated by two geocentric scalar values	4D Element – a temporal part that consist of all space-time between the start and end events. Therefore each year element is related to the space-time element by the mereological temporal-whole part relationship (tuple).	4D State	Node
Month	A period demarcated by two geocentric scalar values	4D Element – a temporal part that consist of all space-time between the start and end events. Therefore each month element is related to a year element by the mereological temporal-whole part- relationship (tuple).	4D State	Node
Day	A period demarcated by two geocentric scalar values	4D Element – a temporal part that consist of all space between the start and end events. Therefore each day element is related to a month elements by the mereological temporal-whole part- relationship (tuple).	4D State	Node
Event	The representation of a point on a scalar value	3D Element – a spatio temporal extent with no temporal extent. It is related to the space-time object by the mereological whole-part relationship (tuple).	3D Element (no temporal extent)	Node
Name	Name	A spatio-temporal extent – in the form of a string or symbol that follows the name pattern	Element	Pattern
Temporal-part-of	None	A Whole-Part relation in the form of a tuple.	Tuple (couple instance)	Node plus two edges
Predecessor Successor	None	A relation in the form of a tuple to enable the temporal ordering of events.	Tuple (couple instance)	Node plus two edges

Table 4.7: Temporal Elements and Relationships

The foundation ontology patterns can now be extended to develop the temporal ontology. The result of the modelling is a domain ontology that is presented in the figure 4.12.

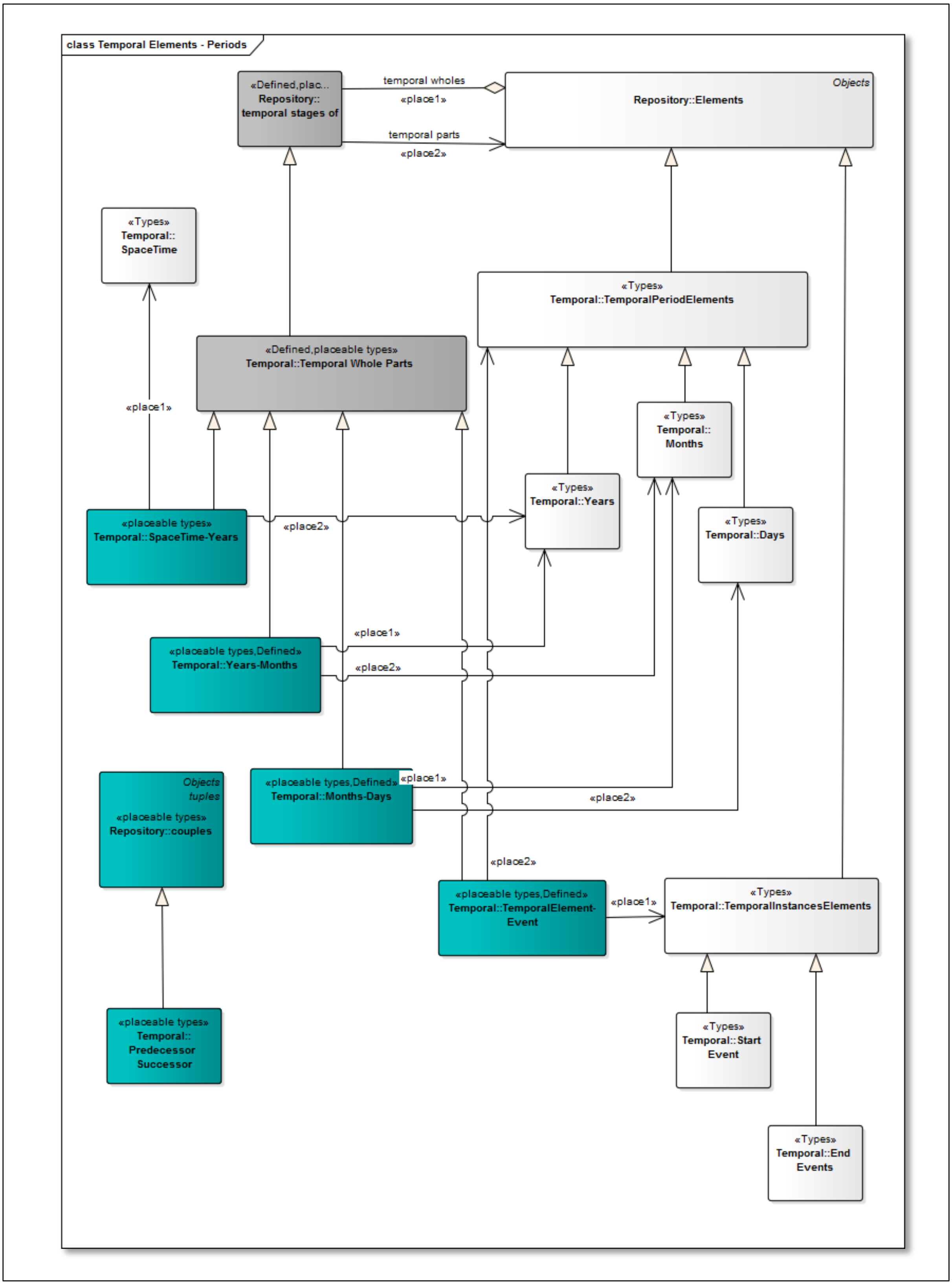


Figure 4.12: Temporal Domain Ontology

4.5.4 4D-SETL – Load Stage

The load stage consists of loading the developed temporal ontology followed by the instance level objects. Firstly, the temporal ontology is loaded to the warehouse system (graph database); this involves connecting the temporal domain level ontology objects with those of the foundation. This integration process has been described as ontological stitching. Within the 4D-SETL framework, indexed name spaces are employed to provide look up services to identify the specific objects that will be integrated. Therefore to enable the objects of the temporal ontology to be integrated, the foundation namespace index is accessed to ascertain the identity of the graph database object that will be integrated. The result of the process is to reproduce the BORO UML patterns within the graph database as depicted in Figure 4.13. Unfortunately the graph visitation tool within Neo4J does not render diagrams that have a resolution for use in documentation. Therefore Figure 4.13 provides an overview only.

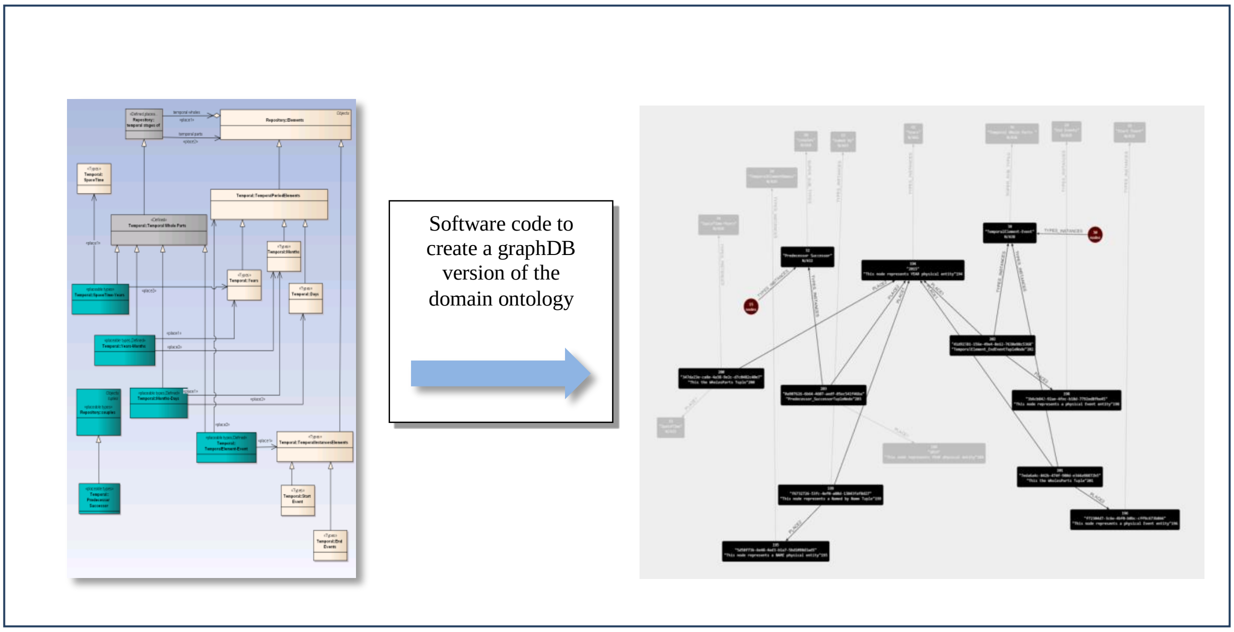

Software code to create a graphDB version of the domain ontology

Figure 4.13: BORO-UML Manual Coding To GraphDB

A Name Space is also created for the temporal ontology which is instantiated as an index within the graph database system. Each element that is loaded to the graph database is inserted into the index as a key, value pair consisting of the exemplar name (string) of the object and the graph database object identifier (node identifier).

Following the loading and integration of the domain ontology, the instance data is loaded. In the case of the temporal ontology, this instance level data is generated programmatically and therefore is a special case. However, this loading process follows the same model as the subsequent datasets. Each dataset source object is integrated into the graph database by creating a pattern of nodes and edges that represent the object. These nodes and edges are, in turn, connected with the temporal domain ontology objects of the graph database by creating tuples (relationship objects) and plain edges (such as *type-instances*) to link a particular day state to the Days Type. When a node is created that represents a unique exemplar name it is also added to the domain name space. In the case of the temporal domain, the granularity of the system is based on that of a day; however month and year individual objects are also included in the domain. For example, the day state named by the name ’21.06.2014’, the month state named by the name ’06.2014’ and the year state named by the name ‘2014’ exist within the graph database. The relationships established within the temporal ontology include super-subtype relationships that are used to specialise general foundational types and the ‘temporal-whole-part tuple’ relations which established the whole part relationships between the space-time spatio-temporal extent element and the year, month and day state (temporal-part) objects. A predecessor-successor tuple type is also employed to provide temporal ordering of each object.

The following pseudo code provides an outline of the node creation via access to the 4D-SETL Application Programmers Interface (API). The node will also be added to the selected (domain) index to enable its reference node identifier to be found during subsequent integration operations.

```
// Using Neo4J Batch Inserter (inserter)   Index employs Apache Lucene
Map nodeProperties                                           // Graph DB Node Key-Values
ontNodeId = inserter.createNode(nodeProperties)                          //inserter is an instance of the batch inserter - pass KV Map
domainToUse = (nodeProperties.get("ontology_domain").toString())           // Set domain (name space)
ontologyUN = ((String) nodeProperties.get("ontology_unique_name")).toString()         // Get the Unique name
indexToUse = domainToUse.concat("_name_index")
Map indexProps = new HashMap()                                            //Initialise Map
switch:                                                      //Select  Index (domain) and add node details
 case: "companies_house_name_index":
indexProps.put("ontology_name", (String) nodeProperties.get("ontology_name"))
ontologyCHNameIndex.add(ontNodeId, indexProps);
case "directors_name_index":
indexProps.put("ontology_name", (String) nodeProperties.get("ontology_name"))
ontologyDirectorsNameIndex.add(ontNodeId, indexProps)
 case "temporal_name_index":
indexProps.put("ontology_name", (String) nodeProperties.get("ontology_name"))
 ontologyTemporalNameIndex.add(ontNodeId, indexProps)
 case "geographic_name_index":
indexProps.put("ontology_name", (String) nodeProperties.get("ontology_name"))
ontologyGeographicNameIndex.add(ontNodeId, indexProps)
case "ons_sic_2007_name_index":
indexProps.put("ontology_name", (String) nodeProperties.get("ontology_name"))
ontologyOnsSic2007NameIndex.add(ontNodeId, indexProps)
```

The name space index look up will be employed during the integration of subsequent datasets to lookup the integration points. Pseudo code to lookup a node is presented below:

```
 String nameKey                                //Ontology element name to search index for
String domain                                  //Domain (namespace) to search
IndexHits ih                                   //Lucene index hits object (employs Apache Lucene to index node and edges)
Long ontTypeNodeId = -3                         //Initialise the graph DB node ID to a negative value
nameSpace = domain.concat("_name_index")       //Switch on the domain (namespace)
Output("switching on namespace =" + nameSpace)  //We have several name spaces established –
switch (nameSpace)                             // for the foundation and each domain ontology
case: " geographic_name_index ":               // Here we search the companies house ontology for an object
ih = ontologyGeographicNameIndex get(nameKey)
if: (ih.size()==1)
ontTypeNodeId = ih.getSingle()                 //Success a unique  node identifier has been found
break
else:
 if(ih.size>1):                                // If there is more than one result, there is a problem with the model
Output("error multiple matches found ")
ontTypeNodeId = -1                              //Set node ID to -1 to indicate error
else:
Output ("error no match found")                //Error  - no results found for the object that was  searched for
ontTypeNodeId  = -2                             //Set node ID to -2 to indicate error
//Setting node id negative ensures it cannot be used
```

Within the integration process, it is worthy of note that it is the *objects* that the exemplar name, names (refer to) that will be the subject of integration rather than their *names*. BORO maintains two separate objects – the ‘thing’ and its name. For example, the event that is named by ‘00:00:00’ is a temporal part of the day named by the string ’21.06.2014’. The name ‘00:00:00’ is not part of the name ‘21.06.2014’. This BORO pattern is employed throughout the 4D-SETL framework.

4.6 The Second Application of 4D-SETL – Geographic Ontology

The first dataset integration has provided the basis of temporal location, enabling datasets that include any form of temporal identifier to be loaded and integrated within the warehouse. This second dataset provides a physical geo-spatial location identifier in the form of a postcode. This location identifier enables other dataset that also reference United Kingdom postcodes to be integrated. These datasets include statutory administrative, electoral, health and other area data. However, in the context of this research, this location identifier will be employed to integrate the Companies House and Directors datasets.

The Office for National Statistics (ONS) (Office for National Statistics, 2015) is the executive office of the UK Statistics Authority, a non-ministerial department which reports directly to the UK Parliament. As a member of the Public Data Group (Public Data Group, 2015), ONS has a remit to publish Open Data that may prove useful to industry and the public. Therefore, ONS publishes a database of all UK postcodes. A postcode is an abbreviated form of address that provides a concise geographic location and a geo-spatial reference that can be employed as a key through which to reference and thus integrate many other datasets. The dataset provides a complete coverage of UK postcodes and includes over 2.5 million records. Table 4.7 provides details of the dataset.

4.6.1 Dataset Two: Office for National Statistics (ONS) Postcode Directory

<table>
<tr><th colspan="2">Dataset Documentation: Postcode Directory</th></tr>
<tr><td>Information Source</td><td>This is drawn from the ONSPD User Guide documentation (UK Office of National Statistics, 2014).</td></tr>
<tr><td>Publisher:</td><td>Office for National Statistics</td></tr>
<tr><td>Publication date:</td><td>12 December 2014</td></tr>
<tr><td>Update frequency:</td><td>Monthly</td></tr>
<tr><td>Size:</td><td>2.5 Million records</td></tr>
<tr><td>Name:</td><td>ONSPD</td></tr>
<tr><td colspan="2"><b>Publisher Description:</b> The Office for National Statistics (ONS) (Office for National Statistics, 2015) is the executive office of the UK Statistics Authority, a non-ministerial department which reports directly to the UK Parliament. As a member of the Public Data Group, (Public Data Group, 2015) ONS has a remit to publish Open Data that may prove useful to industry and the public. In order to fulfil this remit, ONS publishes a database of all UK postcodes which is an abbreviated form of address, made up of combinations of between five and seven alphanumeric characters</td></tr>
<tr><td colspan="2"><b>Description:</b> There are four divisions of the postcode that are: town, city or district falling within the postcode area.
Postcode Area: a unique alphabetic coding by Royal Mail to facilitate the delivering of mail. The area is identified by one or two alpha characters at the start of the full postcode, the letters being derived. 120 postcode areas in Great Britain
Postcode Sector: A sub-area of a postcode district, whose area is identified by the number third from the end of a full postcode. There are approximately 1100 postcode sectors in Great Britain.
Postcode Unit: there are approximately 1.7 million active postcode units in Great Britain and Northern Ireland (2.5 million including discontinues codes) with precise location. Each postcode unit may contain between 1 and 100 addresses. The average number of addresses per postcode is 15 adjoining address locations. The ONSPD contains both ‘live’ postcodes and postcodes which have been terminated by Royal Mail but not subsequently re-used. The ONSPD contains fixed length 7- and 8-character postcode formats and the variable length e-Gif (e-Government Interoperability Framework) standard postcode format.
Postcode Grid Reference: The Postcode grid references in the dataset provide a one metre resolution and the majority are derived from the Ordnance Survey. The grid references provided for Northern Ireland postcodes are derived from the LPS product ‘Pointer’ and use the Irish National Grid system that covers all of Ireland and is independent of the British National Grid. No grid references are provided for postcodes in the Channel Islands and the Isle of Man.
Post Office (PO) Boxes and Non-Geographic Postcodes: Non-geographic postcodes can either be special postcodes assigned to some large users of the postal service or PO Boxes that lie within a (pseudo) postcode district that does not form a discrete part of a post town. These will all have been assigned a grid reference, usually the local Royal Mail sorting office, and the majority have a Postcode Quality Indicator (PQI) of 1(highest) but some were assigned a PQI of 6 (lowest accuracy).
Terminated Postcodes: Postcodes are terminated by Royal Mail, for example due to the demolition of buildings or to postcode reorganisations. Terminated postcodes can are occasionally re-used by Royal Mail but not usually before an elapsed period of three years. In such circumstances, all terminated postcodes and their grid references are retained on the ONSPD and a ‘termination date’ is added which provides a clear indication of a postcode’s status. When a postcode is re-used by Royal Mail the previous grid reference and termination date are removed, thus deleting all reference to the former existence of the postcode from the ONSPD.</td></tr>
</table>

Table 4.8: Postcode Dataset Description (Office for National Statistics, 2015)

4.6.2 4D-SETL - Stage 1 Extract

The first stage of the process is to document each of the data source elements which in this case consist of tabular data. Therefore, column headings, together with descriptive information, are recorded in the form of worksheet. This forms the initial natural language model that describes the fundamental constructs contained within the dataset. As the 4D ontology represents how objects are created and the temporal parts from which they are constituted, detail of the life cycle of the objects are also required.

Scope

The process is to decide which of the fields are within the scope from the integration project and then to seek further information that may be of relevance in the form of supporting documentation. Within this worked example, we will consider fields PCD, DOINTR, OSEAST1M DOTERM and OSNRTH1M are to be in scope from the 53 available fields.

Data description	Range of codes/ Entity code	Comments	Field name
Unit postcode 7 character version	AB1Δ1AA-ZE999ZZ (maximum range	All current ('live') postcodes within the United Kingdom, the Channel Islands and the Isle of Man, received monthly from Royal Mail. Also, all terminated ('closed') postcodes that have not been subsequently re-used by Royal Mail within the United Kingdom and by the postal administrations in the Channel Islands and the Isle of Man.	PCD
Date of introduction	YYYYMM (year and month)	The most recent occurrence of the postcode's date of introduction	DOINTR
Date of termination	YYYYMM(year and month)	If present, the most recent occurrence of the postcode's date of termination, otherwise: null = 'live' postcode	DOTERM
National grid reference – Easting	numeric or null	The Ordnance Survey postcode grid reference Easting to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSEAST1M
National grid reference - Northing	numeric or null	The Ordnance Survey postcode grid reference Northing to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSNRTH1M
The table contains a further 44 fields			

Table 4.9: Partial List of ONSPD Field Descriptions (Office for National Statistics, 2015)

4D-SETL Worksheet

The first action is to collect information relating to the ONSPD experimental datasets in terms of background information and to document these details together with the schema details in a tabular form. Secondly, to decide on the scope of the integration process in terms of the fields that will be selected to be processed and integrated. The third step is to interpret the data. Normally such files are the result of some form of export process from a relational database system. Whilst the data is syntactically structured (in terms of rows and columns), the semantics of such datasets are implicit and therefore cannot be readily integrated with other datasets. However, Chen's (1976) rules for mapping the natural language descriptions into Entity Relational (ER) form can be employed (Chen, 1976) as previously described in Section 4.3.7.

Worksheet Extract: Entity Relational – 3D Real world Semantics

Next the ER model is transformed into the form of ontology using 3D semantics as detailed in Table 4.12.

Table, Table Column Description	Comments	Field name	Entity Semantics	Entity real word semantics
Table Name	Postcode		Entity Type Name	A set of Names of physical locations
Unit postcode 7 character version	All current ('live') postcodes within the United Kingdom, the Channel Islands and the Isle of Man, received monthly from Royal Mail. Also, all terminated ('closed') postcodes that have not been subsequently re-used by Royal Mail within the United Kingdom and by the postal administrations in the Channel Islands and the Isle of Man.	PCD	Attribute Type	Attribute Type: Names of an endurant objects
Date of introduction	The most recent occurrence of the postcode's date of introduction	DOINTR	Attribute Type	Attribute Type: Time index
Date of termination	If present, the most recent occurrence of the postcode's date of termination, otherwise: null = 'live' postcode	DOTERM	Attribute Type	Attribute Type: Time index
National grid reference – Easting	The Ordnance Survey postcode grid reference Easting to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSEAST1M	Attribute Type	Attribute Type: Scalar Value
National grid reference – Northing	The Ordnance Survey postcode grid reference Northing to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSNRTH1M	Attribute Type	Attribute Type: Scalar value

Table 4.10: Worksheet Extract: Entity Relational – 3D Real world Semantics

4.6.3 4D-SETL Transform

Within this stage the dataset referents are transformed. This involves interpreting the dataset element under the 4D Object Paradigm. In this stage, the conceptual patterns for the entities are translated into 4D objects. Firstly, an example record (row) i.e. the postcode represented by the string, '*UB8 3PH'* is transformed to a name that within 4D semantics is interpreted as physical entity. Secondly, name '*UB8 3PH'* is used to name an object that is a physical area.

This object is transformed from a 3D object (endurant) to 4D object (perdurant) that has both spatial and temporal extent. The next step is to identify which foundational category the object belongs to – which in the case of the string *'UB8 3PH'* and the physical area it names are both elements.

The next two fields contain names which represent creation (introduction) and dissolution (termination) dates. These two fields are semantically transformed to the Object Paradigm by interpreting the fields as names (elements) of events (elements with no temporal extension). As with many events, the accuracy is only given to the level of a day – however it can be can assumed that at some point during the day of introduction, there was a physical act such as a signature that meant at that at one instant the introduced state of the postcode started. As with all such events the accuracy is only ever approximate – as time continuous and therefore a point can never be defined with absolute accuracy.

The next two fields are scalar values representing eastings and northings. The terms easting and northing relate to plane coordinates that are used to locate position with respect to a two-dimensional plane surface (map) that depicts features on the curved surface of the Earth. The scale value is referenced in relation to the UK Ordnance Survey National Grid.

These two scalar values therefore define a point location – the location of the physical element named by the postcode. Scalar values are transformed to parameters that are part of an object state. Therefore, within the 4D-SETL framework they become parameters that are incorporated with a node object in the form of a key, value pairs.

Next the schema level objects are transformed. The postcode table itself represents a Type – postcodes. The columns are detailed in Table 4.8.

Table Column Description	Comments	Field name	ER	3D	4D
Unit postcode 7 character version	All current ('live') postcodes within the United Kingdom, the Channel Islands and the Isle of Man, received monthly from Royal Mail. Also, all terminated ('closed') postcodes that have not been subsequently re-used by Royal Mail within the United Kingdom and by the postal administrations in the Channel Islands and the Isle of Man.	PCD	Attribute Type	Attribute Type: Names of endurant objects	1) Type: 2) Element (Name pattern)
Date of introduction	The most recent occurrence of the postcode's date of introduction	DOINTR	Attribute Type	Attribute Type: Time index	1)Name of an event 2)Event
Date of termination	If present, the most recent occurrence of the postcode's date of termination, otherwise: null = 'live' postcode	DOTERM	Attribute Type	AttributeType: Time index	1)Name of an event 2)Event
National grid reference – Easting	The Ordnance Survey postcode grid reference Easting to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSEAST1M	Attribute Type	Attribute Type: Scalar Value	Parameter: scalar value Node K/V
National grid reference – Northing	The Ordnance Survey postcode grid reference Northing to 1 metre resolution; blank for postcodes in the Channel Islands and the Isle of Man. Grid references for postcodes in Northern Ireland relate to the Irish Grid system.	OSNRTH1M	Attribute Type	Attribute Type: Scalar value	Parameter: scalar value Node K/V

Table 4.11: Worksheet Extract: Entity Relational – 3D and 4D Real world Semantics

Types, tuples and Elements are identified by their 4D extension which exists in both space and time. Elements can also be equated to individuals i.e. objects that do not have member instances. In the case of the previous example as both objects extend into both time and space – they can be classified as Elements.

Types are identified by their member instances and tuples are identified by the objects in each of the tuple places. In the example being described the postcode physical geo-political region persists through time and therefore is classified as an element. The next step in the process positions the object within the overall ontological model establishing the relationships with the foundation, upper and other domain ontology objects.

4D Pattern Identification

Next the dataset patterns are interpreted. There are a number of 4D patterns which can be identified and extended for use within the domain model. These include, naming and structural patterns such as whole-part relations.

The foundational pattern to apply in the example case is that of Type membership – asserting that a physical geopolitical region is an instance of the Elements Type.

Comments	Field name
Name pattern	Postcode
Temporal location (from temporal domain ontology)	DOINTR
Temporal location (from temporal domain ontology)	DOTERM
Parameter pattern	OSEAST1M
Parameter pattern	OSNRTH1M

Table 4.12: Geographic Patterns

Following this analysis, information is recorded in the 4D-SETL Worksheet which is depicted in the Figure 4.14.

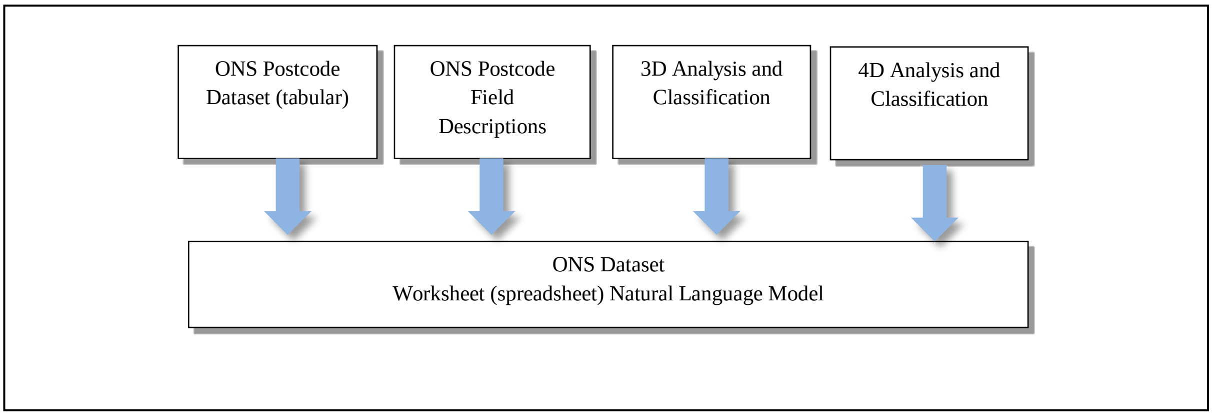

ONS Postcode Dataset (tabular)
ONS Postcode Field Descriptions
3D Analysis and Classification
4D Analysis and Classification
ONS Dataset
Worksheet (spreadsheet) Natural Language Model

Figure 4.14: Inputs to the natural language model Worksheet

4D-SETL Domain Ontology Design (BORO UML)

The geographic domain ontology design progresses as described during the previous domain design i.e. 4D-SETL worksheet, Enterprise Architect UML and manual transcription to the graph database.

4.6.4 4D-SETL Load - Semantic Integration

The first stage is the integration process involves ontologically 'stitching' the domain ontology objects into the BORO 4D foundation. This is undertaken in the BORO UML modelling stage which extends the BORO 4D foundation and upper ontology to encompass the modelled objects of the domain. Within the first iteration this was undertaken manually with specific software being developed to generate the domain ontology objects.

The second stage of the process involves transforming and loading the instance level data to the graph database. The data file containing the postcode data was therefore processed to extract and ontologically transform and 'stitch' into the domain ontology. In addition to the foundation – elements are also integrated with the temporal ontology which has been established. This integration is achieved through the accessing the foundation and temporal name space indices to locate the integration points to which the instance level data would be integrated. In this case it is the introduction and dissolution events that are integrated with the calendar ontology.

4.7 Evaluation

This initial iteration has produced a number of artefacts of which the 4D-SETL framework is the prime. The warehouse system containing the foundation ontology and two domain ontologies has been instantiated to evaluate the framework; however, at this stage the model is rather sparse but provides the foundation of geospatial (geographic) and temporal location which will provide a foundation for the integration of subsequent datasets.

4.7.1 Research Output Artefacts

The first iteration of the design science design and implement research activities produced four main artefacts:

a) The 4D-SETL Framework

b) The tool-chain through which the 4D-SETL framework could be applied.

c) Two domain ontologies that are used to structure the integrated data.

d) A warehouse system populated with the foundation, geographic and temporal ontologies together with the combined instance level data.

The evaluation of these artefacts is the third phase of the iteration. Guidance as to the evaluation method applicable to each of these artefacts is drawn from the design science methodology provided by March and Smith (1995), who assert that the evaluation activity must examine the qualities of the artefacts produced to help to define what the research aims has accomplished. Two evaluation methods are selected that are deemed by accepted design research mythology to be applicable to the artefacts produced.

Firstly, a technical experiment is possible as a prototype instantiation is available through which to perform such testing. Secondly, an illustrative scenario is employed to describe the facets of the 4D-SETL framework which prove effective in the performance of semantic integration.

4.7.2 The Technical Experiment

In addition to the worksheets and BORO UML models, software artefacts have been developed that consist of java application code (classes) together with JUnit code that is used during the technical experiment of the application code and to confirm it function results are in accordance with that expected. The main subject of the technical experiment is the ability of the 4D-SETL framework to integrate and process data at scale and thus confirm the viability of the framework.

Technical Experiment Host Environment

The technical experiment is to execute the code and to examine the test results in terms of the execution times, and resultant graph database artefacts. Figure 4.15 depicts the physical systems architecture of the system that was configured to support the 4D-SETL experiment.

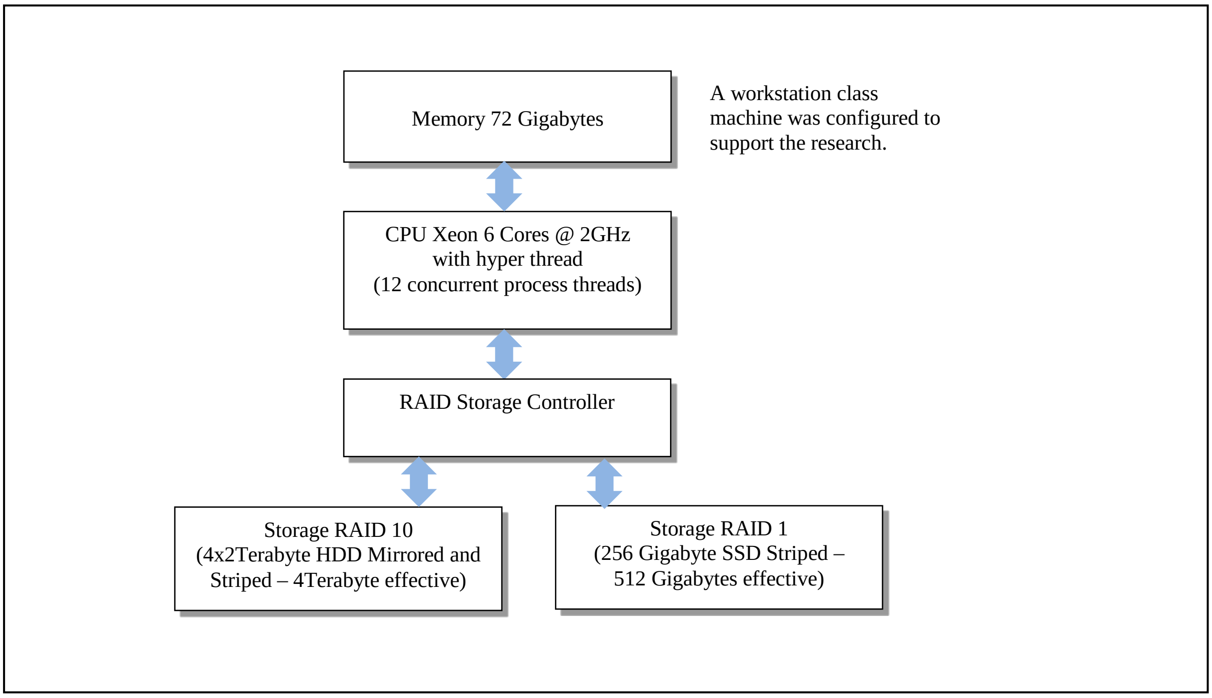

Memory 72 Gigabytes
A workstation class machine was configured to support the research.
CPU Xeon 6 Cores @ 2GHz with hyper thread (12 concurrent process threads)
RAID Storage Controller
Storage RAID 10 (4x2Terabyte HDD Mirrored and Striped – 4Terabyte effective)
Storage RAID 1 (256 Gigabyte SSD Striped – 512 Gigabytes effective)

Figure 4.15: Technical Experiment Hardware Configuration

Technical Experiment - Foundation Ontology Load

The first experiment that is executed generates an empty graph database together with the relevant configuration file then to create the foundational ontology (graph) within the warehouse. From the scale perspective this is rather limited as the base foundation only consists of 26 Types. Each node and edge is created with an associated key value store that is indexed to provide lookup facilities. The experiment concluded with the warehouse populated with the foundational ontology. The following listing was retrieved to verify the results of the experiment containing details of the nodes, relationships and indexes created. See Appendix A for the resultant node-edge listing. The test results are detailed within Table 4.14.

Technical Experiment - Temporal Ontology Load

The next technical experiment loads the warehouse graph database with the temporal ontology, again this is a rather small scale experiment. However, it does confirm that the basic mechanism used throughout the integration process functions. In this first application, each element of the temporal domain ontology is integrated with the foundation. This is achieved through an index lookup to locate the graph node identifier of the target of the connection (edge), the source of the connection is established when a node that is the subject of the integration is created (the create process returns this identifier). During this experiment each element of the temporal ontology is associated with the foundation through either a relationship – tuples or a plain edges (for Type member and Super Sub Type relationships). Following the load of the Types, often termed the T-Box within AI (the terminological assertions), instance level data is loaded. This is often termed the A-Box (assertional box) the assertions (knowledge) related to individuals. This load is undertaken in three stages as temporal parts are associated i.e. a day (21.06.2014) is a temporal part of a month (06.2014) etc. Therefore the month element must be present within the ontology to be able to asset that a particular day is a temporal part of a particular month. A predecessor-successor relationship is also asserted within the ontology to provide temporal ordering of states.

The graph is starting to grow with the addition of the temporal instance level data. As each temporal element such as a day has an associated name (pattern) and start and end events (day).

Geographic Ontology Load (Types and Instances)

Following the temporal ontology the geographic ontology is loaded followed by the instance level model that consists of 2.3 million records.

The results of the Foundation, Temporal and Geographic ontology load are detailed in table 4:14.

As can be gathered from the table, there are resource costs associated with storing the ontological model and instance level data within the graph warehouse.

Firstly, rather than storing a group of elements as is the case within a RDBM relation instance (table row)– the graph model requires the creation of individual nodes to represent each element, then to assert a relations for each individual element. For example to assert that an individual day is named 2014-06-21, has a start end event, is of type day and is part-of the month 2014-06. Thus systems that have larger scale physical disk and memory resources are required to support the graph model. There is also a cost associated with loading each element node and the relationships that hold for such an element. This requires extensive index lookups to ascertain the identifiers of the related elements. For example, the day element 2015-06-14 is a mereological part of the month 2014-06. Therefore after creating the node representing the element – the relationship will be created. This requires an index look up to ascertain the node identifier of the month element 2014-06. This accounts for the long load times when creating large numbers of nodes and relationships and the extensive index look up operations that are required to locate the relationship end-points.

Although slow to load the graph model has advantages over that tabular RDBM model as relationships can easily be discovered without the cumbersome join operations that are required by RDBMS based systems.

The temporal ontology forms the base for all temporal information that will be added to the graph store. For example a company formation (incorporation) event will be a temporal part of a particular day. In the same manner the geographic ontology forms the base for all location information added to the graph database; the location of a company will be a physical part of a particular physical location. Thus to retrieve a list of all companies formed on a particular day at a particular location simply requires a graph traversal from the nodes representing a particular day/location.

Graph database creation and Foundation ontology load Test			
Execution time 1.8 Seconds Database disk storage 2.133 megabytes			
Nodes	Properties	Relationships	Relationship Types
26	374	78	6
Temporal ontology instance level – Years (1862-2015) Load test			
Execution time 3.5 Seconds Database disk storage 6 megabytes			
Nodes	Properties	Relationships	Relationship Types
1 594	18193	3043	6
Temporal ontology instance level – Months(1862-2015) Load test			
Execution time 20 seconds Database disk storage 51 megabytes			
Nodes	Properties	Relationships	Relationship Types
20074	228865	38155	9
Temporal ontology instance level – Days(1862-2015) Load test			
Execution time 693 seconds #(11 minutes) 154 years multiplied by 365 (approx.) = 156210 days Each day will have a name and an associated start and end event Database disk storage 1.5GB			
Nodes	Properties	Relationships	RelationshipTypes
638,791	6,866 ,011	1,106,848	9
Geographic Ontology Types Load test			
Execution time 3.1 seconds Database disk storage 1.5GB			
Nodes	Properties	Relationships	Relationship Types
638 798	6 866 067	1 106 855	9
Geographic Ontology Instances Load test			
Execution time 8645 seconds (2.4 Hours) Database disk storage 18.2GB			
Nodes	Properties	Relationships	Relationship Types
12,749,950	130,230,645	20,399,920	10

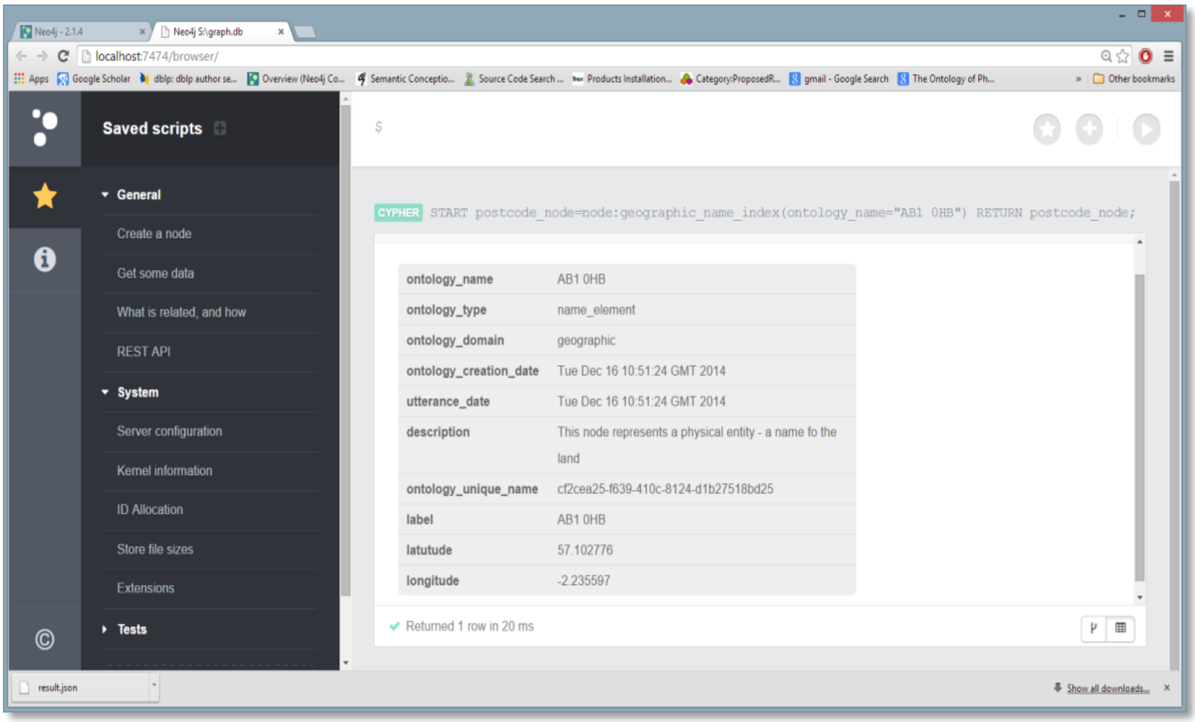
Table 4.13: Technical Experiment – Load Results

Loading Instance Data - Indicative Metrics

Creating the graph based model from imported instance data provides a load performance slightly better than that of a Relational Database (inserting 10,000 postcode (subgraphs) records into Neo4J took 30 seconds vs PostgreSQL RDBM 40 seconds (using the standard transactional interface). However, this is without the extensive lookup operations that will be required to load and integrate larger datasets at scale. It is expected that performance will degrade as the number of integration point look-ups increases. This will be investigated during the subsequent interactions.

Query Response - Indicative Metrics

Querying the graph, via an index lookup operation, to locate and return a node and its related parameters took approximately 20 milliseconds. This performance would be acceptable for an interactive web application. The information retrieval performance will be investigated further during the subsequent interactions when graph traversals will be tested and evaluated.



```
CYPHER START postcode_node=node:geographic_name_index(ontology_name="AB1 0HB") RETURN postcode_node;
```

ontology_name	AB1 0HB
ontology_type	name_element
ontology_domain	geographic
ontology_creation_date	Tue Dec 16 10:51:24 GMT 2014
utterance_date	Tue Dec 16 10:51:24 GMT 2014
description	This node represents a physical entity - a name fo the land
ontology_unique_name	cf2cea25-f639-410c-8124-d1b27518bd25
label	AB1 0HB
latutude	57.102776
longitude	-2.235597

✓ Returned 1 row in 20 ms

Figure 4.16: Simple Graph Query – Returning Node Parameters

Technical Experiment Evaluation

The technical experiment empirically tested the load of the graph database. Following the completion of these test the warehouse contained the following:

a) Foundation ontology.

b) Temporal Ontology (Types and Instances).

c) Geographic Ontology (Types and Instances representing 2.5 million locations).

The experiments revealed that the graph database performed well, including the creation of graph elements (ontology) models at scale. There is however, a heavy load in terms of index lookups that are required to effect the integration. An improvement to the load software will be implemented during the following iteration to cache the domain ontology integration points within the bulk data load processing application.

In relation to the evaluation of the effectiveness of the 4D-SETL, this related to addressing the need for solutions that will scale (issue 8). The Graph database has been loaded with several million nodes and edges representing the foundation temporal and geographic ontologies. Furthermore, the graph has been queried to assess the data retrieval capability in terms of response time, which is within the response times that would support interactive applications.

4.7.3 Illustrative Scenario

Grounding

The domain ontologies produced are grounded by the foundational ontology and therefore each instance node within the graph database is connected to the foundation via the domain level objects. Therefore, we can conclude that 4D-SETL is effective in addressing the grounding issue described within Table 1.1.

Name and Named Object - Clear separation

As described previously, the event that is named by ‘00:00:00’ is a temporal part of the day named by the string ’21.06.2014’. The name ‘00:00:00’ is not part of the name ‘21.06.2014’. This BORO pattern is employed throughout the 4D-SETL framework. Therefore, we can conclude that 4D-SETL is effective in providing a clear separation of the Name and Named element into separate but related structures Table 1.1 Issue 6).

Temporal Location Patterns

The temporal location patterns physicalised time and enabled a more accurate account of what time represents and how it becomes an integral part of the system rather than some form of external index. Within the ontic commitments inherent to the adoption of BORO, which takes the eternalist philosophical stance, objects can exist in the past, present or future and all are equally real. Therefore the 4D-SETL is well suited to storing historical data and to model possible future scenarios (with the addition of objects to model possible worlds). This addresses the weakness described within Table 1.1, Issue 5.

The temporal ontology being graph based also raises the possibility of developing algorithms to identify temporal patterns within data, for example the page rank algorithm could be employed to find the day (node) which had the maximum number of company dissolutions (future work).

Physical Location Patterns

Physical location patterns are a straightforward representation of an object's relative position in space. However, on closer examination of the source data, there are a number of problems. Firstly, terminated postcode Names are reused after a period of three years – to refer (or name) to a new physical location. As there is no indication in the dataset to show that this reuse has occurred, gives rise to the possibility that when historic data is loaded to the warehouse, a postcode may reference a completely different physical area to the one that was originally intended when the dataset was compiled. Postcodes may be allocated to large organisations that do not represent a geographic location. Postcodes are of varying accuracy and are only given as a point co-ordinate.

Other datasets could be added to provide a full polygon on the postcode area, therefore we could judge that model fruitful in terms of the domain models produced by 4D-SETL can meet currently unspecified requirements and can be easily extended.

It can, however, be judged in terms of the fidelity of the model to the reality it attempts to abstract. This provides advantages for semantic interpretation and the foundation for a warehouse that can be extended to hold any number of new datasets and their associated ontologies without changing any of the existing underlying structures. Typically a data warehouse implemented using RDBMS and the associated ER model will need to be backed up and reloaded to implement schema changes.

Within the 4D-SETL framework the full life cycle of the objects abstracted from reality to form the ontological models has been considered. This has uncovered a number of issues and enabled consideration of their effects on the accuracy of the information system. The postcode reuse issue could be addressed – as a name is a record of an utterance event when names a particular physical location, if we had a full historical dataset the included these utterance events, which unfortunately is not the case with the experimental dataset used in this research.

4.8 Lessons Learned

4.8.1 4D-SETL Manual Ontology Transcription

The manual transcription of the BORO UML Models to the Graph database was employed during the initial iteration, however it was found impractical due to being both time consuming and error prone. Therefore, the next iteration will need to provide an algorithm to automatically translate the BORO UML patterns from the Enterprise Architect UML design tool to the graph database.

4.8.2 Extended API

The load operation was implemented by directly accessing the Graph Database Application Programmers Interface (API). This was time consuming and led to much duplicated code. Therefore, the next iteration will feature an API to call common patterns such as named-by, located at etc. The API should also cache domain integration points (such as Types) to reduce the overhead of index lookup operations.

4.8.3 Batch Mode – Bulk Load

Following the initial testing of the Semantic Extract and Load times, it became apparent that to load large datasets, a batch mode inserter needed to be employed rather than using the transaction interface. This will restrict the use of a number of features such as traversal and transactions during the graph database population; however, it provides a speed increase of approximately six fold versus the transaction interface. Therefore, rather than being implemented as an improvement during the next iteration, it was immediately implemented to ensure the bulk load operation complied in a timely manner.

4.8.4 Pattern Heuristics

The selection of patterns is based on the

4.9 Summary and Conclusion

This iteration of the design science, core design-build-evaluate cycle, represents the process that is used to generate new knowledge. This iteration consists of two major stages. The first stage is the design and build of the 4D-SETL framework itself (including the loading of the foundational ontology to the graph database), the second stage consist of the design and development activities that are conducted via the application of the 4D-SETL framework itself. This second activity consisted of the extraction transformation and loading of the first two experimental datasets. At this stage, the overall design science evaluation is weak, due to the simplicity of the initial two datasets. However, the framework has been applied, experience gained and lessons learned. The integration has also resulted in the framework, and a warehouse instantiation populated with ontologies that represent all UK postcode locations and all temporal states (year, month and day) from 1862 to the present. The scope of the calendar coverage was selected based on the needs of the Companies House integration were the first company incorporation event in the dataset occurs during the year 1862.

Analysis of the geographic dataset has revealed that there are issues related to the postcodes. Firstly, they not represent a non-geographic location, as is the case for large corporations. Secondly, due to the fact that codes are reused – the code, over the course of time refers to a range of different locations. This is a barrier to the integration of historic data if the movement of companies over time – would not be absolutely reliable.

The 4D-SETL analysis stage, which considers not just structural elements of the dataset but how the elements evolve over time has proved of value by providing insight into such issues as name reuse.

CHAPTER 5. DESIGN SCIENCE SECOND ITERATION – 4D-SETL IMPROVEMENT AND FURTHER APPLICATION

5.1 Introduction

This chapter describes the execution of the second iteration of the design science methodology - the design-build-evaluate cycle. As with the first iteration the overall aim is to discover new knowledge related to the exploitation of 4D perdurantist ontologies for semantic integration through act of designing, building and evaluating artefacts. This aim is met through applying the 4D-SETL framework to more complex and larger scale datasets. Therefore, more complex ontological patterns are required for loading and integration of larger scale instance level data with the number of graph nodes and edge connections grows into the millions. An additional objective of the iteration is to improve the framework and address the deficiencies identified in the previous iteration by redesigning the 4D-SETL and software tool-chain.

This cycle builds on the experience and knowledge created during the previous iteration. The 4D-SETL framework is again applied to the integration of two experimental datasets with the resultant integrated datasets stored within a graph database warehouse system. The evaluation methods identified by and described by Hevner *et al.* (2004) that are applicable to the artefacts resulting from this iteration, namely the technical experiment and the applicable illustrative scenario.

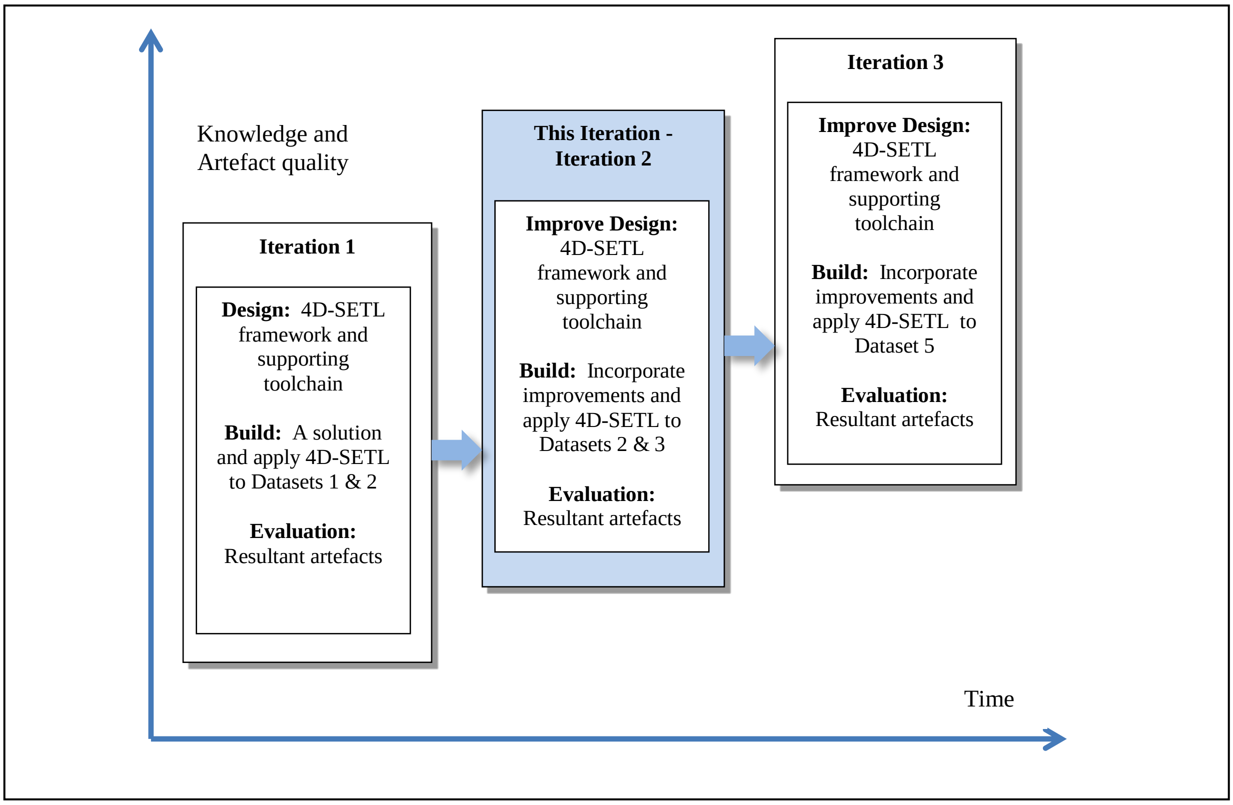

Knowledge and Artefact quality
Iteration 1
Design: 4D-SETL framework and supporting toolchain
Build: A solution and apply 4D-SETL to Datasets 1 & 2
Evaluation: Resultant artefacts
This Iteration - Iteration 2
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Datasets 2 & 3
Evaluation: Resultant artefacts
Iteration 3
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Dataset 5
Evaluation: Resultant artefacts
Time

Figure 5.1: Research Design-Build-Evaluate Cycle Interaction Two

5.1.1 Aim and Objectives of Iteration Two

The overall aim of the second iteration is to continue the evaluation of the effectiveness of employing a 4D ontology to facilitate the semantic interpretation and integration of datasets. This aim is met through the set of objectives outlined in Table 5.1.

Stage	Activity	Objective	Section Reference
Design	1	Improve the design the of 4D-SETL framework.	5.
	2	Improve the design the tool chain to support the framework	5.2.1
Build	3	The application of the 4D-SETL framework to develop and load the SIC 2007 domain ontology and instance level data.	5.3
	4	The application of the 4D-SETL framework to develop and load the Companies House domain ontology and instance level ontology (bulk load).	5.4
Evaluation:	5	Technical experiment heuristic testing of the warehouse artefacts.	5.5.1
	6	Illustrative scenario: describing the artefact’s effectiveness.	5.5.5

Table 5.1: Stages and Activities and Within this Iteration

Following the completion of the evaluation of the artefacts produced by the first iteration of the design science cycle, the knowledge gained is again employed to inform the subsequent and final iteration. Through this process, the quality of the artefacts produced by the subsequent iteration of the design science cycle can be improved. Figure 5.2 depicts 4D-SETL framework, activities and related artefacts

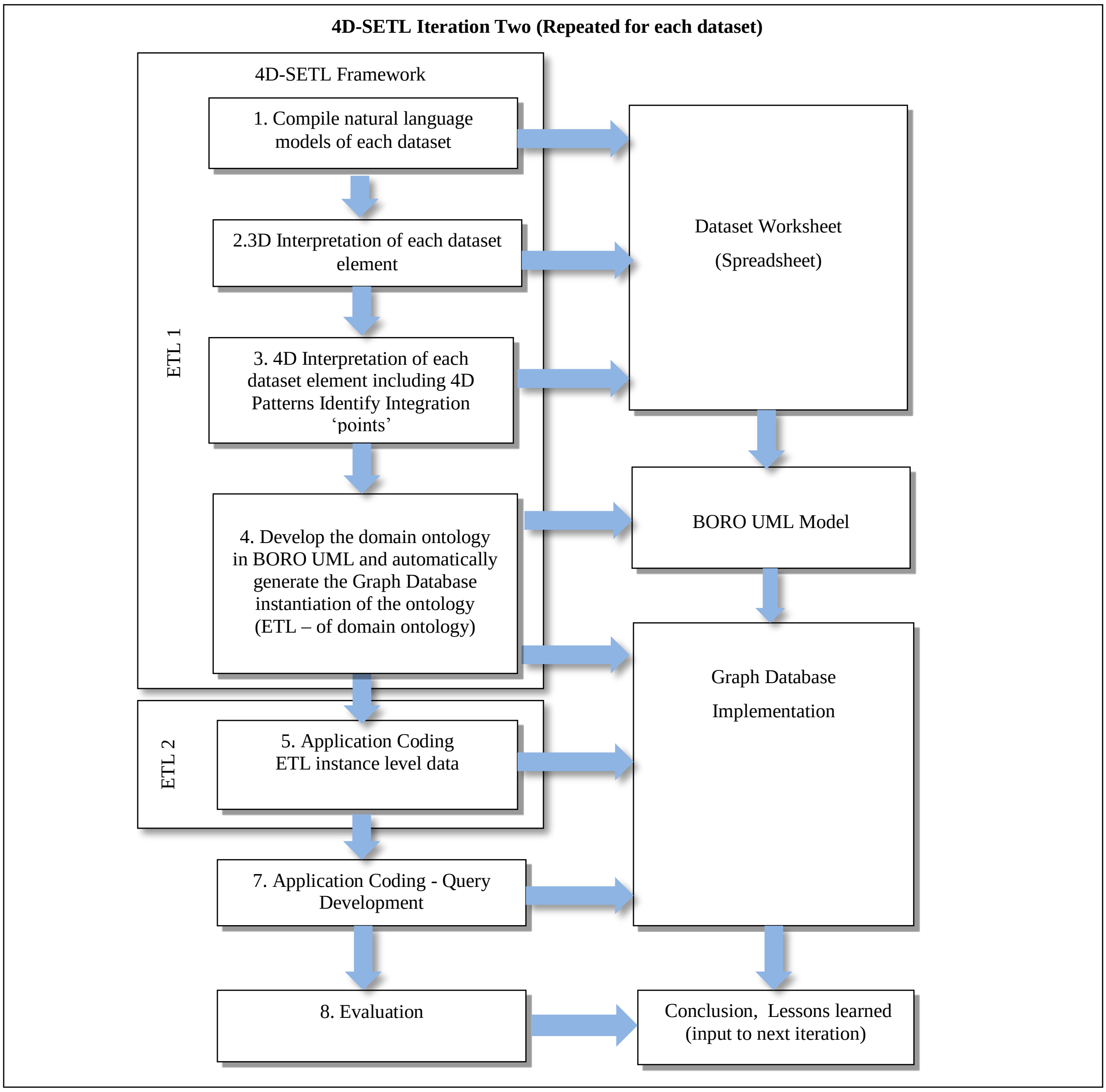

4D-SETL Iteration Two (Repeated for each dataset)
4D-SETL Framework
1. Compile natural language models of each dataset
2.3D Interpretation of each dataset element
3. 4D Interpretation of each dataset element including 4D Patterns Identify Integration 'points'
4. Develop the domain ontology in BORO UML and automatically generate the Graph Database instantiation of the ontology (ETL – of domain ontology)
ETL 1
ETL 2
5. Application Coding ETL instance level data
7. Application Coding - Query Development
8. Evaluation
Dataset Worksheet (Spreadsheet)
BORO UML Model
Graph Database Implementation
Conclusion, Lessons learned (input to next iteration)

Figure 5.2: Overview of the Second DS Iteration and the Resultant Artefacts

5.1.2 Experimental Data Sources for Iteration Two

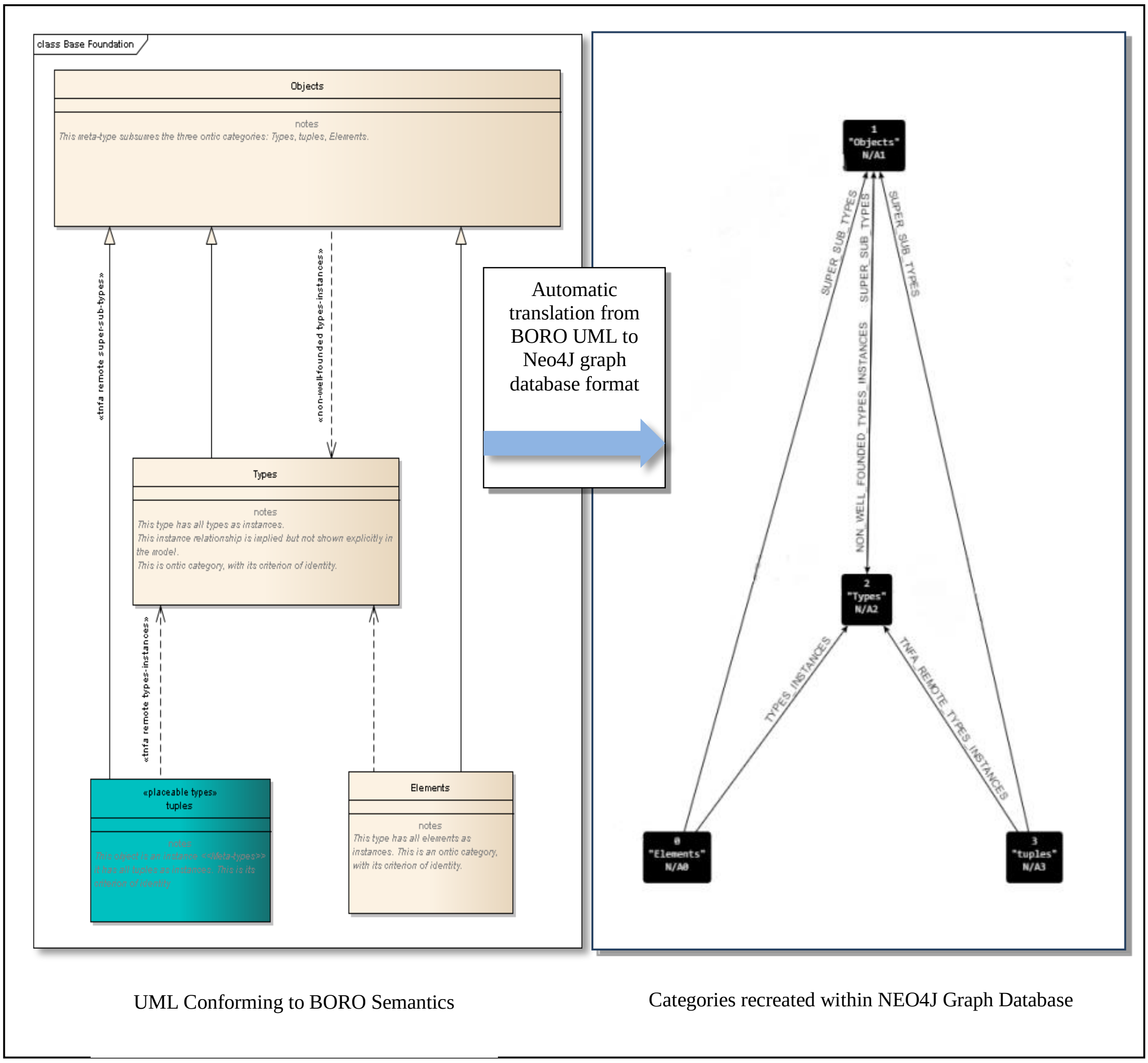
The second iteration utilises two data sources related to the UK Business Domain, the first relates to industrial activity classification. The second relates to company information in the form of UK company basic financial and registration details. Both datasets are in tabular form. However, the Companies House dataset requires extensive analysis of the background information as there are many structures that are not evident from the dataset alone – such as company legal status taxonomy etc.

5.2 4D-SETL Enhancement

A number of deficiencies were identified in the tool chain supporting the 4D-SETL framework during the first application of the 4D-SETL framework. These improvements are developed during the course of this iteration to enhance the effectiveness of the framework. These enhancements are described in the following sections.

5.2.1 BORO UML Model – Graph Database Automatic Translation

During the initial iteration of the design science cycle, in order to establish the foundation and domain ontologies that form the 4D-SETL graph database 'schema', the BORO UML models that represent these ontologies were manually transcribed through the development of software application code. This was sufficient for the purposes of the initial exploration of the graph database technology and the framework; however, it was found to be time consuming and error prone. Therefore, an enhancement was required to solve these problems. Consequently, for the second iteration, a new version of the application code was designed and developed to automatically translate and load the foundation and domain ontologies to the warehouse graph database system as depicted in Figure 5.3.


class Base Foundation
Objects
notes
This meta-type subsumes the three ontic categories: Types, tuples, Elements.
«tnfa remote super-sub-types»
«non-well-founded types-instances»
Types
notes
This type has all types as instances.
This instance relationship is implied but not shown explicitly in the model.
This is ontic category, with its criterion of identity.
«tnfa remote types-instances»
«placeable types»
tuples
notes
This object is an instance <<Meta-types>> it has all tuples as instances. This is its criterion of identity
Elements
notes
This type has all elements as instances. This is an ontic category, with its criterion of identity.
Automatic translation from BORO UML to Neo4J graph database format
1 "Objects" N/A1
2 "Types" N/A2
0 "Elements" N/A0
3 "tuples" N/A3
SUPER_SUB_TYPES
SUPER_SUB_TYPES
SUPER_SUB_TYPES
NON_WELL_FOUNDED_TYPES_INSTANCES
TYPES_INSTANCES
TNFA_REMOTE_TYPES_INSTANCES

UML Conforming to BORO Semantics

Categories recreated within NEO4J Graph Database

Figure 5.3: BORO UML to Graph DB Automatic Translation

As with the previous software development, a Test Driven Development (TDD) approach was employed to create and configure the research artefacts. This enhanced process is depicted in Figure 5.4.

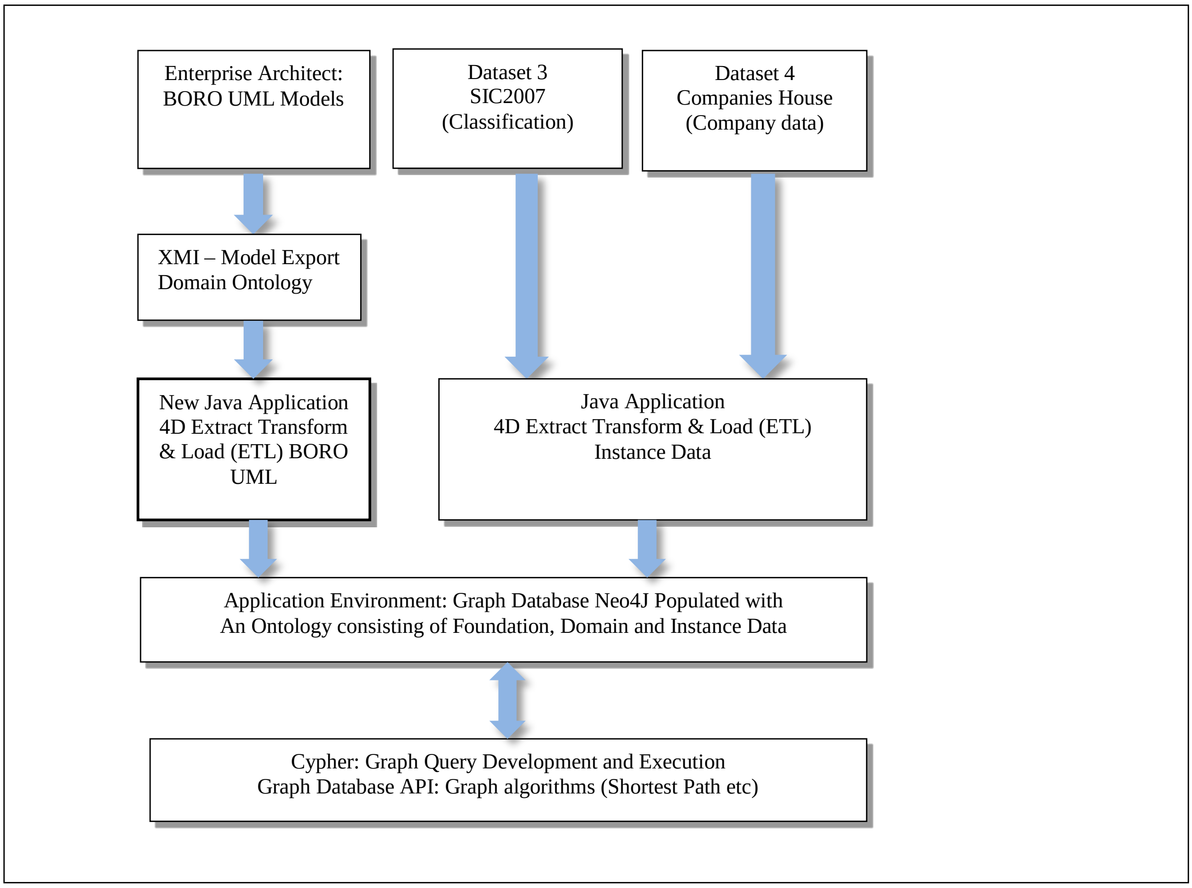

Enterprise Architect:
BORO UML Models
Dataset 3
SIC2007
(Classification)
Dataset 4
Companies House
(Company data)
XMI – Model Export
Domain Ontology
New Java Application
4D Extract Transform
& Load (ETL) BORO
UML
Java Application
4D Extract Transform & Load (ETL)
Instance Data
Application Environment: Graph Database Neo4J Populated with
An Ontology consisting of Foundation, Domain and Instance Data
Cypher: Graph Query Development and Execution
Graph Database API: Graph algorithms (Shortest Path etc)

Figure 5.4: Iteration Two System Software Configuration

The BORO UML models that represent the 4D ontological models are exported to XMI directly from the Enterprise Architect design tools. New application code has been developed to convert this XMI export from this tool and process the artefacts in a similar manner to the instance level datasets via an Extract, Transform and Load. Typically XMI is employed to transform and translate UML models between different design tool vendors, as described by Kovse and Härder (2002). However, within 4D-SETL the BORO UML (XMI) model is transformed into a graph database representation. The foundation and each of the domain ontologies are modularised as UML packages that can be exported individually.

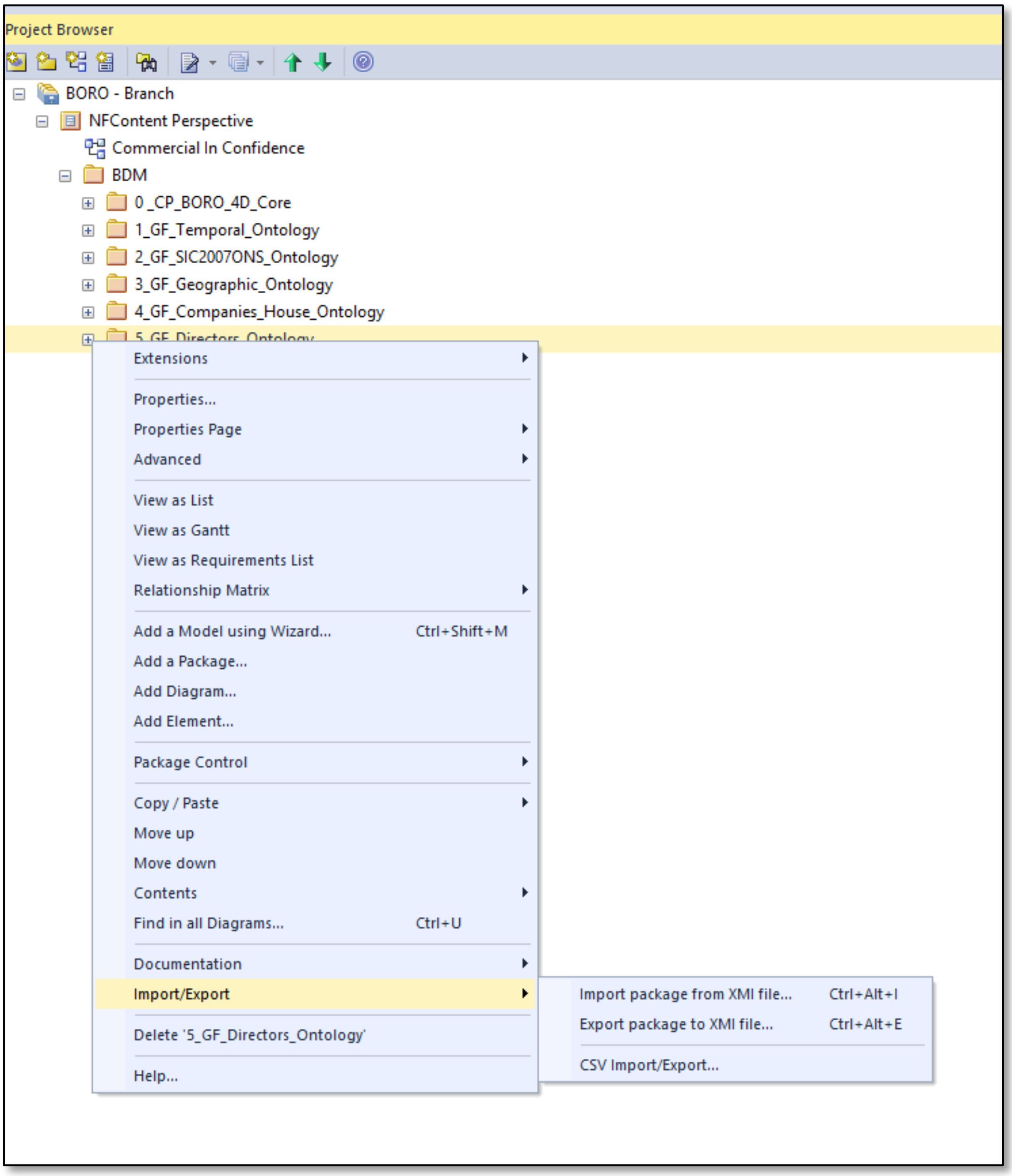


Figure 5.5: Design Tool Enterprise Architect BORO UML (XMI) Export

Following the export, the XMI is parsed and Types, Elements, Foundational Relationships and tuples are Extracted, Transformed and Loaded (ETL) to the graph database.

The following code is used to parse the XMI (UML XML Model) to extract a given UML Type (Classes, Associations, SuperSubTypes, Instances, Custom Profile types).

Pseudo Code

```
Using jdom2 a Java Document Object Model (DOM)  in-memory XML model that can be used to read, write, create and modify XML Docs
Using Java filewriter to write output to a delimited file (which is ETL as per the dataset files)
Document  doc                              // new DOM object
String fourDType                           //Initialise values
String UML type
String className
String fileName
String filter                              // filter that will be used to select and return elements of the BORO UML XMI
umlCsvOut = new FileWriter(fileName)      //create new file to store the results of the extract        //Set namespace
Namespace xmi = Namespace.getNamespace("http://schema.omg.org/spec/XMI/2.1")    //set xmi name space
ElementFilter filter1=new ElementFilter("packagedElement")    // filter on UML Packaged Elements
Element rootNode=doc.getRootElement      //Set traversal start point at the XMI document root
For loop (Element c:rootNode.getDescendants(filter1))   // Traverse xml document getting all descendant nodes
    List<Attribute> attribList1  = c.getAttributes()             // Load each node returned to a list element
    Iterator<Attribute> iterator1 = attribList1.iterator()      //create iterator
    if :(className.equals(c.getAttributeValue(type, xmi)))
        umlCsvOut.write(fourDType+",")                         //Write the first column with the 4D object type
    loop while (iterator1.hasNext)      //Inner(while) loop -
        if(className.equals(c.getAttributeValue(type, xmi)))         //extract type
            Attribute attrib1=iterator1.next()
            umlCsvOut.write(attrib1.getValue())
            if(iterator1.hasNext())
                umlCsvOut.write(",")
        else:  Iterator1.next()
    end while loop
    if(className.equals(c.getAttributeValue(type, xmi)))
        umlCsvOut.write(System.getProperty("line.separator" ))
end for loop
umlCsvOut.close()    //Close File
```

5.2.2 Framework Enhancement

Following the change to the framework and supporting tool-chain, the ‘schema’ level ontology element are now extracted (from the UML), then processed in the same ways as the instance level dataset, i.e. it is transformed from a delimited file to a set of nodes and edges that represent the ontological model that is loaded to the graph database. Therefore, the 4D-SETL framework now encompasses two ETL stages; one for processing the ontology produced via BORO UML such as Types and domain patterns and one for processing the bulk dataset ontology (instances or classes).

5.2.3 Domain Ontology Type Node ID Caching

Following the evaluation of the 4D-SETL instance data load times, it became apparent that to process and load large datasets, the API that was used to generate nodes and edges for the import process would need to cache the identifiers of commonly accessed node types (domain ontology types). This was implemented during this iteration.

5.3 The Third Application of 4D-SETL – SIC 2007 Integration

The design science iteration now moves to the third application of the 4D-SETL framework which will process and integrate a dataset representing a classification system consisting of a hierarchy of taxonomic ranks and classes.

5.3.1 The SIC 2007 Experimental Dataset

The Standard Industrial Classification (SIC) was first introduced into the UK in 1948 for use in classifying companies by the economic activity in which they engage. The classification provides the means though which to analyse information related to business activity, and a means for classifying industrial activities into a common structure (Office of National Statistics, 2011). The current classification system SIC 2007 is published by the Office for National Statistics (ONS) which is the UK's independent producer of official statistics and is the recognised national statistical institute for the UK. It is the executive office of the UK Statistics Authority and although they are separate, they are still closely related (Office of National Statistics, 2011).

As with the geo-spatial dataset (postcodes) SIC 2007 is also a key dataset as it provides a common reference to other datasets related to business activity. It is also employed by the Companies House dataset that is also integrated during the course of this iteration. Table 5.2 details the dataset

Dataset Documentation: SIC 2007 Description	
Information Source	This is drawn from the UKSIC documentation (UK Office of National Statistics, 2014).
Publisher:	Office for National Statistics
Publication date:	2007
Update frequency:	Since 1948 UKSIC has been updated in 1958, 1968, 1980, 1992, 1997, 2003 and 2007.
Size:	The scope of the integration with SIC 2007 is to import and structure the complete dataset. SIC 2007 has 21 sections, 88 divisions, 272 groups, 615 classes and 191 subclasses. NB The taxonomic ranks: sections, divisions, classes and subclasses are modelled in BORO UML . the remaining types are bulk loaded from delimited files
Name:	ONSPD
Publisher Description: The Office for National Statistics (ONS) (Office for National Statistics, 2015) is the executive office of the UK Statistics Authority, a non-ministerial department which reports directly to the UK Parliament. As a member of the Public Data Group, (Public Data Group, 2015) ONS has a remit to publish Open Data that may prove useful to industry and the public. In order to fulfil this remit, ONS publishes a database of all UK postcodes which is an abbreviated form of address, made up of combinations of between five and seven alphanumeric characters	
Documentation: The Office of National statistics has developed and published the Standard Industrial Classification (2007) with the intention of providing a national standard system for the classification of the business economic activity within the UK. In addition to providing a classification system for the UK, the standards also align with the Nomenclature statistique des Activités économiques dans la Communauté Européenne (NACE) and the United Nations International Standards Classification System (ISCS). Within the statutory reporting framework required by all companies in the UK, businesses are required to report their class of activity within the company annual return submitting up to four activity codes to Companies House. Over time there have been several different SIC systems used in the UK which reflect the changing nature of business activity within the UK. The latest version of these codes (SIC 2007) was adopted by the UK as from 1st January 2008. With the agreement of the Office of National Statistics (ONS), Companies House uses a condensed version of the full list of codes available from ONS.	

Table 5.2 SIC 2007 Experimental Dataset

5.3.2 4D-SETL – Extract Dataset Three SIC 2007

As previously described, the first stage of the 4D-SETL processes is to document the dataset in the form of a worksheet containing each of the data source column headings together with descriptive information. The primary source of information is in the form a natural language model provided by the ONS that describes the fundamental constructs contained within the dataset.

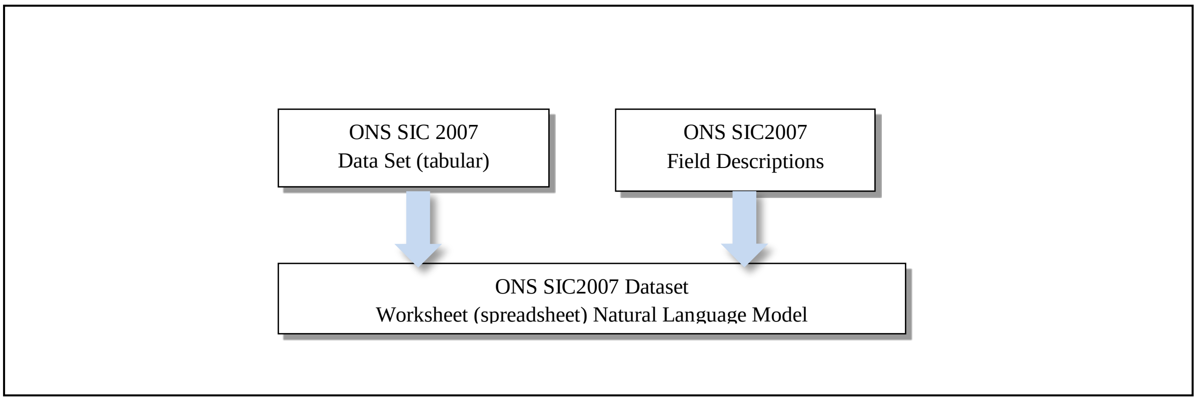


Figure 5.6: Inputs to the natural language model Worksheet

Pattern Identification

The pattern that can be applied in the case of the SIC 2007 is that of a classification system which is composed of *Types* and *PowerTypes (Types of Types),* i.e. second order objects that are described as taxonomic ranks. Within the dataset there are no individuals only *Types*. SIC 2007 follows the well-known structure that has been adopted by numerous domains, the most notable of which is the Linnaean system for biological classification. The structure is taxonomical and divided into levels. These levels are known as taxonomic ranks (or simply ranks) within the Linnaean system they are Kingdom, Phylum, Class, Order Sub Order Family and Genus. The ranks of SIC 2007 are: sections, divisions, groups, classes and subclasses. An example is provided in Table 5.1.

Rank	Specific Instance	Description
Section	Section B	Mining and quarrying
Division	Division 5	Mining of coal and lignite
Group	Group 5.1	Mining of hard coal
Class	Class 5.10	Mining of hard coal (repeat class name)
Subclass	Subclass 5.10/1	Mining of hard coal from deep coal mines (underground mining)

Table 5.3: SIC 2007 Ranks and Example Instances

As Table 5.3 shows, besides providing a classification structure, SIC also provides a standard naming scheme for all levels of economic activities.

5.3.3 4D-SETL – Translation Stage

Following the translation process, a 4D model has been developed that consists of 4D spatio-temporal extents that are names (in the form of physical 'strings') that name states. As discussed previously, a company can have a temporal part that is being a mobile phone manufacturer. This temporal part of the company's spatio-temporal extent (STE) can also be a member of a Type defined by SIC2007 that has all these states as members. The identity of this Type is therefore defined by the member spatio-temporal parts.

Object	3D semantics	4D semantics	BORO UML	Graph Database
SIC 2007	No interpretation	4D 3rd Order Type. An instance of a Powertype-Powerype-Powertype. This translated to a class of classes of classes.	PowerType (x3) Instance	Node
Section	No interpretation	4D 2nd Order Type. An instance of a Powertype-Powertype	PowertType(x2) Instance	Node
Division	No interpretation	4D 2nd Order Type. An instance of a Powertype-Powertype	PowertType(x2) Instance	Node
Group	No interpretation	4D 2nd Order Type. An instance of a Powertype-Powertype	PowertType(x2) Instance	Node
Class	No interpretation	4D 2nd Order Type. An instance of a Powertype-Powertype	PowertType(x2) Instance	Node
Subclass	No interpretation	4D 2nd Order Type. An instance of a Powertype-Powertype	PowertType(x2) Instance	Node

Table 5.4: Objects and Relationships

The foundation patterns can now be extended to develop the SIC 2007 ontology.

Foundational Object Categorisation

Applying the REV-ENG BORO process to derive the basic category of each of these objects yields the following:

a) The SIC 2007 taxonomic rank objects: Sections, Divisions, Groups, Classes and Sub Classes contain member instances and are therefore categorised as Types.

b) Each SIC 2007 individual business activity object contains members that are physical states of a company and are therefore Types.

Pattern Identification

As the SIC 2007 taxonomic rank objects are Types that have members that are Types, the objects conform to the BORO ontology Power Type pattern. This is a general pattern that can be employed to model higher order objects. This scheme can be extended ad infinitum to model any level of higher order objects. Thus, if one wished, it would be possible to include a third order Type that contained members that are the classification schemes SIC 2007 and the previous six schemes dating back to 1948.

Following the identification of the patterns the 4D-SETL worksheet is completed with the 4D interpretation and the framework can then move to the next stage, the development of the ontology.

4D-SETL SIC 2007 Domain Ontology Design (BORO UML)

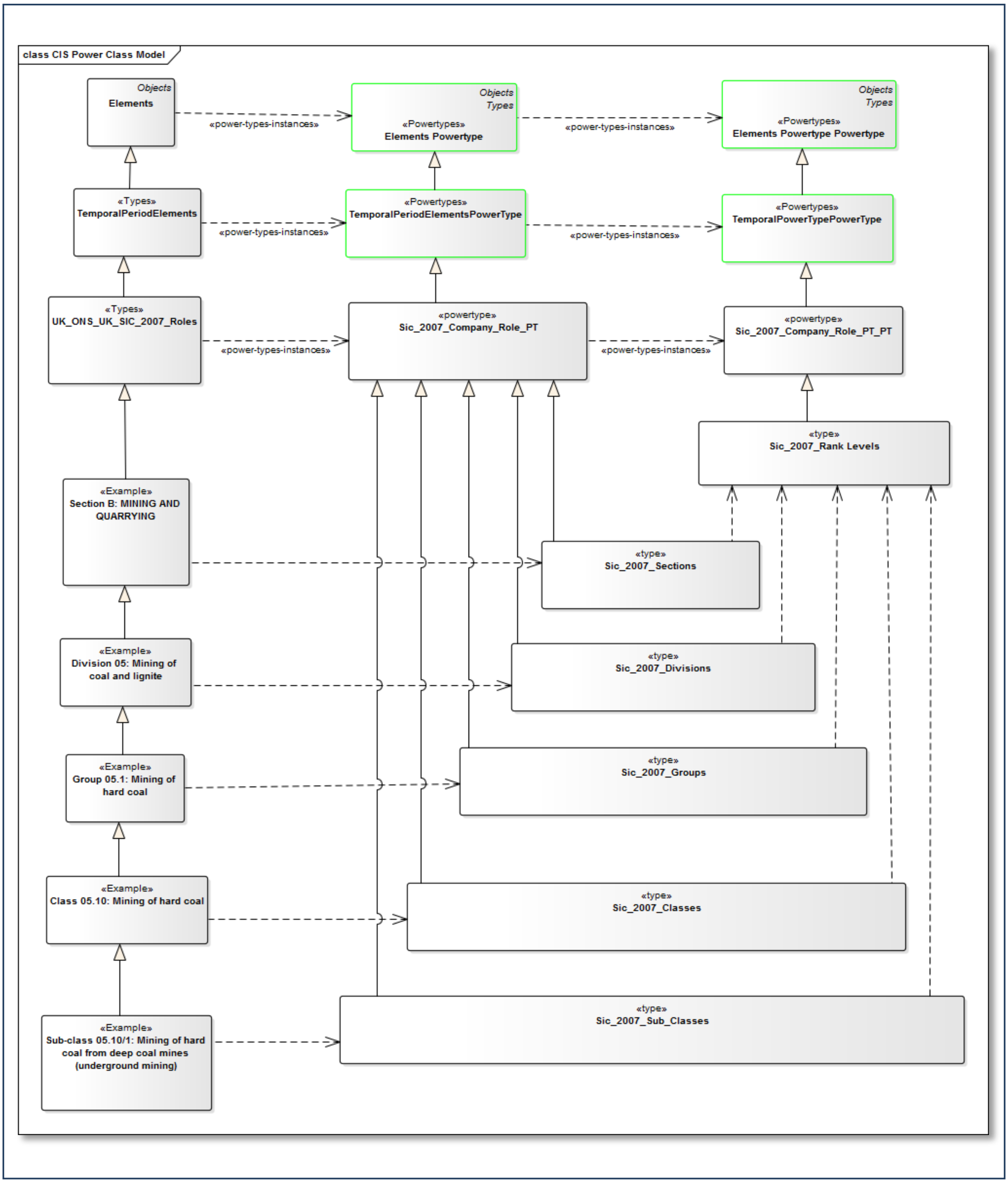


Figure 5.7: The SIC2007 Ontology

The domain ontology design draws on the knowledge collected within the 4D-SETL Worksheet, which details the ontological nature of the elements being modelled. The ontology is designed and further documented within an Enterprise Architect UML design tool ready for the next stage of the process which is the ETL of the BORO UML to the graph database.

The SIC 2007 depicted within figure 5.7 includes example class instances which visualises the relationships between the elements of the ontology and aids in the understanding of the domain ontology.

5.3.4 4D-SETL Domain Ontology Load

Semantic Integration

The semantic integration is achieved by matching the ‘Exemplar’ name elements within the different information sets. This is achieved over two stages which first load the objects generated by the BORO UML export, and then the ‘data’ which consists of the individual SIC types is loaded.

Ontology Stitching – Domain to Foundation

The first stage is the integration process involves ontologically ‘stitching’ the domain ontology objects into the BORO 4D foundation. This is undertaken in the BORO UML modelling stage which extends the BORO 4D foundation and upper ontology to encompass the modelled objects of the domain ontology.

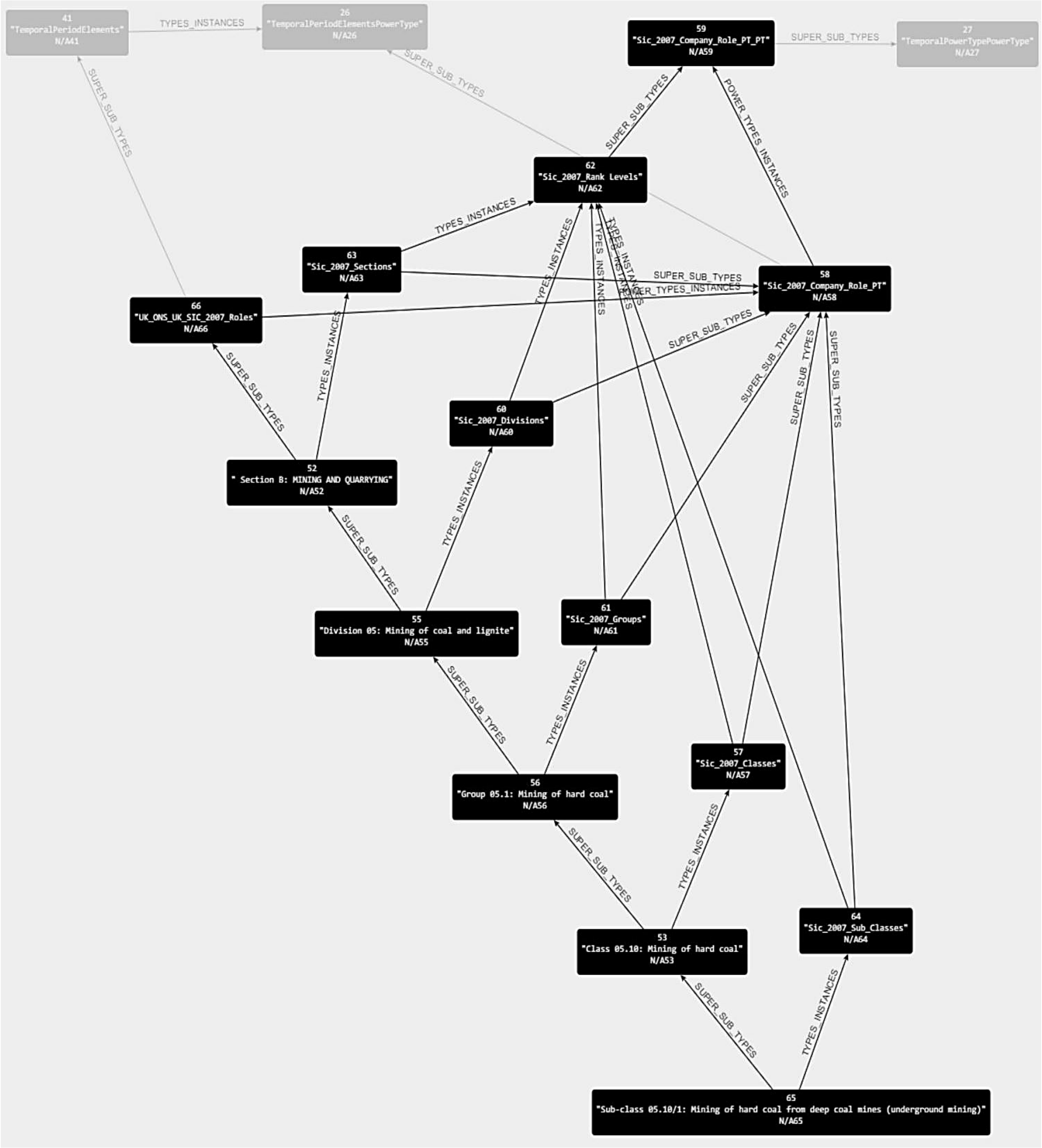


Figure 5.8: Neo4J Console Showing The SIC 2007 Ontology Within the Graph Database

Ontology Stitching – Domain Imported Types

The next stage is the integration process involves ontologically ‘stitching’ the new imported domain ontology Types to the SIC2007 domain ontology. This is undertaken using the new extract transform and load software developed in Java. Types are imported and ontologically stitched to the 4D-SETL graph model.

Each field from the source dataset in the case of the SIC 2007 represents ontology ‘Types’ (schema level knowledge) or model level knowledge which is commonly referred to as terminological knowledge (TBox). Each dataset row (or relation), provides the relationship with other field elements. Each object is ontologically transformed and ‘stitched’ into the domain ontology. The objects that represent Types are assigned unique identifiers during the load process. These object identifiers are bound directly to the modelled elements object as parameters. It is important to note that the objects position in the ontology structure defines its identity, not its name.

Integration Point Identification

During the load operation, the SIC 2007 index is queried to ascertain the node addresses of the objects to which the imported Types will be integrated. Therefore, a load order is established to ensure that the taxonomic rank (power types) exist within the warehouse before the individual objects representing classes are imported and integrated.

5.4 The Fourth Application of 4D-SETL - Companies House Dataset

The design science iteration now moves to the fourth application of the 4D-SETL which employs an open data from UK Companies House (2013).

5.4.1 Experimental Data Source Four: Companies House Dataset

<table>
<tr><th colspan="2">Dataset Documentation: SIC 2007 Description</th></tr>
<tr><td colspan="2">This is drawn from the Companies House Documentation</td></tr>
<tr><td>Publisher</td><td>Companies House</td></tr>
<tr><td>Date of publication</td><td>20-12-2014</td></tr>
<tr><td>Update frequency</td><td>Monthly (complete dataset (i.e.no changes change/delta file)</td></tr>
<tr><td>Scope</td><td>15 of the 52 fields available are selected for integration</td></tr>
<tr><td colspan="2"><b>Publisher:</b> The main functions of Companies House are the incorporation and dissolution of limited companies; the examination and storage of statutory company information required by the UK Companies Act of 2006 and to make this information available to the public. All limited companies in England, Wales, Northern Ireland and Scotland are registered at Companies House, an Executive Agency of the Department for Business, Innovation and Skills (Companies House, 2013). In November 2011, the Government announced that Companies House was to become part of a Public Data Group of trading funds. This group seeks to maximise the value of their data for long-term economic and social benefit, including through the release of data free of charge. Therefore, Companies House make this information available free of charge in the form of a series of downloadable data snapshots, updated monthly, which contain basic company data of live companies on the Companies house Register. The information made available includes the company number, address, status, incorporation date, account and annual return filing dates, SIC code, URI and basic information about mortgage charges. Within the data there are more than 3.5 million limited companies registered in the UK. The data is dynamic with more than 400,000 new companies incorporated each year (Companies House, 2013). As information is a copy of the official company register maintained by Companies House it can form a core dataset through which other datasets can be related and integrated.</td></tr>
<tr><td colspan="2"><b>Documentation</b>: As previously described, the first stage of the 4D-SETL processes is to document the dataset in the form of a worksheet containing each of the data source column headings together with descriptive information. The primary source of information in the form of a natural language model that describes the fundamental constructs contained within the dataset is from government legislation in the form of the Companies Act of 2006 (An Act of the House of Commons, 2006). This forms the primary source of the natural language description of what the fields represent. The secondary source of information is Companies House field descriptions, which provides outline information. The lifecycle of each element described is only fully apparent after detailed analysis of the law. For example, there is a provision within the law to change the form of the Companies Registration Office (CRO) Number. Other important information relating to name re-use, the restoration of companies to the register can also be gleaned from the Companies Act.</td></tr>
</table>

Table 5.5: Dataset Documentation: SIC 2007 Description

5.4.2 4D-SETL Transform

There are a number of patterns to apply in the case of the Companies House such as the name and located-at pattern. The Companies House dataset fields are described in Table 5.6. Note the definitive reference for this information is UK law, in the form of the Companies Act 2006 (An Act of the House of Commons, 2006).

Field Name	Description	Semantic Interpretation
CompanyName	There are six chapters of the companies act of 2006 devoted to company names. Company names are also covered in the chapters related to company dissolution and restoration. In summary, company names may be changed by the company itself or by an order of the registrar. The names of companies that are dissolved may be reused by another company. A company that is restored to the register may use its original name; however, if that name has been reused, a company will be restored with a name that is its registration number for a (temporary period).	A Company name follows the name pattern. The objects that it names is a state of a company (a named state). This is a spatio-temporal extension that is a temporal part of a company. If a company changes name then a new named state and name are required. The names of companies within the register are unique, however the same name 'string' may over time have been used to name several different companies. Therefore, if integrating historic information using the name as a means of establishing identity then the period over which the name was employed must be taken into consideration.
CompanyNumber	The registrar (of companies) shall allocate to every company a number, which shall be known as the company's registered number. The registrar may on adopting a new form of registered number make such changes of existing registered numbers as appear necessary. A change of a company's registered number has effect from the date on which the company is notified by the registrar of the change. For a period of three years beginning with that date, any requirement to disclose the company's registered number imposed by regulations under section 82 or section 1051 (trading disclosures) is satisfied by the use of either the old number or the new.	The company Registration Number is a unique name which is allocated by the registrar of companies. The registrar may change the form or format of this registration number; therefore, the name follows the pattern of naming a state of a company. There will be a period that a company can be identified by either the old or new company number. This state lasts for a period of three years following a registration number name change.

Table 5.6: The Companies House dataset fields (An Act of the House of Commons, 2006)

Field Name	Description	Semantic Interpretation
RegAddress.PostCode	location (unique within geographic name space) Related by tuple location state.	This field is used to associate a company with a physical location which has been defined within the geographic ontology
CompanyCategory	Legal State - e.g Private Limited Company - name of a legal status state	This field is used to associate a company a Type within a taxonomy ontology (which is defined within the company ontology
CompanyStatus	The trading status of a company	A series of Types
DissolutionDate -	Name of an event /add to graph and index	The dissolution event fro the company entity NB this is not final companies may be restored to the register by a legal process
IncorporationDate	Name of an Event /Add to graph and index	The creation event for the company entity
SICCode.SicText_1	Name of a role state	Used to integrate with the Sic 2007 ontology
SICCode.SicText_2 -	Name of a role state	Used to integrate with the Sic 2007 ontology
SICCode.SicText_3 -	Name of a role state	Used to integrate with the Sic 2007 ontology
SICCode.SicText_4 -	Name of a role state	Used to integrate with the Sic 2007 ontology
URI-name	Name	Name pattern
PreviousName_1.CONDATE	The date the previous company name ceased to be used.	Name pattern named state dissolution event.
PreviousName_1.CompanyName	The previous company name	Name pattern – named state

Table 5.6: Continued. The Companies House dataset fields

4D-SETL Ontology Design

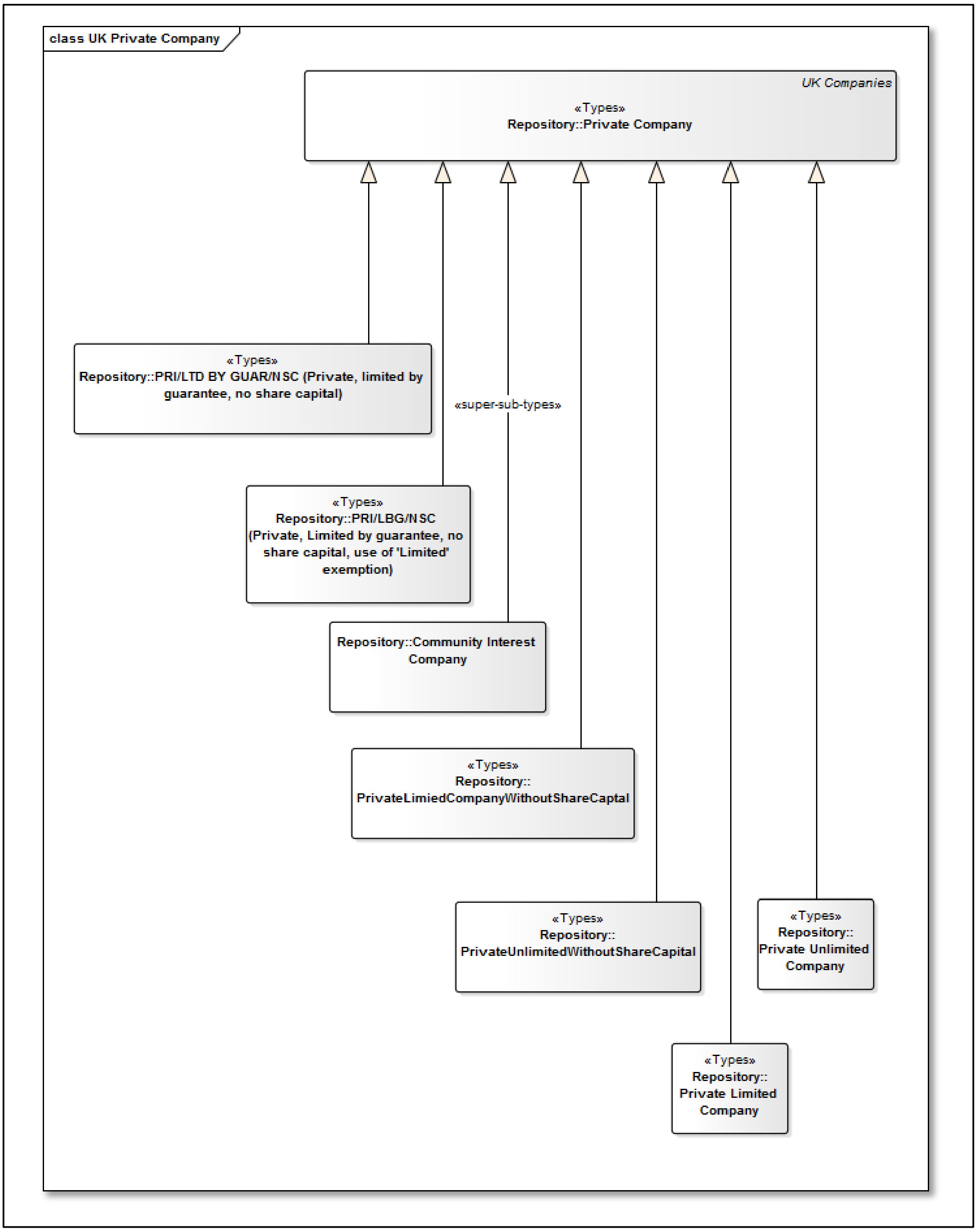

class UK Private Company
UK Companies
«Types»
Repository::Private Company
«Types»
Repository::PRI/LTD BY GUAR/NSC (Private, limited by guarantee, no share capital)
«super-sub-types»
«Types»
Repository::PRI/LBG/NSC (Private, Limited by guarantee, no share capital, use of 'Limited' exemption)
Repository::Community Interest Company
«Types»
Repository::PrivateLimiedCompanyWithoutShareCaptal
«Types»
Repository::PrivateUnlimitedWithoutShareCapital
«Types»
Repository::Private Unlimited Company
«Types»
Repository::Private Limited Company

Figure 5.9: Company Ontology Legal Status - Taxonomy Pattern

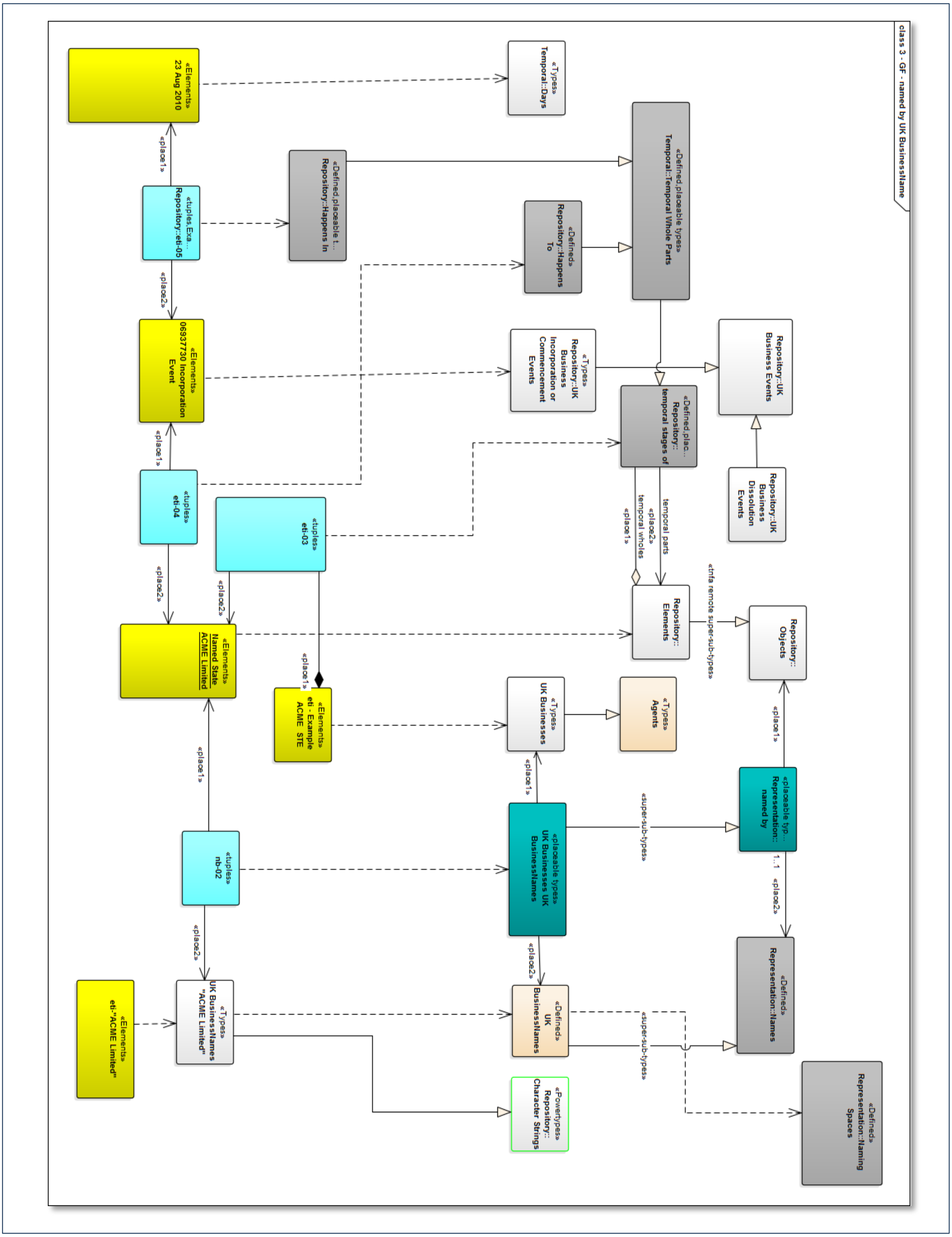


Figure 5.10: Part of the Company Ontology - Named State Pattern

The domain ontology design draws on the knowledge collected within the 4D-SETL Worksheet which details the ontological nature of the elements being modelled. The ontology is designed and further documented within an Enterprise Architect UML design and automatically transcribed to the graph database.

Load process

The data load and integration process extracts and integrates each element form the bulk dataset provide by Companies House is now described. Firstly, each row is extracted and each individual field is transformed and integrated. The basic operation is as follows:

i) Insert the contents of any field that contains an exemplar name (integration point such as the company registration number) into the Companies House Index.

ii) Lookup integration points in the foundation and Temporal, Geographic and SIC 2007 domain ontologies.

iii) Apply the relevant pattern.

5.5 Evaluation

5.5.1 Technical Experiment

The main subject of the technical experiment is the ability of the 4D-SETL framework to integrate and process larger datasets and thus confirm the viability of the framework to scale. During this technical experiment, further experiments will be undertaken to assess the ability of the 4D-SETL to bulk load large datasets and to confirm the warehouse implementation can function to extract data through graph traversal. Again the physical system depicted in Figure 4.16 will be employed to support the 4D-SETL experiment.

5.5.2 Research Output Artefacts

The second iteration of the design science design and implement research activities produced the following artefacts:

a) The tool-chain through which the 4D-SETL framework could be applied;

b) Designs in the form of worksheets, BORO UML ontologies, and software code; and

c) A prototype graph database system instantiation.

The evaluation of these artefacts is the third phase of the iteration. Guidance as to the evaluation method applicable to each of these artefacts is drawn from the design science methodology provided by March and Smith (1995) who assert that the evaluation activity must develop metrics through which to compare the performance of the artefacts produced and help to define whether the research aims have been achieved.

5.5.3 Tool-chain

The processes by which individual data elements are added to the graph in the form of subgraph patterns was developed during this stage; however, it was time consuming to configure and error prone. During the next iteration this support element needs to be improved and therefore during the next stage the creation of these patterns will be abstracted in the form an Application Programmers Interface (API) which will simplify this process.

5.5.4 Technical experiment

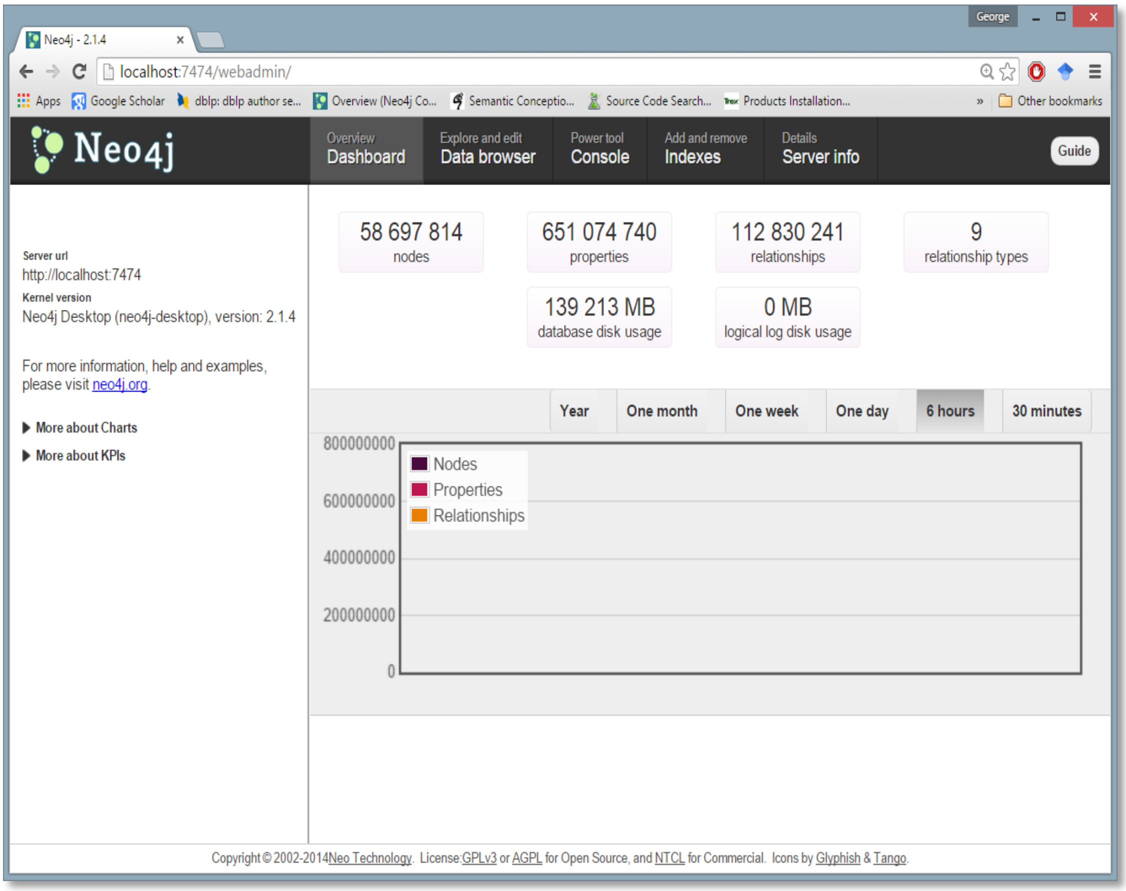
The prototype has proven that the 4D-SETL framework can be applied to implementing a practical system that is capable of importing and structuring data in a manner that reflects the Object Paradigm. The 4D-SETL framework has been applied and tested by modelling and implementing higher order taxonomic ranks structures in the form of SIC2007 dataset with the Companies House dataset representing a bulk data load requires integration to calendar, geographic, SIC 2007 and the structuring of company Type hierarchies.

Bulk Load

The Companies House Data represents the first complex dataset bulk warehouse data load. Part one of the five part dataset contains eight hundred and fifty thousand company records. Each record is extracted transformed and loaded to the graph database as patterns that reflect the model structure defined during the design stage. The load process consists of locating the domain ontology Types that the imported element is a member instance. Following integration with the domain ontology, the imported object is integrated with other instance level objects and Types within the warehouse system; for example a company has a spatio-temporal extent that is a member instance of a SIC 2007 Type.

There are extensive look-up operations required to perform the integration operation. This involves finding the exemplar names that exist within the warehouse and the Elements to which they refer. This process is slow and compute-intensive. However, bulk load operation should not have to be performed on a regular basis as the ontologies stored within the warehouse can be added to without recourse to the unload, schema update, reload operation that is required with a RDBMS based warehouse systems.

The first part of the Companies House dataset consisting of eight hundred and fifty thousand company records was loaded to the warehouse. This process took approximately 8.7 hours to complete.



Server url
http://localhost:7474
Kernel version
Neo4j Desktop (neo4j-desktop), version: 2.1.4

For more information, help and examples, please visit neo4j.org.

58 697 814 nodes

651 074 740 properties

112 830 241 relationships

9 relationship types

139 213 MB database disk usage

0 MB logical log disk usage

Copyright © 2002-2014Neo Technology. License:GPLv3 or AGPL for Open Source, and NTCL for Commercial. Icons by Glyphish & Tango.

Figure 5.11: Warehouse Neo4J Loaded – 58 Million Nodes 112 Million Edges

Path Traversal

The experiment finds a geographic location with an index look-up then traverses the graph to find all business entities that have a registered office at the location. The first pass of the code was executed and returned results in 1577 milliseconds however, when re-run the cache effect was evident as the second path traversal was undertaken in 60 milliseconds. The path traversal results are detailed in Appendix B. The following is pseudo code provides details of the Graph DB query employed to generate the results.

```
1. Output ("Start new path traversal ")
2. StartTimeE=TimeNow()
3. Result [Array of Graph Node Structures ] <-  Run Graph Query
4. Cypher: Use Geographic Domain (namespace) to find the start node for the traversal
5. (start a=node:geographic_name_index, where ontology_unique_name =  "W5 2NP"
6. Traverse Graph to find physical location that is named
7. Traverse graph to match all companies that are related by the located-at relationship
8. Set EndTime = TimeNow()
9. Output Execution Time =  EndTime-StartTime
10 for each returned node:
11.fetch and print Company Name
```

The execution of the path traversal application resulted in a node listing which is detailed in Appendix B. The results were validated against the source data set to ensure the accuracy of data returned.

Technical experiment has been used to assess the effectiveness of the framework to function at scale and thus address the weakness describe within Table 1.1, Item 8.

5.5.5 Illustrative scenario

SIC 2007

The SIC 2007 has been loaded to the warehouse as a graph in its natural structural form, of taxonomic ranks and classes, therefore enabling navigation of the graph to search for related companies. For example, it is be possible to locate a company with a very specific subclass SIC 2007, then to traverse to the next rank node to find companies using a broader classification.

The SIC 2007 Taxonomic ranks have been modelled in their original structure, demonstrating the effectiveness of the framework to model the abstractions of reality 'as-is' without simplification (Table 1.1, Issue 3). In addition the SIC 2007 ETL has also demonstrated the effectiveness of the framework to dispense with the model strata and distortion (Table 1.1, Issue 2). Within the framework there is no artificial partition between those model elements that are classed and Type or those classed as instances. Thus the ETL of the SIC 2007 data consisted entirely of Types.

A company may undertake one or many different activities over its lifetime. The 4D-SETL can effectively model these changes and integrate them with the SIC 2007 classification system. Thus a company can have an initial state of being a paper manufacturer, then at a later date, have a state of being a mobile phone manufacturer. These states may be contiguous or overlap. Thus, the model can capture both the structural and dynamic aspects of how a company changes over time. Therefore, the effectiveness of the framework to address the weakness related of having different models for static and dynamic elements (Table 1.1, issues 5 and 7) has been described.

Query Processing

The 4D-SETL warehouse being in graph form is more suitable for discovering relationships within the data, rather than processing aggregated data. For example it is possible to discover all relationships that exist between two companies such as shared directors, location, roles (SIC 2007 activities) using an algorithm from the Neo4J library (all simple/shortest paths between two nodes) such as that can find all available paths between the two nodes representing each company. Other algorithms provide:

a) The cheapest path between two nodes using the A* algorithm (Hart, Nilsson and Raphael, 1968).

b) The cheapest path between two nodes using the Dijkstra algorithm (Dijkstra, 1959).

c) All simple paths of a certain length between two nodes.

There are also a number of additional algorithms such as Page Rank (Brin and Page, 1998) available from third party open software providers. As the current implementation does not employ weighted edges; the algorithms a) thought to c) are not be applicable. However, the application of such algorithms is an area of that would benefit from further investigation during future research.

5.6 Summary and Conclusion

This iteration has provided further knowledge related to the subject of semantic integration through the instantiation of system based on the 4D-SETL framework. The utility of the approach has been demonstrated through the resultant system ability to model and instantiate a number of complex ontological structures, such as higher order taxonomic ranks. In addition, the resultant warehouse has been tested at some level of scale by importing the Companies House dataset. The accuracy of the data returned by a querying the Graph Database has also been confirmed.

5.6.1 Framework Improvement

During this second iteration the design science cycle, the 4D-SETL framework has been improved and is incorporates within two discrete Extract, Transform and Load (ETL) Stages. Firstly the Schema level ontology is processed, transformed and loaded. Then the second stage ETL – also consisting of an Semantic Extract, Transform and Load process is used to process the individual records of the source data set to load the instance level data to the warehouse.

5.6.2 Ontology Data Load Manager API

During the second iteration of the design science cycle, in order to establish the instance level data the 4D-SETL graph patterns that represent the BORO foundation were manually coded. This was time consuming and therefore during the next iteration this new functionality will be added to provide an Application Programmers Interface (API) which will present the programmer with an interface through which to instantiate a particular ontological patterns and provide the required parameters (fields). This also improves consistency and completeness of the graph representation of the BORO domain model and therefore the correctness of the model (Zowghi and Gervasi, 2003). To clarify, here we are considering the correctness of the graph DB rendering of the BORO UML not the correctness of the business domain to which the BORO UML represent (stands-in for).

CHAPTER 6. DESIGN SCIENCE THIRD ITERATION – FURTHER 4D-SETL IMPROVEMENT AND APPLICATION

6.1 Introduction

This chapter describes the execution of the third and final iteration of the design science design-build-evaluate cycle. This iteration continues to explore the problem of semantic integration by employing the 4D-SETL framework to integrate the last experimental dataset. Again, as the semantic integration process progresses the dataset integration becomes more complex; therefore, enabling the 4D-SETL framework to be further tested. In tandem with these activities a number of enhancements that were identified during the second cycle are implemented to improve the overall tool-chain support for the 4D-SETL framework.

6.2 Design Science Research -Iteration Three

This cycle builds on the experience and knowledge created during the previous iteration. The 4D-SETL framework is again applied to the integration of the final experimental dataset with the resultant of the combined data being stored within the prototype warehouse system. The evaluation is of the artefacts produced is achieved by technical experiment to confirm the scalability of the framework and its ability to function as a warehouse capable of returning accurate results from information queries.

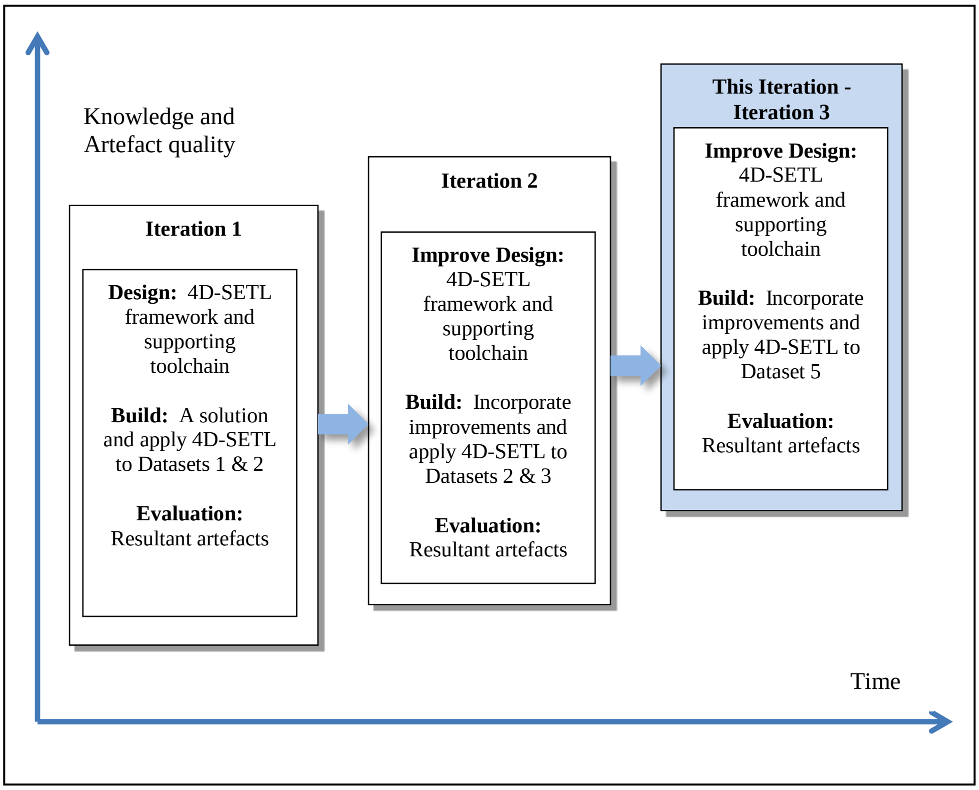

Knowledge and Artefact quality
Iteration 1
Design: 4D-SETL framework and supporting toolchain
Build: A solution and apply 4D-SETL to Datasets 1 & 2
Evaluation: Resultant artefacts
Iteration 2
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Datasets 2 & 3
Evaluation: Resultant artefacts
This Iteration - Iteration 3
Improve Design: 4D-SETL framework and supporting toolchain
Build: Incorporate improvements and apply 4D-SETL to Dataset 5
Evaluation: Resultant artefacts
Time

Figure 6.1: Research Design-Build-Evaluate Cycle Interaction Three

6.2.1 Aim and Objectives of Iteration Three

So far, the 4D-SETL framework has been applied (during the previous 2 Design Science interactions) to integrate four datasets of various scale and complexity and the resultant artefacts have been evaluated via technical experiment and illustrative scenario. The aim of this third iteration is to continue the evaluation of the effectiveness of the 4D-SETL by applying it to integrate the final (and largest) dataset. Technical experiment is employed to confirm the scalability and the ability of the framework, thorough employing the query processing capabilities of the graph database (path traversals) to retrieve accurate results. Illustrative scenario is employed to describe the effectiveness of the framework to address problems related to semantic data integration. Finally the ontology developed by the 4D-SETL process is evaluated. This aim is met through the activities undertaken in Table 6.1.

Stage	Activity	Objective	Section Reference
Design	1	Improve the execution and application of the 4D-SETL framework by improving the design of the tool chain to address the deficiencies identified during the second DS iteration.	6.3
Build	2	The application of the 4D-SETL framework to develop and load Directors domain ontology	6.4.
	3	The application of the 4D-SETL framework to develop and load Directors instance level ontology)	6.4.5
Evaluation:	4	Technical experiment testing of the warehouse artefacts Illustrative scenario: describing the artefact qualities and advantages. Ontology Evaluation.	6.5

Table 6.1: Stages and Activities and Within this Iteration

6.3 4D-SETL Improvements Ontology Data Load Manager (ODLM)

During the previous iteration of the design science cycle, establishing the integration with the domain ontology and other individual elements within the warehouse involved manually coding the relationships to establish the basic ontological patterns. This code has been abstracted to an Applications Programmer Interface (API), which is now used to establish the ontological patterns that mirror the patterns of the domain ontology and foundation.

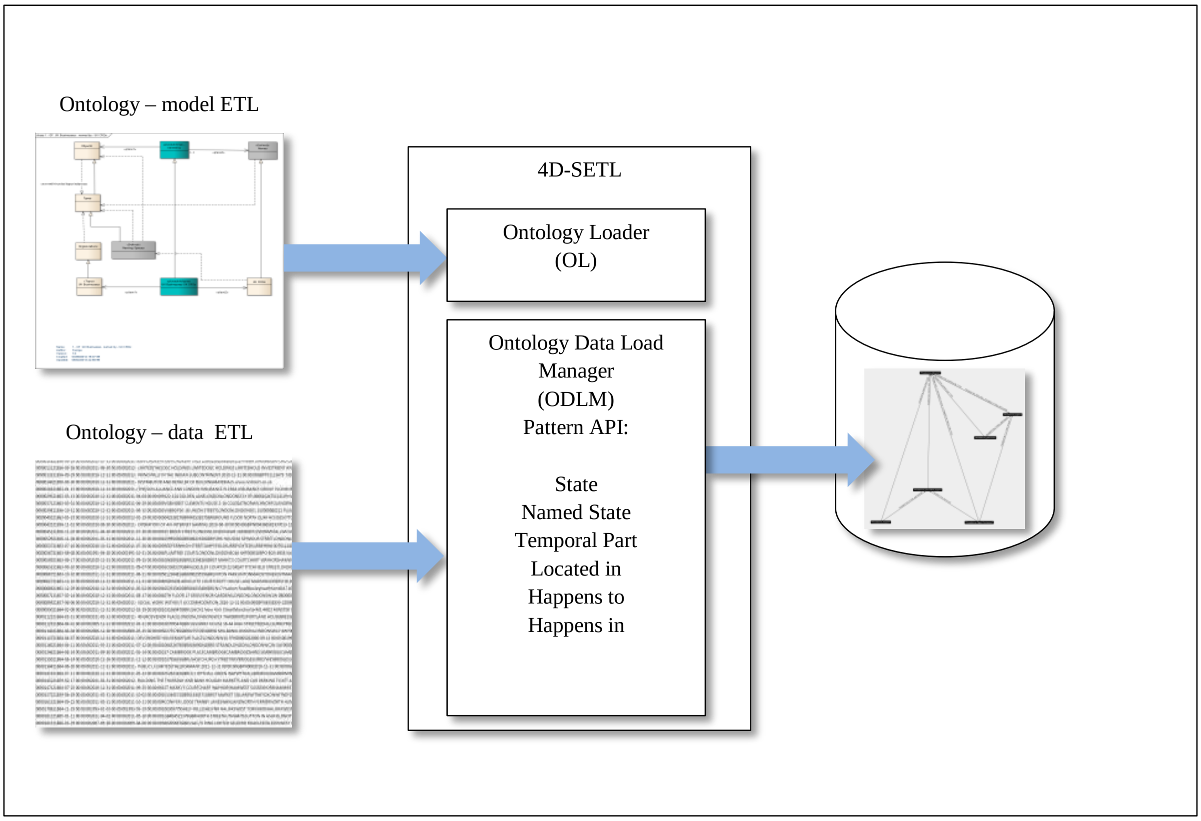


Figure 6:2: Ontology Data Load Manager

6.4 The Fifth Application of 4D-SETL – Directors Dataset Integration

The DS iteration now moves to the fifth application of the 4D-SETL framework. An overview of the activities undertaken and the resultant artefacts is presented in Figure 6.3.

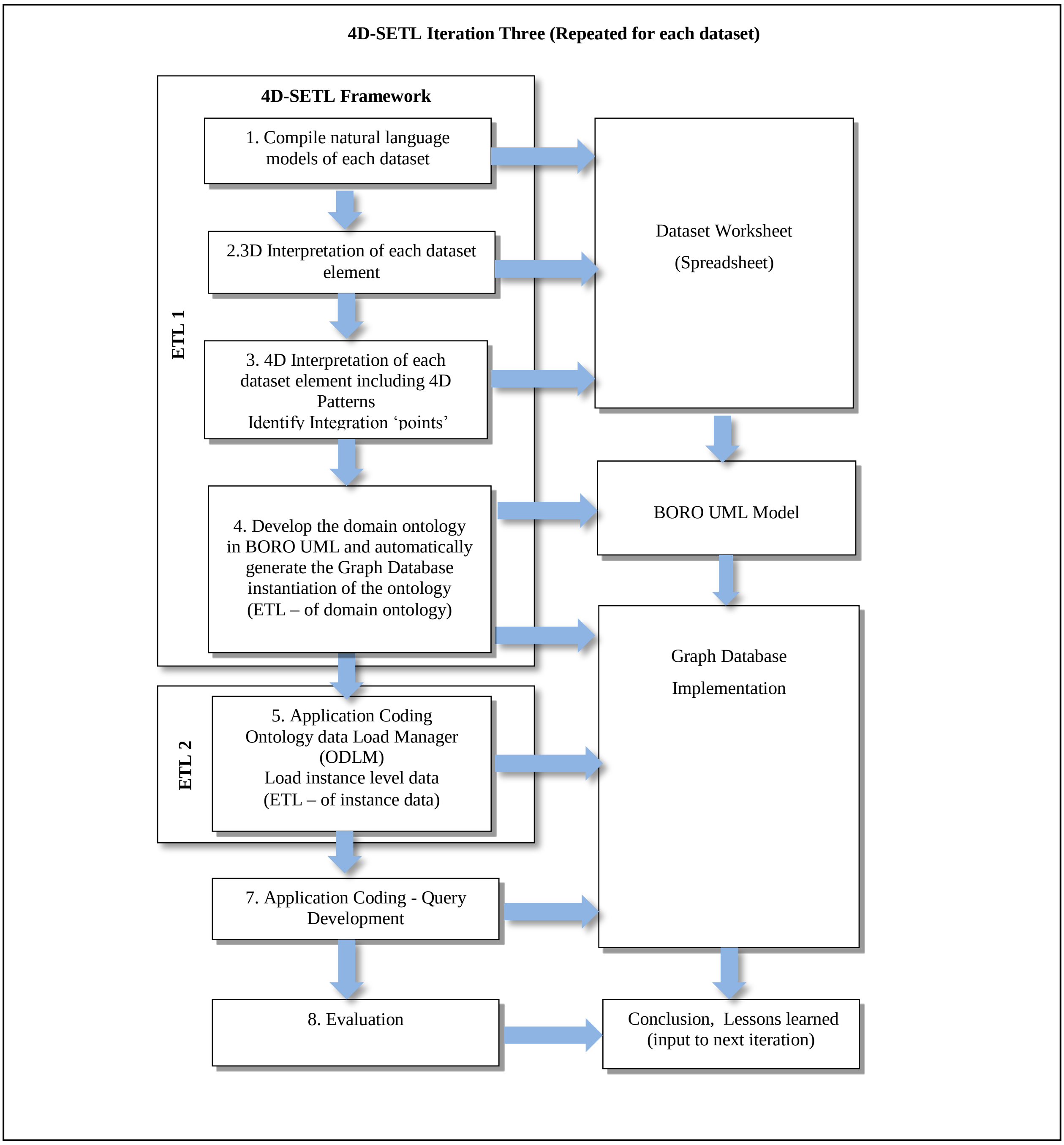

4D-SETL Iteration Three (Repeated for each dataset)
4D-SETL Framework
ETL 1
1. Compile natural language models of each dataset
2.3D Interpretation of each dataset element
3. 4D Interpretation of each dataset element including 4D Patterns
Identify Integration 'points'
4. Develop the domain ontology in BORO UML and automatically generate the Graph Database instantiation of the ontology (ETL – of domain ontology)
ETL 2
5. Application Coding Ontology data Load Manager (ODLM) Load instance level data (ETL – of instance data)
7. Application Coding - Query Development
8. Evaluation
Dataset Worksheet (Spreadsheet)
BORO UML Model
Graph Database Implementation
Conclusion, Lessons learned (input to next iteration)

Figure 6.3: Overview of the Third DS Iteration and the Resultant Artefacts

As depicted, the process now includes an Ontology Data Load Manager and API that is employed during the second stage ETL (bulk dataset load) to simplify the configuration of this process.

6.4.1 Experimental Data Source Five: UK Company Officers

Dataset Documentation: Directors Dataset	
This is drawn from the Companies House Documentation	
Publisher	Companies House
Date of publication	20-12-2012
Update frequency	Monthly (complete dataset)
Size	12.5 Million records
Provenance	This has been derived from a Companies House register for statuary reporting company officer details and therefore can be deemed to be a high quality reference dataset.
Scope	Of the thirty fields within the dataset sixteen have been selected to be in scope for the integration process.
Publisher: Company officer data is released by Companies House in the form of files that provide a copy of director, secretary and LLP member details as they are held in the Companies House Directors Register at the given point in time when the snapshot copy is made. Typically these details are provided for each live company and Limited Liability Partnership (LLP). Data comprises information drawn from prescribed forms submitted by companies and LLPs. The worksheet also documents the provenance of the dataset, publisher, publication date, update frequency in the case of a directors' information.	
Documentation Each person within the Companies House register acting as a company officer is allocated a unique Person Number. A change to an individual's Usual Residential Address (URA) or name (but not Service Address) will generally result in a new increment of the Person Number being created.	

Table 6.2: Directors Dataset Documentation

6.4.2 4D-SETL – Extraction Stage

As with the previous iterations (endurant) semantics (3D) are extracted from the dataset. The table represents the Directors Type (or Class). Each row represents an instance the relationship between a Company instance and a Company Officer instance. Therefore, when an Officer has multiple appointments they may appear in the dataset multiple times. A record (row) may relate a company (companies as legal or non-human persons can act as the officers of other companies) to another company. Columns are interpreted as Attribute Types and individual fields as Attribute Instances.

6.4.3 4D-SETL – Translation Stage

Name pattern

The interpretation and semantic extraction process starts with the named states that are a temporal part of a person. There are two such names – the person's name (family, first, middle names) and their person-number. Both of these names can be changed, whilst still referring to the same person. If the person is a company (a non-human legal person) then the name will be that of the company (which can also change). Any change in these names will result in the creation of a new named state.

Events

There are three creation events: director appointment date, secretary appointment date and date of birth (for a human officer) and two dissolution events: director and secretary resignation dates (date of death is not included in the dataset). These events are a spatio-temporal part of a specific day and of the spatio-temporal part of the STE to which the event happens.

Temporal States

There are two spatio-temporal states; secretary and director. These states are parts of both the company and the person (human or legal) that is serving in the role.

6.4.4 Semantic Integration Points

Person to Company

Within the Director data set, there is a record (tuple) that associates the Company Registration Number and Person Number relationship. This will act as the integration point between a person (human) STE and the company STE of which they are a temporal part. A director STE and secretary STE can exist concurrently – being the same person fulfilling the role.

Company to Company

Within the Director data set, there is a record (tuple) that associates the Company Registration Number and the Company Name. This acts as the integration point between a person (legal non-human) STE and the company STE of which they are a temporal part. A director STE and secretary STE can exist concurrently – being the same person (legal non-human) fulfilling the role.

Company to Location

Within the Director data set there is a source record (tuple) that associates a Person to geographic location (postcode). Thus if the exemplar postcode name can be matched with the warehouse geographic index – then the person’s physical location can be integrated and the ‘located-at’ pattern can be applied. No attempt has been made to assert a location for non-UK addresses. However, as the complete postcode database has been loaded to the warehouse all UK addresses should result in a match.

Company Events to Calendar

If an event within the Director data set can matched with the exemplar calendar name with the temporal name index – then the events temporal location can be fixed and thus use the happens-to pattern applied.

Therefore, following the translation process a 4D model has been developed that consists of 4D spatio-temporal extents that are names (in the form of a physical ‘strings’) and STE objects that represent a director or secretary. These in turn have temporal parts that are also temporal parts of a company. Spatio-temporal extent objects have also been identified with a temporal extent of zero thickness that represents creation and dissolution events of these states.

Field	Description	3D Interpretation	4D
Registration Number	Company registration Number of which the person (human or company) is an officer	Name attribute	A name STE
Unique Reference Number	Companies House Identifier	Name attribute	A name STE
company_indicator	Y= Indicates that the officer is a company	Attribute	Indicates that another company (legal person) rather than a human is a temporal part of the company STE instance.
director_status	Indicates that the officer is a Director C= Current P= Past	Attribute	Indicates that the temporal part of a Person or Company STE is a Director.
director_appointment_date	Date	Timestamp	Creation event element (3D)
director_resignation_date	Date	Timestamp	Dissolution event element STE (3D)
company_secretary_status	Indicates that the officer is a secretary	Timestamp	Indicates that the Person or Company has a temporal part that is company secretary.
company_secretary_appointment_date	Date	Timestamp	Creation event of the secretary temporal part
company_secretary_resignation_date	Date	Timestamp	Dissolution event of the secretary temporal part
Surname	Name	Name attribute	A name STE
christian_name	Name	Name attribute	A name STE
middle_names	Name	Name attribute	A name STE
company_name	Provides a company name if the officer is a company	Name attribute	A name STE
Nationality	Country	Name attribute	Location of the birth or naturalisation creation event. A nationality temporal part of a person STE
Postcode	Address code	Location (Endurant)	A name STE Integration Point – Exemplar Location name
date_of_birth	Date Only applies to a human officer	Timestamp	Creation event STE

Table 6.3: Elements and Relationships

The foundation and patterns can now be extended to develop the Director ontology. The Directors domain ontology is designed as before using Enterprise Architecture and the BORO UML profile semantics. During the design process, it is useful to insert example instances to model how such instances relate to the ontology structure. Figure 6.4 depicts part of the Director ontology showing the relationship between director and secretary states and the company and human agent that they are a temporal part. In this example, the same human agent has been both a secretary and a director of the same company. These states temporal extents may overlap, i.e. the appointment and resignation dates cover some of the same period (this is not depicted). Figure 6.5 depicts part of the Director ontology showing the events that can happen to create and dissolve Director/Secretary states.

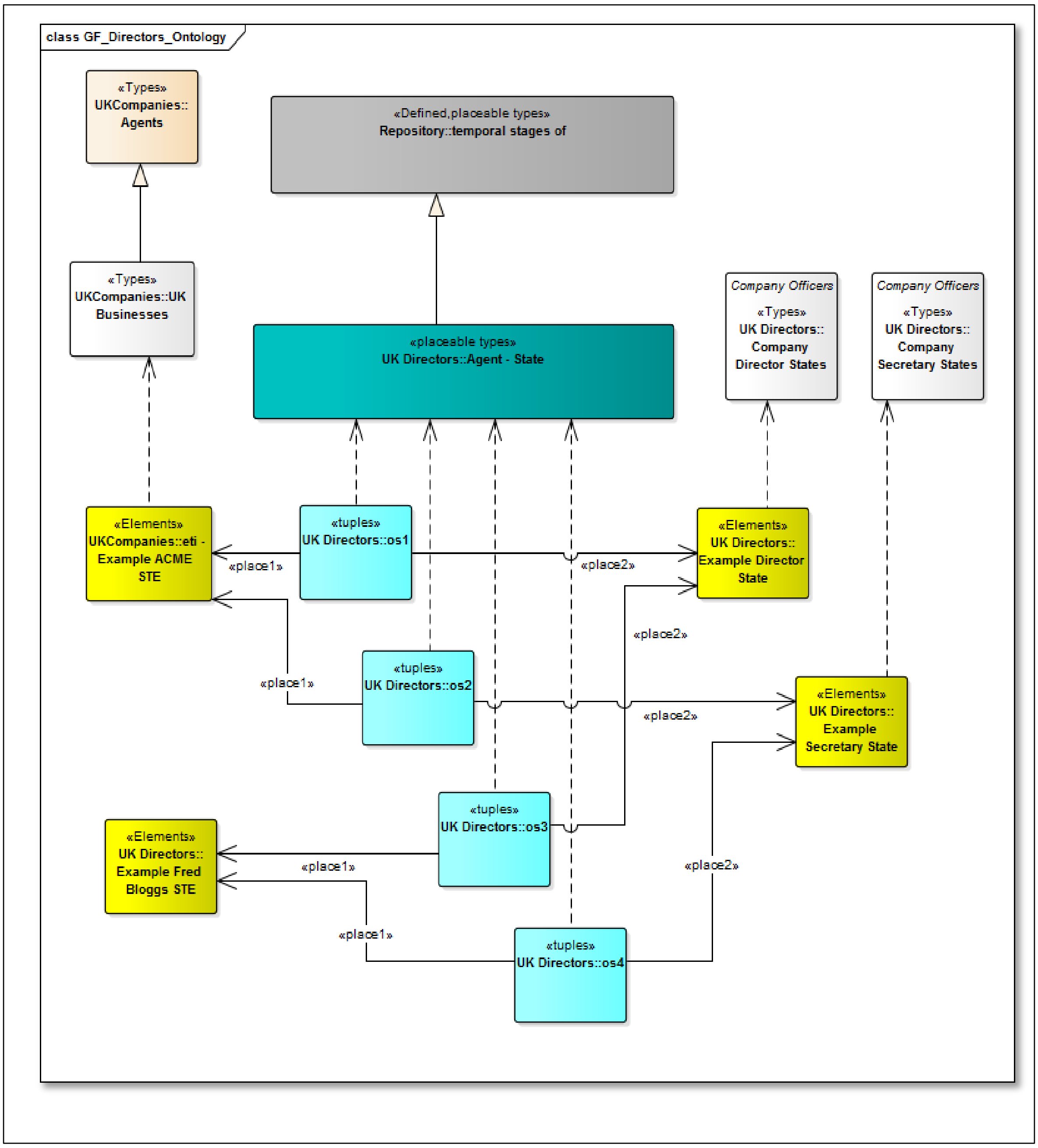


Figure 6.4: BORO UML Director Ontology

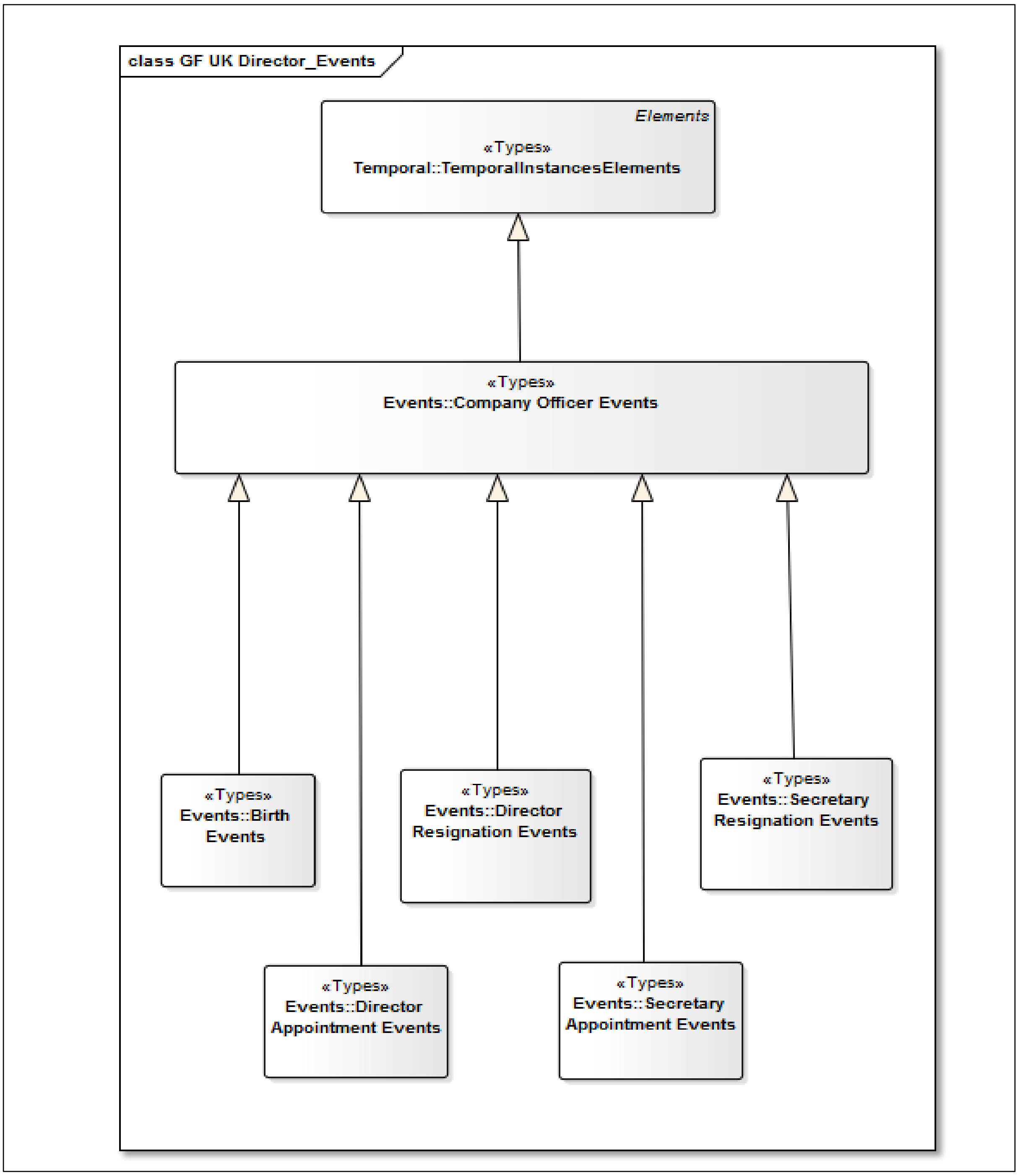

class GF UK Director_Events
Elements
«Types»
Temporal::TemporalInstancesElements
«Types»
Events::Company Officer Events
«Types»
Events::Birth Events
«Types»
Events::Director Resignation Events
«Types»
Events::Secretary Resignation Events
«Types»
Events::Director Appointment Events
«Types»
Events::Secretary Appointment Events

Figure 6.5: BORO UML Director Event Types

6.4.5 4D-SETL Load

The load process involves the use of the 4D-SETL Ontology Loader and the Ontology Data Load Manager (ODLM) to load and process firstly the ontology and then the bulk dataset. This involves processing each of the dataset records to extract the data related to each field and add it to the graph database. The operation utilises the graph database index system to firstly insert any exemplar names that will be used to locate or integrate a particular director element. The next stage involves using the Application Programmers Interface (API) to insert each director subgraphs into the graph database. The API simplifies this process by performing the various name space indices lookup (integration points) and to generate the relevant patterns (nodes and edges) as required.

The load and integration process now moves to the identification of the integration points that will be established between the director dataset and the elements already loaded within the warehouse - see table 6.2.

Objects	Integration Points
Director Appointment Event	Temporal Part of a day and of a Director State. Is an instance of the appointment event Type.
Director Resignation Event	Temporal Part of a day and of a Director State. Is an instance of the resignation event Type.
Secretary Appointment Event	Temporal Part of a day and of a Secretary State. Is an instance of the appointment event Type.
Secretary Resignation Event	Temporal Part of a day and of a Secretary State
Person Birth Event	Temporal Part of a day and of a Person State
Person	A STE – the temporal whole of a person
Director State	A temporal part of a person and a company
Secretary State	A temporal part of a person and a company
Location State	A temporal part of a person and a physical location
Temporal Whole Parts	Establishing temporal whole-part relationships between all temporal parts and the STE whole elements (Company and Director).
Director Name State (First, Middle and Family)	Name Pattern – applied as a state (as the name can change)
Director Unique Reference Name State	Name Pattern – applied as a state (as the name can change)

Table 6.4: Director Dataset Integration Points

6.5 Evaluation

The final iteration of the 4D-SETL design-build stage has been completed. The following section describes the artefacts produced and the final evaluation of 4D-SETL.

6.5.1 Research Output Artefacts

The third iteration of the design science design-build-evaluate stage of this research produced the following artefacts:

a) An improved 4D-SETL framework tool-chain.

b) Designs in the form of worksheets, BORO UML ontologies, and software code.

c) A prototype graph database populated with the fifth domain ontology (Directors); that includes a large collection of instance level data.

The evaluation of these artefacts is the third stage of the iteration. Again, as with the previous two iterations, guidance to the evaluation method to apply for these artefacts is drawn from the design science methodology provided by March and Smith (1995).

Therefore, the objective of this evaluation is to assess the effectiveness of the artefacts produced and to judge whether the aim of the research has been met or not. This is achieved by referencing the weaknesses identified through reference to literature and to judge if the artefacts produce solve, or partially solve the problems related to semantic integration.

6.5.2 Technical Experiment

Bulk Load Experiment

The Director dataset consists of approximately 12.5 million records that is the fifth bulk data load to the warehouse. Each of these records represents a relationship between a company officer (a director and or secretary) and a company. Each record also details appointment and resignation events and location (address) information. As with the previous datasets, the 4D-SETL is applied in two stages – firstly the domain ontology (terminological entities) are subject to the semantic ETL which results in an additional graph model being loaded to the graph database. Then, the bulk data is subject to the second ETL process, with each record being extracted transformed and loaded to the graph database as patterns that reflect the model structure defined during the design of the domain ontology. The load process consists of locating the domain ontology Types that the imported individual element is a member instance. Following integration with the domain ontology, the imported object is integrated with other instance level objects within the warehouse system; for example a director role is a spatio-temporal part (state) of both the person (human or legal) and the company that the person is a director or secretary. This results in extensive look-up operations being required to perform the integration operation and involves finding the exemplar names that exist within the warehouse and the elements to which they refer. This process is slow and compute-intensive. However, bulk load operation should not have to be performed on a regular basis as the ontologies stored within the warehouse can be added to without recourse to the; unload, schema update and reload operation that is often required by RDBMS based warehouse systems. The full data load took approximately one hundred hours complete.

Information Accuracy, Identity

A previous experiment has confirmed the warehouse instantiation can operate at scale. This experiment confirms the accuracy of information retrieval. This is achieved through a graph traversal that starts at a node that represents a company spatio-temporal extent, then to traverse the graph to return all of the director/secretary states that are parts of that particular company. Thus, having found all nodes that represent a director/secretary state, the traversal continues from each of these nodes to locate all nodes that represent a person (human or legal) that shares the director/secretary state. This graph traversal thus identifies the overlap between, for example, the director state of the person ‘Fred Blogs’ and the ‘Managing

Director of ACME Limited'. As they share the same temporal part – they are the same physical thing. This is the same relationship as described in Chapter two which used the example of the President of the USA and Barak Obama, i.e., a temporal part that overlaps two spatio-temporal extents is the same thing – establishing identity.

Listing of the traversal experiment (Companies House issued Person Numbers) can be found in Appendix C. To validate the results of the traversal, the output of the graph traversal were compared with the input source data to confirm the accuracy of the results.

Thus the framework has addressed the problem of establishing identity and identity over time, as described in Table 1.1, Issue 7.

6.5.3 Illustrative Scenario

Although technological implementation of the ontology is possible within a RDBMS, the development of such tabular representations normally occurs with 'workarounds' that could have a negative impact on ontological alignment. However, in this research, because a graph database was employed, no restrictions were imposed on expressivity or structure and the graph model (ontologies) more closely reflected the nature of the reality rather than the structures imposed by the development tools or execution environment. Thus, employing the 4D-SETL framework enabled the foundational, domain and instance level ontological models to be represented by a single graph that was implemented 'as-is' within the run-time environment. This demonstrates the ability of the framework to address the need for a solution that can be used to interpret and unify data that conforms to a range of different semantics. Thus the framework addresses the weakness described within Table 1.1, Issue 4.

Bulk Load and Query Performance

Bulk load operation is costly in terms of the number of index look-ups that are required to create load and integrate the set of nodes, edges and properties that represent each dataset element. However, within an operational system, bulk load operation should not have to be performed on a regular basis as the model and instance level ontologies stored within the warehouse updated without recourse to the; unload, schema update and reload operation that is normally required by RDBMS based warehouse systems. In terms of data load times and query processing the graph database performance characteristics were found to be the opposite to that of a RDBMS, as the size of the graph database increased the traversal query speed remained constant; however, the load times (subgraph insertion) increased as the indexes that are searched to locate the nodes that would be integrated increased in size. This is opposite to that of RDBMS which tend to have constant record insertion performance but with performance loss at query time when the database becomes large due to the compute intensive task of joining large tables. These performance characteristics are due in part to the nature of graph database that employ a method termed index-free adjacency which enable a path to be traversed by simply following the pre-established links between nodes and edges based on set of rules. The compute intensive job is to insert data which involves inserting a node then looking up the adjacent nodes to which it is related and should be joined.

A Single Graph

A high degree of integration and semantic consistency has been achieved as:

a) Every event is a temporal part of a day and the state it creates or dissolves.

b) Companies and company officers are related through role states that are temporal parts of both the person who is the officer and the company that they are part of. They are also integrated with a geographic location.

c) Companies are related via integration with the company status (private limited, plc etc.) taxonomy and the SIC 2007 classification system ranks and classes.

d) All instance level elements within the system are integrated with one or more domain ontologies and, through this, to the foundation.

Identity

As the extentialist criteria of identity is employed, the temporal parts of a company that are director states are also a temporal part of the human director and therefore are the same thing. Graph traversal can be employed to retrieve this temporal mereological detail. This addresses the problem that relates to dividing models into static and dynamic types, that do not capture the state of an object over time.

Graph Based 4D Query Formulation

As 4D-SETL framework results in a graph representation that reflects the 4D patterns of the foundational ontology, queries to retrieve information must also adopt the same 4D paradigm. As in the previous example, the query (graph traversal) is returning a list of people who have a director state that is a constituent temporal part of a particular company. This query returns all these states – everyone who is or who has ever been a director. Therefore it would be possible for the query to return all present directors by eliminating directors that have a resignation event from the results. The system will also manage more complex patterns – e.g. a person who has been a director of the same company more than once (i.e. resigned and reappointed).

Relational Query Performance

To support queries that span relations, RDBMS requires tables to be joined; for example to find all the addresses of the directors of a particular company, would require the companies, directors and address tables to be joined and filtered to obtain the results. Whereas, the graph based solution requires a simple traversal through the edges (tuples) that form these relationships. Therefore, rather having to undertake the compute intensive task of joining very large tables to locate the relationships, these have been pre-established by the 4D-SETL graph bulk load operation. This concurs with the objective measurements for the traversal queries undertaken by Vicknair *et al* (2010), which compared the performance of a graph database with that of a RDBMS and found the graph database to be faster, sometimes by an order of magnitude.

6.5.4 Ontology Evaluation

Geerts and McCarthy (2000) state that although there is no standard measure that can be employed to evaluate an ontology, a set of objective criteria proposed by Gruber (1995) has been widely adopted. Table 6.1 presents the criteria and relates how 4D-SETL fulfils each of the criterion. In addition, domain ontologies produced through the 4D-SETL framework have been modular in nature which has simplified the development and integration process.

Criteria	Description	4D-SETL Compliance
Clarity	Clarity entails ontological definitions that are: Context independent. (socially and computationally) Ontology should effectively communicate the intended meaning of defined terms. Ontologies are also not limited to definitions in the logic sense. Ontology states the axioms that constrain the possible interpretations for the defined terms. Formalism is a means to this end. When a definition can be stated in logical axioms, it should be. Where possible, a complete definition (a predicate defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). All definitions should be documented with natural language.	4D-SETL (BORO) objects are context independent (socially and computationally). 4D-SETL is based on the premise that it is the position of the object within the ontological structure that defines its meaning rather than a defining term. Furthermore, clarity is enhanced by adopting an extensionalist stance regarding identity. 4D-SETL (BORO) ontology includes structures representing the physical thing that stands-in for both objects and the names by which they are known. As the foundation and the domain ontologies produced are based on the BORO which is eternalist they are also free of time based context. 4D-SETL (BORO) effectively communicates the intended meaning of defined terms (it is a name that names something in reality that is an Element a Type or a Tuple). 4D-SETL (BORO) The domain ontologies are not limited to definitions in the logic sense. They can be reflect reality rather than being constrained by a formal logic. The Ontology asserts the axioms that constrain the possible interpretations for the defined terms through the ontological structure. Formalism, The ontology is consistent with the semantics defined within BORO UML. BORO is extensional rather than intensional, therefore class members are asserted rather than described by their attributes. Therefore defining the necessary and sufficient attributes will not be applicable in our case. All definitions are documented with natural language.

Table 6.5: Ontology Evaluation Criteria (Gruber, 1995)

Criteria	Description	4D-SETL Compliance
Coherence	An ontology should be coherent: that is, it should sanction inferences that are consistent with the definitions. Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples.	In relation to the criterion of ontological coherence, 4D-SETL aims to produce inferences that are consistent with reality rather than a FOL logical theory. The experimental data includes the objects, relationships and lifecycles described by natural language documentation from the dataset source and from other authoritative sources such as the companies act (An Act of the House of Commons, 2006). This has led to domain ontologies that maintain coherence with the objects they represent. Furthermore coherence has been demonstrated through information retrieval via query proce4ssing.
Extendibility	It should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialize the ontology monotonically	The 4D-SETL being based on a graph database can be extended and specialised to develop domain ontologies monotonically by adding new types relationships and instances. It has been proven through technical experiment that new elements can be added to the ontology without recourse to the revision to the existent elements and structures. In terms of extending the ontological model to deal with complex structures such as the SIC 2007 classification system 4D-SETL has proved to be extensible and flexible. In addition new instance level information has been successfully integrated within the graph database.
Minimal encoding bias	The ontology should be specified at the knowledge level without depending on a particular symbol-level encoding.	The graph database model employed by 4D-STEL imposes the minimum encoding bias consisting of nodes, edges and properties.
Minimal Commitment	An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities. An ontology should make as few claims as possible about the world being modelled Gruber (1995) states that an ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities and should make as few claims as possible about the world. Furthermore that ontological commitment is based on consistent use of vocabulary. Also an ontology serves a different purpose than a knowledge base, and therefore need only describe a vocabulary for talking about a domain rather than solve a problem or answer arbitrary queries about a domain.	4D-SETL does not meet this criterion as it: i. Commits to the BORO foundation and to view that is grounded in a consistent metaphysical theory (Object Paradigm). This helps to solve the semantic data integration problem that relates to a lack of grounding. ii. The framework does not differentiate between a knowledge base and an ontology. Queries may be processed by 4D-SETL and the ontology used as an integral part of a run-time warehouse system. Is the ontological structure and an objects place in such a structure that defines meaning – rather than a term within a vocabulary (realist rather than an idealist stance).

Table 6.5 Continued: Ontology Evaluation Criteria (Gruber, 1995)

6.5.5 4D Ontology Pattern Selection and Heuristics

During the course of the three iterations the knowledge of the 4D patterns has improved and their application understood. The patterns that emerge from a number of basic types:

> **Location in time and space.** Events are located in a particular spatio-temporal element – in our case a particular day (all space for the 24 hour slice of space time). Physical location (relative) is given by an Element being a part of some other Element – such as a geopolitical region (postcode).

States – the complex changes that something like a company undergoes can be understood via the fusion of all the states that constitute such an element. States within the implementation of this project constituted role states, such as a person being a director or a company being a manufacturer of mobile phones. The majority of these are socially constructed named states. To fully understand the lifecycle of such company elements requires the study of the law which creates and changes them. The events which cause and end such states are related to a temporal location via happens-in patterns and to the element state by happens-to pattern.

Classification systems – such as SIC 2007 and other taxonomies – these can be modelled as-is and integrated with the other elements of the ontology as required. These make use of PowerTypes to structure the Types of Types that constitute the taxonomy.

The sequence of events and states – the history. As the ontology and graph structure is monotonic – new elements can be added without modification to the existing structure and as such the ontology has history baked-in; providing access to all previous states of an element.

However, there remains the learning curve required to make use of such structures that relates to the development of the detailed knowledge of the foundation and how it can be extended.

6.6 Summary and Conclusion

Following the completion of this evaluation of the artefacts produced by this iteration of the design science cycle, the knowledge gathered is deemed sufficient to provide a final evaluation of the artefacts and thus, to fulfil the aim of the project. However, there still remain a number of deficiencies that would benefit from further study of the framework and the supporting tool-set.

The iteration has provided further knowledge related to the exploitation of a perdurantist foundational ontology and a graph database to undertake semantic integration. The effectiveness of the approach has been evaluated through technical experiment and illustrative scenario and in the judgment of the researcher the 4D-SETL framework has met the original aim of the research project.

CHAPTER 7. CONCLUSION AND PROPOSED FUTURE RESEARCH

7.1 Introduction

This chapter represents the final stage of the overarching design science research methodology that was adopted from Vaishnavi and Kuechler (2004) forming the summary and conclusion of this thesis.

The chapter firstly provides a reflection on the value of utilising a design science research methodology. Secondly, the project key findings are presented in relation to the original research aim and objectives, and how these results relate to previous research. The original contribution made to knowledge by this research project is then discussed, together with the advantages that the 4D-SETL approach offer to both the academic and business communities. Finally, potential future research based on the 4D-SETL framework is explored.

7.2 The Value of the Design Science Research Methodology

Key to the project was the methodology adopted from Vaishnavi and Kuechler (2004) which builds on the previous work of others such as Hevner *et al*. (2004) to provide guidance through which to plan and execute a design science research project within the field of information systems. As the research project was conceived without a clear view of how a perdurantist foundational ontology and graph database can be exploited for semantic integration, the core design science design-build–evaluate iterations, as stated by Hevner, *et al*. (2004) provided the means through which to search and discover a solution. This resulted in the 4D-SETL framework that was designed, built and evaluated over the course of three design science iterations. One of the crucial contributions that the design science research methodology provided was the academic discipline and rigour to enable the project to be differentiated from routine information systems design practice. This achieved through adherence to the seven guidelines:

1. The creation of an innovative, purposeful artefact. In the form of the 4D-SETL framework.

2. The artefact is applicable to a specific problem domain; semantic data integration.

3. The evaluation of the effectiveness of the artefact has been undertaken in relation to addressing identified weaknesses in current approaches.

4. The artefact is novel and innovative in solving the known problem of semantic data integration a more effective manner.

5. 4D-SETL has been rigorously defined and the process by which it is created has been described

6. A problem space is constructed; semantic data integration and a mechanism posed or enacted to find an effective solution, 4D-SETL.

7. The design-science research is communicated through this thesis to both technical and business audiences.

7.3 Meeting the Research Aim and Objects

The 4D-SETL framework was designed, built, evaluated and improved over the course of three design science iterations. During the course of each of these cycles the 4D-SETL framework was applied to semantically integrate a number of datasets, which constituted the activity through which to improve the framework and to assess its effectiveness. As the framework exploited a perdurantist foundational ontology and graph database, these were evaluated through undertaking technical experiment and describing the effectiveness through illustrative scenarios. Therefore, the 4D-SETL framework was designed, built and evaluated to serve the main aim of the research.

Rather than being purely theoretical, the utility of the 4D-SETL framework has been evaluated through its application to ETL a number a number of associated datasets to instantiate a prototype warehouse (graph database) populated with the foundational ontology and a number of domain ontologies containing large collections of individual instances all of which have been successfully integrated. It can therefore be concluded that through the 4D-SETL framework, the research aim and the original objectives that were established at the outset of the design science project have been successfully met.

7.4 Key Findings

The first key finding was that the 4D-SETL framework proved an effective means of performing semantic data integration. The BORO perdurantist foundational ontology provided the philosophically grounded view of reality solving the grounding problem described by Cregan (2007). Thus BORO provides a coherent lens through which to view and model the world together with the foundational ontological elements and patterns through which the domain ontologies could be developed to represent the datasets to be semantically integrated as described by (Partridge, 2005). Therefore in terms of domain ontology development, this work concurs with the work of (Keet, 2011), who stated that employing foundational ontologies provides advantages in terms of the quality and interoperability of domain ontologies. Developing such domain ontologies provided the means of semantically integrating data conforming to different models and theories. This finding corresponds with that of Partridge (2002). However, this research builds on this previous work by employing the perdurantist foundation ontology to provide the fundamental structure for a warehouse system instantiated using a graph database system.

A related finding was employing a graph database provided the means of importing and restructuring data in a manner that directly reflects the ontological model patterns without the normal translation to tabular RDBMS or OO form solving the ‘impedance mismatch’ problem described by Ireland *et al* (2009). Dispensing with RDBMS storage in favour of a property graph data model removed the partitioning of the storage structures between data and schema and allows both ‘schema’ ontological model objects and instance level objects to be updated at run time. This supports the work related to graph databases by (Webber, 2012).

The 4D-SETL Framework demonstrated that patterns could be established within the warehouse that directly reflected the physical or socially constructed patterns of reality such as taxonomies and taxonomic ranks, the latter of which employed the power-type pattern (type of types) to more accurately reflect the nature of such classification systems. These aspects of 4D ontologies (along with others) provided a greater level of flexibility and reusability when evolving the warehouse system and therefore concur and take forward the initial findings of Partridge (2005).

Furthermore, the researcher found that employing a realist foundational ontology facilitated the creation of domain ontologies that are focused beyond the artefacts of computation to the reality to which these artefacts relate, and therefore confirms the previous research of Smith and Welty (2001) that conclude, that philosophical ontology can be applied to solve practical problems, in our case semantic integration of data.

Another useful outcome was to demonstrate that it was possible for 4D-SETL BORO ontologies to capture both structure and how a structures change over time within the same model; thus, concurring with the finding of Bailey (2011). However, this research builds on this previous work by implementing the model within a graph based warehouse.

The monotonic nature of 4D-SETL enabled ontologies and datasets to be added to the system without recourse to a database rebuild. Such monotonic stores are amenable to functional programming and therefore may better support parallel programming environment to process data at scale.

An additional finding was that the graph patterns that emerged after the completion of the integration met the criterion that Hüsemann *et al.* (2000) describe as the common understanding of a well-designed warehouse schema as it has a form of a “star”, i.e., it consists of a fact table (in the case of 4D-SETL graph node) that contains the facts of interest relating to the areas of interest. The resultant graph actually consisted of a number of such stars which are centred on temporal facts (events on a particular day), location facts (geographic location), director and company etc.

Of further interest, the star patterns were integrated through the relationships between these nodes so it is possible to extract information relating to events on a particular day, relating to a particular type of company in a particular geographic area etc. The 4D-SETL warehouse, being graph based, was therefore more suitable for discovering relationships within data rather than for processing aggregate data. For example it was found that it was possible to discover all relationships that exist between two elements using an algorithm from the Neo4J library that can find all available paths or the shortest path between two nodes. The Cipher graph database query facilities as described by Webber (2012) also allow the discovery of more complex patterns of relationships between the people, company officers, company activities, events and physical location.

Finally, it was found through the evaluation and empirical experiment on the prototype warehouse (graph database) that data load and information retrieval response times that the prototype could be developed into a practical information system. This was confirmed by testing data query (graph traversal) experiments that produced indicative response time within bounds that support interactive applications as described by Bhatti (2000). This also confirmed the graph database performance evaluation undertaken by Vicknair *et al.* (2010). Thus, using a graph database and the parameter graph model to store the ontology and query information via graph traversal circumvents the issues that limit the ability of systems built using triple stores and tableau calculus based reasoner technology to deal with ontologies that contain very large instance level elements (Bock *et al.*, 2008).

In conclusion, the utility of the 4D-SETL was demonstrated through the resultant systems ability to model and instantiate a number of complex ontological structures, such as higher order taxonomic ranks. The patterns specialised from the core foundational BORO ontology patterns were found to offer a high degree of flexibility and reusability when evolving the graph based warehouse system, and therefore concurs with the research of Partridge (2002) and de Cesare *et al*. (2013) and Bailey and Partridge (2009).

7.5 **Original Contribution to Knowledge**

Within this section the original contribution made to knowledge by this study of applying perdurantist foundational ontologies to semantic data integration is discussed along with advantages that the outputs of the study can offer for academic and business communities.

7.5.1 Major Contribution

The strengths and originality of this study lie in a number of areas. As the mainstream semantic research is concentrated within the OWL and DL based ontologies (i.e. not 4D) and highly scalable graph databases are a relatively recent technology, there has been little previous research into the application of 4D ontologies to semantically integrate data within a database employing the property graph model. Therefore, the primary output of this research is that within the scope of the study, that the the 4D-SETL framework can be exploited as an effective means of performing semantic data integration, in that it addresses a number of the problems that are inherent to current practice. Furthermore, due to the focus on OWL based approaches, there has been little work that has focused on development of a framework to guide researchers and industry practitioner of the use and utility of such an approach. Therefore, a contribution to knowledge from this research is the 4D-SETL framework which has been developed by the researcher. This artefact is a novel method for performing semantic integration that exploits a perdurantist (4D) foundational ontology as a core component of both the method but also of the technical artefacts produced as a result of the application the framework – the warehouse graph database system.

7.5.2 Other Contributions

In terms of other contributions to the corpora of knowledge related to semantic integration, the prototype instantiation has proved a useful vehicle for exploring the use of the 4D paradigm and through the process of developing models, restructuring and semantically aligning and integrating disparate datasets. In addition, the prototype warehouse provided direct evidence of the effectiveness of the artefact in terms of observed performance, which is deemed to be a valuable of design research output by Peffers *et al.* (2012).

The exploitation of a perdurantist foundational ontology and graph database was researched in this study and the approach evaluated through illustrative scenario and by conducting technical experiments on the prototype warehouse instantiation that resulted from the application of 4D-SETL. The prototype provides evidence that an instantiation based on 4D-SETL could function at scale and provide useful information retrieval services. Therefore 4D-SETL has been proven, within the confines of the datasets integrated, to work in practice – a further contribution to knowledge. This was an important contribution, as in order for ontologies to provide concrete and visible benefits to information systems design and engineering practice, it is essential to take ontologies beyond the modelling/design stage and attempt to use them not only to influence the implementation of technological artefacts (e.g., databases and software), but also to implement the ontologies in the technology itself. This research has resulted in an instantiation that is able to exploit the foundational ontology, modelled domain ontologies (together with instance level) to create a software implementation that maintains direct traceability to the ontology. This represents a major outcome from the research – a warehouse system that has instantiated a 4D foundation, several domain ontologies and large scale instance level ontology models within a single coherent environment.

A further benefit of this research has been to show, through illustrative scenario, the advantages of the 4D-SETL approach over more traditional design practice (e.g., relational and object based systems). These advantages include the way in which the perdurantist foundational ontology offers a coherent lens through which to view and model reality thus, providing a repeatable and consistent method for developing domain ontologies, supporting the assertion of Partridge (2005). In addition, the fundamental graph structures can be used within the warehouse to organise and structure data and provide a clear and unambiguous view of identity over time.

Furthermore, this project has shown that using this method 4D-SETL can provide guidance for academics and system developers as to how 4D ontologies can be applied to graph based schemata which is useful as there is a lack of guidelines. Thus, evolving the 4D-SETL framework represents a notable area of research. Applying the design science research methodology facilitated this by using a number of iterations of the design-build-evaluate cycles with the results of each cycle contributing to the next to produce the final version of 4D-SETL. This methodology thus provides the documented guidelines for performing the 4D-SETL process and can therefore inform other who wish to undertake similar semantic data integration projects.

In terms of the research offering value to business, there are a number of advantages that can be asserted:

a) A 4D-SETL system can be used to interpret (design domain models) and integrate semantic data in a consistent way to produce a single coherent graph (warehouse database) that holds large collections of instance level objects together with the foundation and domain ontologies, which have direct traceability to the BORO UML ontology design time artefacts.

b) A graph based storage scheme that reflects the 4D ontological patterns can combine structural and temporal information within a single coherent information system.

c) The graph based structure enables new schema and instance data to be added to the warehouse without the normal system rebuild which is necessary using more traditional relational database management system approaches. Thus the warehouse forms a monotonic store.

d) As the warehouse stores current and historic information it is therefore possible to query the system to retrieve, through graph traversals, any current or previous state.

e) A graph based approach that enables graph traversal and graph algorithms to be employed to discover relationships – between events, people, company officers, company activities and physical locations.

f) As the graph database employed supports tens of billions of nodes and relationships, the research has produced a system that, in relation to the limited number of datasets integrated, has proven to work at scale.

This thesis presents a design science research project which has been undertaken to explore the exploitation of a foundational perdurantist ontology and a graph database to perform sematic data integration. Although the contributions made to the area of knowledge do not solve all the challenges of semantic data integration, they do consider and highlight the many challenges that are inherent within the problem space. Hopefully the 4D-SETL framework described will help other researchers and commercial practitioners to build on this work to; exploit foundational ontologies and to understand and philosophical grounding that underlie such ontologies and to utilise graph databases as a means of realising practical information system that directly reflect the patterns and structures inherent to such ontologies. As the 4D-SETL methodology differs significantly from current information systems solutions, it is hoped that this work will contribute to the corpus of knowledge related to Ontology-Driven Information Systems Engineering (ODISE).

7.6 Research Limitations and Potential for Future Research

Undertaking this research project has been an invaluable experience. With the benefit of the learning that has taken place during the course of the study, the researcher is aware 4D-SETL is an initial implementation and that the framework and the tools and technologies need to be improve. One limitation to the study was that the researcher did not address the issue of the automatic configuration of the Extract, Transform and Load process for both stages - it was only completed automatically for the ontology and not the instance Load stage. As the emphasis was on proving that the framework to function at scale, the data stage configuration was achieved mainly through custom coding, which proved time consuming. Another limitation was that not all data fields from the available datasets were integrated and processed, which would have increased the scale of the resultant warehouse instantiation.

The experimental testing of the resultant warehouse artefacts has been limited to the loading and a number of basic information retrieval tests in the form of graph traversals. The system would benefit from the addition of other related datasets and further experimental evaluation to study the systems behaviour under various load conditions such as multiple simultaneous client access etc.

7.6.1 Disadvantages and Barriers to Adoption

There a number of disadvantages and barriers to adoption that will need to be overcome before commercial and technical adoption of the perdurantist approach inherent in the 4D-SETL framework. As with previous paradigm shifts the unification of space and time into one fabric (space-time), has not changed contemporary society or most engineering practice. Software design and modelling practice is still largely based on the ‘Entity’ paradigm, a simplified version of the Aristotelian substance paradigm, which for over two millennia has provided the predominant paradigm which views the physical world as being constructed from three dimensional entities that endure through time (Partridge, 2005). Thus, the primary barrier to adoption is the lack of understanding of the view inherent to perdurantist (4D) ontologies and therefore the BORO. There is also a lack of software tools that can be employed at both design-time and run-time to engineer solutions based on 4D and higher order ontologies.

Consequently, it may not be currently realistic for most software and system developers to undertake the implementation of systems based on this approach.

7.6.2 Potential for Future Research

Therefore, in order to build on to the positive contributions to knowledge made by this research, there are a number of other relevant areas that there is potential for future research projects to explore.

It would be useful to complete the coding related to the automatic configuration of the dataset Extract and Transform process as this would save time by reducing the need to custom code for each new data set. The thesis presents a framework and implementation of Semantic 4D Extract Transform and Load (4D-SETL) to solve the problem of semantic integration of knowledge (data) from a number of sources. One area worthy of investigation is the integration of additional datasets from Linked Open Data and semantically annotated web pages. As the system is grounded by the use of a foundational ontology and employs reference data with a high degree of provenance, coverage and accuracy, such as system could act as a hub through which to integrate data from less dependable sources (LOD). Therefore, it would be useful to study how such data could be structured and integrated.

More work could also be undertaken to develop specific criteria and metrics to measure the quality of the developed 4D domain ontologies. Although BORO provides a clear and concise means of selecting the foundation category that an object is a member in a similar way to that employed by Ontoclean (Guarino and Welty, 2002) some guidance as to the anti-patterns that can occur and the problems they can introduce would beneficial.

Within the current project, semantic data has been has been stored within a graph database. A useful addition would be to apply the framework to derive other NOSQL implementations such as key-value stores in order to determine the utility of extending the 4D-SETL framework to encompass other storage (warehouse) options.

The 4D-SETL approach has been tested empirically within the current research as it was applied to construct the ontology models and reflect them within a graph database. However, this framework and its implementation would benefit from further testing and validation, furthermore the domain ontologies could be extended to encompass more fine grained knowledge related to UK Business; such as business to business transaction details.

More work could also be done on the extraction of knowledge from the warehouse. This could be in the form of sub-graphs relating to a period of time or a network of facts relating to a major business event such as a company bankruptcy and dissolution. The researcher believes that the framework and implementation would scale to encompass such models.

The 4D Semantic Extract Transform and Load method provides a precise description for the development of integrated knowledge stores. The method has been validated through empirical experiment, however, there remains further work to test the method for dealing with more complex business processes and relationships, big data.This environment also provides an opportunity to carry forward the fundamental foundational structures into the application – such as those which are derived from foundational ontological models thus maintaining a clear semantic view at the application level.

REFERENCES

Ahmad, M. and Odeh, M. (2012) 'Semantic Derivation of Enterprise Information Architecture from Business Process Architecture', *Computer Theory and Applications (ICCTA), 2012 22nd International Conference on.* IEEE, 79-84.

Al-Debei, M., Asswad, M., de Cesare, S. and Lycett, M. (2012) 'Conceptual modelling and the quality of ontologies: Endurantism vs. perdurantism', *International Journal of Database Management Systems (IJDMS),* 4(3).

An Act of the House of Commons (2006) *Companies Act*United Kingdom of Great Britain and Northern Ireland: United Kingdom of Great Britain and Northern Ireland.

Andreessen, M. and Bina, E. (1994) 'NCSA Mosaic: A Global Hypermedia System', *Internet Research,* 4(1), pp. 7-17.

Arsanjani, A. (2002) 'Developing and Integrating enterprise Components and Services', *Communications of the ACM,* 45(10), pp. 30-34.

Bailey, I. (2011) 'Enterprise Ontologies–Better Models of Business', in *Intelligence-based systems engineering.* Springer, pp. 327-342.

Bailey, I. and Partridge, C. (2009) 'Working with extensional ontology for defence applications', *Ontology in Intelligence Conference.*

Barabási, A. and Albert, R. (1999) 'Emergence of scaling in random networks', *Science,* 286(5439), pp. 509-512.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The semantic web', *Scientific American,* 284(5), pp. 28-37.

Bhatti, N., Bouch, A. and Kuchinsky, A. (2000) 'Integrating user-perceived quality into web server design', *Computer Networks,* 33(1), pp. 1-16.

Bizer, C., Heath, T. and Berners-Lee, T. (2009) 'Linked data-the story so far', *International Journal on Semantic Web and Information Systems,* (Special Issue on Linked Data).

Bock, J., Haase, P., Ji, Q. and Volz, R. (2008) 'Benchmarking OWL reasoners', *Proc. of the ARea2008 Workshop, Tenerife, Spain (June 2008).*

Booch, G., Jacobson, I. and Rumbaugh, J. (2000) 'OMG unified modeling language specification', *Object Management Group ed: Object Management Group,* 1034.

Bradley, F.H. (1899) *Appearance and reality: A metaphysical essay.* Macmillan.

Bricker, P. (2014) 'Ontological Commitment', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy*. Winter 2014 edn.

Brickley, D. and Guha, R. (2000) 'Resource Description Framework (RDF) Schema Specification 1.0: W3C Candidate Recommendation 27 March 2000', .

Brin, S. and Page, L. (1998) 'The anatomy of a large-scale hypertextual Web search engine', *Computer Networks and ISDN Systems,* 30(1–7), pp. 107-117.

Campbell, A. and Shapiro, S. (1995) 'Ontological Mediation: An Overview', *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing.* 1995. AAAI Press, Menlo Park, CA,.

Chandrasekaran, B., Josephson, J. and Benjamins, R. (1999) 'What are ontologies, and why do we need them?', *IEEE Intelligent systems,* 14(1), pp. 20-26.

Chen, P. (1976) 'The entity-relationship model—toward a unified view of data', *ACM Transactions on Database Systems (TODS),* 1(1), pp. 9-36.

Codd, E. (1971) 'Normalized data base structure: A brief tutorial', *Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control.* ACM, 1-17.

Codd, E. (1970) 'A relational model of data for large shared data banks', *Communications of the ACM,* 13(6), pp. 377-387.

Cohen, S. (2004) 'Identity, Persistence, and the Ship of Theseus', *Department of philosophy, University of Washington (http://faculty.washington.edu/smcohen/320/theseus.html),* .

Companies House (2013) *Companies House*. Available at: http://www.companieshouse.gov.uk/about/functionsHistory.shtml (Accessed: 4/2014 2014).

Cregan, A. (2007) 'Symbol grounding for the semantic web', in *The Semantic Web: Research and Applications.* Springer, pp. 429-442.

Cycorp Inc. (2015.) *Cyc Ontology*. Available at: www.cyc.com (Accessed: 01/11 2015).

Dahl, R. (2007) *Charlie and the chocolate factory*. Penguin UK.

de Cesare, S., Foy, G. and Partridge, C. (2013) 'Re-engineering Data with 4D Ontologies and Graph Databases', *Advanced Information Systems Engineering Workshops.* Springer, 304-316.

Dean, J. and Ghemawat, S. (2008) 'MapReduce: simplified data processing on large clusters', *Communications of the ACM,* 51(1), pp. 107-113.

Dijkstra, E. (1959) 'A note on two problems in connexion with graphs', *Numerische mathematik,* 1(1), pp. 269-271.

Doan, A., Noy, N.F. and Halevy, A.Y. (2004) 'Introduction to the special issue on semantic integration', *ACM Sigmod Record*, 33(4), pp. 11-13.

DoDAF Architectures Framework Working Group (2009) 'DoDAF Architecture Framework version 2.0', *Department of Defense United States, .*

Dorr, C. (2005) 'What We Disagree About When We Disagree About Ontology', in Kalderon, M.E. (ed.) *Fictionalism in metaphysics.* Oxford; New York: Clarendon Press; Oxford University Press, pp. 234-235-286.

Einstein, A. (1920) *Relativity: The Special and General Theory.*. New York: Henry Holt and Company.

Fowler, M. (2002) *Patterns of enterprise application architecture.* Addison-Wesley Longman Publishing Co., Inc.

Fowler, M. and Sadalage, P. (2012) *NoSQL distilled : a brief guide to the emerging world of polygot persistence.* Boston, Mass. ; London: Addison-Wesley.

Fraser, S., Beck, K., Caputo, B., Mackinnon, T., Newkirk, J. and Poole, C. (2003) 'Test driven development (TDD)', in *Extreme Programming and Agile Processes in Software Engineering.* Springer, pp. 459-462.

Frege, G. (1948) 'Sense and reference', *The philosophical review,* , pp. 209-230.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. and Schneider, L. (2002) 'Sweetening ontologies with DOLCE', in *Knowledge engineering and knowledge management: Ontologies and the semantic Web.* Springer, pp. 166-181.

Geerts, G. (2011) 'A design science research methodology and its application to accounting information systems research', *International Journal of Accounting Information Systems,* 12(2), pp. 142-151.

Geerts, G. and McCarthy, W. (2002) 'An ontological analysis of the economic primitives of the extended-REA enterprise information architecture', *International Journal of Accounting Information Systems*, 3(1), pp. 1-16.

Geerts, G.L. and McCarthy, W.E. (2000) 'The ontological foundation of REA enterprise information systems', *Annual Meeting of the American Accounting Association, Philadelphia, PA.* Citeseer, 127-150.

Google, Inc., Yahoo, Inc., and Microsoft Corporation (2015) *schema.org*. Available at: www.schema.org (Accessed: 01/15 2015).

Grenon, P. and Smith, B. (2004) 'SNAP and SPAN: Towards dynamic spatial ontology', *Spatial cognition and computation*, 4(1), pp. 69-104.

Gruber, T.R. (1993) 'A Translation Approach to Portable Ontology Specifications.', *Knowledge Acquisition,* 5(2), pp. 199-220.

Gruber, T.R. (1995) 'Toward principles for the design of ontologies used for knowledge sharing?', *International journal of human-computer studies*, 43(5), pp. 907-928.

Guarino, N. and Welty, C. (2002) 'Evaluating ontological decisions with OntoClean', *Communications of the ACM*, 45(2), pp. 61-65.

Guizzardi, G., de Almeida Falbo, R. and Guizzardi, R. (2008) 'Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology.', *CIbSE*. , 127-140.

Harnad, S. (1990) 'The symbol grounding problem', *Physica D: Nonlinear Phenomena*, 42(1), pp. 335-346.

Hart, P., Nilsson, N. and Raphael, B. (1968) 'A formal basis for the heuristic determination of minimum cost paths', *Systems Science and Cybernetics, IEEE Transactions on*, 4(2), pp. 100-107.

Herre, H. (2010) 'General Formal Ontology (GFO): A foundational ontology for conceptual modelling', in *Theory and applications of ontology: computer applications*. Springer, pp. 297-345.

Hevner, A., March, S. and Park, J. (2004) 'Design research in information systems research'', , no. 1, pp. 75–105.', *MIS Quarterly, [Online]*, 28(1), pp. 75-76-105.

Hevner, A., March, S., Park, J. and Ram, S. (2004) 'Design science in information systems research', *MIS quarterly*, 28(1), pp. 75-105.

Hofweber, T. (2014) 'Logic and Ontology', in Zalta N, E. (ed.) *The Stanford Encyclopedia of Philosphy*. 2014th edn. Stanford University.

Hudak, P. (1989) 'Conception, evolution, and application of functional programming languages', *ACM Computing Surveys (CSUR)*, 21(3), pp. 359-411.

Hüsemann, B., Lechtenbörger, J. and Vossen, G. (2000) *Conceptual data warehouse design*. Universität Münster. Angewandte Mathematik und Informatik.

IDEAS Group (2011) *International defence enterprise architecture specification*. Available at: http://www.ideasgroup.org/ (Accessed: 01/15 2015).

Iivari, J. (2007) 'A paradigmatic analysis of information systems as a design science', *Scandinavian Journal of Information Systems*, 19(2), pp. 5.

Ireland, C., Bowers, D., Newton, M. and Waugh, K. (2009) 'A classification of object-relational impedance mismatch', *Advances in Databases, Knowledge, and Data Applications, 2009. DBKDA'09. First International Conference on*. IEEE, 36-43.

Jain, P., Hitzler, P., Sheth, A. and Verma, K. (2010) 'Ontology alignment for linked open data', in *The Semantic Web–ISWC 2010*. Springer, pp. 402-417.

Kääriäinen, J., Eskeli, J., Teppola, S., Välimäki, A., Tuuttila, P. and Piippola, M. (2009) 'Extending global tool integration environment towards lifecycle management', *On the Move to Meaningful Internet Systems: OTM 2009 Workshops.* Springer, 238-247.

Kalfoglou, Y. and Schorlemmer, M. (2003) 'Ontology mapping: the state of the art', *The knowledge engineering review,* 18(01), pp. 1-31.

Kappel, G., Kapsammer, E., Kargl, H., Kramler, G., Reiter, T., Retschitzegger, W., Schwinger, W. and Wimmer, M. (2006) *Lifting metamodels to ontologies: A step to the semantic integration of modeling languages.* Springer.

Katasonov, A. and Lattunen, A. (2014) 'A Semantic Approach to Enterprise Information Integration', *Semantic Computing (ICSC), 2014 IEEE International Conference on.* IEEE, 219-226.

Keet, M. (2011) 'The use of foundational ontologies in ontology development: an empirical assessment', in *The Semantic Web: Research and Applications.* Springer, pp. 321-335.

Kent, W. (1978) *Data and reality : basic assumptions in data processing reconsidered.* Amsterdam ; Oxford: North-Holland Publishing Co.

Kovse, J. and Härder, T. (2002) 'Generic XMI-based UML model transformations', in *Object-oriented information systems.* Springer, pp. 192-198.

Kripke, S.A. (1959) 'A completeness theorem in modal logic', *The journal of symbolic logic,* 24(01), pp. 1-14.

Kuhn, T. (1964) *The structure of scientific revolutions.* Chicago: Phoenix Books.

Lewis, D.K. (1986) *On the plurality of worlds.* Cambridge Univ Press.

Look, B.C. (2013) 'Leibniz's Modal Metaphysics', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy.* Spring 2013 edn.

Lowe, E.J. (1998) 'Ontology.', in Hondreich, T. (ed.) *The Oxford Companion to Philosophy.* New York: Oxford University Press, pp. 634.

March, S. and Smith, G. (1995) 'Design and Natural Science Research on Information Technology', *Decision Support Systems,* 15(4), pp. 251-252-266.

Markosian, N. (2014) 'Time', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy.* Spring 2014 edn.

McCarthy, W. (1982) 'The REA accounting model: A generalized framework for accounting systems in a shared data environment', *Accounting Review,* , pp. 554-578.

McGuinness, D. (2005) 'Ontologies come of age', *Spinning the semantic web: bringing the World Wide Web to its full potential,* , pp. 171.

McTaggart, J.E. (1908) 'The unreality of time', *Mind*, , pp. 457-474.

Motik, B., Grau, B., Horrocks, I., Wu, Z., Fokoue, A. and Lutz, C. (2009) 'Owl 2 web ontology language: Profiles', *W3C recommendation*, 27, pp. 61.

Munir, K., Odeh, M. and McClatchey, R. (2012) 'Ontology-driven relational query formulation using the semantic and assertional capabilities of OWL-DL', *Knowledge-Based Systems*, 35, pp. 144-159.

Myroshnichenko, I. and Murphy, M. (2009) 'Mapping ER schemas to OWL ontologies', *Semantic Computing, 2009. ICSC'09. IEEE International Conference on.* IEEE, 324-329.

Neo Technology, I. (2015) *Neo4j*. Available at: www.neo4j.com (Accessed: 02/11 2015).

Newell, A. and Simon, H. (1976) 'Computer science as empirical inquiry: Symbols and search', *Communications of the ACM*, 19(3), pp. 113-126.

Niles, I. and Pease, A. (2001) 'Towards a standard upper ontology', *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001.* ACM, 2-9.

Noy, N.F. (2004) 'Semantic Integration: A Survey of Ontology-based Approaches', *SIGMOD Rec.*, 33(4), pp. 65-70.

Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Baumann, S., Vembu, S. and Romanelli, M. (2007) 'DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO)', *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(3), pp. 156-174.

Object Management Group (OMG) (2011) *Business Process Model and Notation (BPMN)*. Available at: http://www.omg.org/spec/BPMN/2.0/PDF/ (Accessed: 12/14 12/14).

Office for National Statistics (2015) *Home Page*. Available at: http://www.ons.gov.uk/ons/index.html (Accessed: 01/14 2015).

Office of National Statistics (2011) *UK Standard Industrial Classification 2007 (UK SIC 2007)<br />2011-08-06T15:25:00Z*. Available at: http://www.ons.gov.uk/ons/guide-method/classifications/current-standard-classifications/standard-industrial-classification/index.html (Accessed: 4/08 2014).

Ogden, C., Richards, A., Malinowski, B. and Crookshank, G. (1946) *The meaning of meaning*. Harcourt, Brace & World New York.

Partridge, C. (2002) 'The role of ontology in integrating semantically heterogeneous databases', *Rapport technique*, 5(02).

Partridge, C., Gonzalez-Perez, C. and Henderson-Sellers, B. (2013) 'Are conceptual models concept models?', in *Conceptual Modeling*. Springer, pp. 96-105.

Partridge, C., Mitchell, A. and de Cesare, S. (2013) 'Guidelines for developing Ontological Architecures in Modelling and Simulation', in Tolk, A. (ed.) *Ontology, Epistemology, and Teleology for Modeling and Simulation.* Berlin Heidelberg: Springer, pp. 27-57.

Partridge, C. and Stefanova, M. (2001) 'A Synthesis of State of the Art Enterprise Ontologies, Lessons Learned', *The BORO Program, LADSEB CNR*, .

Partridge, C. (2005) *Business objects.* 2nd edn. Oxford: Butterworth Heinemann.

Pease, A. and Niles, I. (2002) 'IEEE standard upper ontology: a progress report', *The Knowledge Engineering Review,* 17(01), pp. 65-70.

Peffers, K., Rothenberger, M., Tuunanen, T. and Vaezi, R. (2012) 'Design science research evaluation.', in Peffers, K., Rothenberger, M. and Kuechler, B. (eds.) *Design Science Research in Information Systems. Advances in Theory and Practice. Lecture Notes in Computer Science.* Berlin Heidelberg: Springer, pp. 398-399-410.

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007) 'A design science research methodology for information systems research', *Journal of Management Information Systems,* 24(3), pp. 45-77.

Public Data Group (2015) *Home Page*. Available at: https://www.gov.uk/government/groups/public-data-group (Accessed: 01/20 2015).

Quine, W.V.O. (1948) 'On What There Is', *The Review of Metaphysics,* 2(5), pp. 21-22-38.

Quine, W.V. (1952) *Methods of logic.* Routledge and Kegan Paul.

Rector, A. (2003) 'Medical informatics', *The description logic handbook.* Cambridge University Press, 406-426.

Roddick, J. (1995) 'A survey of schema versioning issues for database systems', *Information and Software Technology,* 37(7), pp. 383-393.

Rosen, G. (2014) 'Abstract Objects', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy.* Fall 2014 edn.

Rumbaugh, J., Jacobson, I. and Booch, G. (2004) *Unified Modeling Language Reference Manual, The.* Pearson Higher Education.

Russell, B. (1902) *Letter to Frege*.

Saltor, F., Castellanos, M. and Garcia-Solaco, M. (1991) 'Suitability of Data models As Canonical Models for Federated Databases', *SIGMOD Rec.,* 20(4), pp. 44-48.

Schon, D. (1992) 'Designing as reflective conversation with the materials of a design situation', *Research in Engineering Design,* 3(3), pp. 131-147.

Shadbolt, N., O'Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H. and Hall (2012) 'Linked open government data: Lessons from data. gov. uk', *IEEE Intelligent Systems*, 27(3), pp. 16-24.

Sheth, A.P. (1999) 'Changing focus on interoperability in information systems: from system, syntax, structure to semantics', in *Interoperating geographic information systems.* Springer, pp. 5-29.

Sheth, A.P. and Larson, J.A. (1990) 'Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases', *ACM Comput.Surv.*, 22(3), pp. 183-236.

Shvaiko, P. and Euzenat, J. (2013) 'Ontology matching: state of the art and future challenges', *Knowledge and Data Engineering, IEEE Transactions on*, 25(1), pp. 158-176.

Sider, T. (2003) *Four-dimensionalism: An ontology of persistence and time.* Oxford.

Sider, T. (2001) 'Criteria of personal identity and the limits of conceptual analysis', *Noûs*, 35(s15), pp. 189-209.

Sider, T. (1996) 'Intrinsic properties', *Philosophical Studies*, 83(1), pp. 1-27.

Simon, H.A. (1996) *The sciences of the artificial, by Herbert A. Simon.* 3rd edn. Cambridge: M.I.T. Press.

Smart, P.R. and Engelbrecht, P.C. (2008) 'An analysis of the origin of ontology mismatches on the semantic web', in *Knowledge Engineering: Practice and Patterns.* Springer, pp. 120-135.

Smith, B. and Ceusters, W. (2010) 'Ontological Realism as a Methodology for Coordinated Evolution of Scientific OntologiesSmith, Barry, and Werner Ceusters. 5.3 (2010): 139-188.', *Applied Ontology*, 5(3), pp. 139-140-188.

Smith, B. and Welty, C. (2001) 'Ontology: Towards a new synthesis', *Formal Ontology in Information Systems.* ACM Press, USA, pp. iii-x, 3-9.

Smith, B. (2004) 'Beyond concepts: ontology as reality representation', *Proceedings of the third international conference on formal ontology in information systems (FOIS 2004).* , 73-84.

Snowdon, P. (2009) 'Peter Frederick Strawson', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy.* Fall 2009 edn.

Sowa, J.F. (1999) 'Knowledge representation: logical, philosophical, and computational foundations', .

Sparx Systems Pty Ltd. (2015) *Enterprise Architect*. Available at: http://www.sparxsystems.com/ (Accessed: 05/01 2015).

Stanford Center for Biomedical Informatics Research (2015) *Protégé* . Available at: protege.stanford.edu (Accessed: 01/16 2015).

Stell, J.G. and West, M. (2004) 'A four-dimensionalist mereotopology', *Proceedings of the International Conference on Formal Ontology in Information Systems.* , 261-272.

Stoermer, H. and Bouquet, P. (2009) 'A novel approach for entity linkage', *Information Reuse & Integration, 2009. IRI'09. IEEE International Conference on.* IEEE, 151-156.

Strang, C. and Rees, D. (1963) 'Symposium: Plato and the Third Man', *Proceedings of the Aristotelian Society, Supplementary Volumes,* 37, pp. 147-176.

Strawson, P.F. (1964a) 'Identifying reference and truth-values', *Theoria*, 30(2), pp. 96-118.

Strawson, P.F. (1964b) *Individuals: an essay in descriptive metaphysics cP.F. Strawson.* London: Methuen.

Strozzi, C. (2010) *NoSQL Relational Database Management System*. Available at: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page (Accessed: 11/16 2012).

The Eclipse Foundation (2015) *The Eclipse Integrated Development Enviromment (IDE)*. Available at: http://www.eclipse.org (Accessed: 05/01 2015).

Turing, A. (1964) 'Computing machinery and intelligence', in Anderson, A. (ed.) *Minds and machines.* Engelwood Cliffs New Jersey: Prentice Hall.

UK Office of National Statistics (2014) *National Statistics Postcode Lookup (UK) Aug 2014*. Available at: https://geoportal.statistics.gov.uk/geoportal/catalog/search/resource/details.page?uuid=%7B9992801C-8454-4A7A-A84F-6DD25D622E0B%7D (Accessed: 08/08 2014).

Vaishnavi, V. and Kuechler, W. (2004) 'Design research in information systems', .

Van Inwagen, P. (1998) 'Modal epistemology', *Philosophical Studies,* 92, pp. 431-432-444.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. and Wilkins, D. (2010) 'A comparison of a graph database and a relational database: a data provenance perspective', *Proceedings of the 48th annual Southeast regional conference.* ACM, 42.

Visser, P.R., Jones, D.M., Bench-Capon, T. and Shave, M. (1997) 'An analysis of ontology mismatches; heterogeneity versus interoperability', *AAAI 1997 Spring Symposium on Ontological Engineering, Stanford CA., USA.* , 164-172.

W3C (2009) *OWL 2 Web Ontology Language*. Available at: http://www.w3.org/TR/owl2-overview/ (Accessed: 11/20 2012).

W3C (2004) *Resource Description Framework (RDF): Concepts and Abstract Syntax*. Available at: http://www.w3.org/TR/rdf-concepts/ (Accessed: 11/20 2012).

Walls, J.G., Widmeyer, G.R. and El Sawy, O.A. (1992) 'Building an information system design theory for vigilant EIS', *Information systems research,* 3(1), pp. 36-59.

Warmer, J. and Kleppe, A. (2006) 'Building a flexible software factory using partial domain specific models', .

Webber, J. (2012) 'A programmatic introduction to Neo4j', *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity.* ACM, 217-218.

West, M., Partridge, C. and Lycett, M. (2006) 'Enterprise data modelling: Developing an ontology-based framework for the shell downstream business', *Formal Ontology Meets Industry (FOMI 2006),* .

William, S., Johannesson, P. and Bubenko, J. (1996) 'Semantic similarity relations and computation in schema integration', *Data & Knowledge Engineering,* 19(1), pp. 65-97.

www.vim.org (2015) *The VIM Editor*. Available at: www.vim.org (Accessed: 04/01 2015).

Zachman International, I. (2015) *Zachman Enterprise Achitechture Framework*. Available at: www.zachman.com (Accessed: 01/15 2015).

Zowghi, D. and Gervasi, V. (2003) 'On the interplay between consistency, completeness, and correctness in requirements evolution', *Information and Software Technology,* 45(14), pp. 993-1009.

APPENDICES

Appendix A: BORO Foundation Graph DB Node listing

```
{"ea_model_name":"uml:Class","ontology_node_id":"9eaf0445-3f51-4497-b860-
5b8cfe564b7d","ontology_type":"types","ontology_name":"Elements","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_DF1D47F3_05F1_427a_A0FF_967BFA5B10CC"}
{"ea_model_name":"uml:Class","ontology_node_id":"f9a1ffcf-8507-4ec5-b900-
59646e9ced6b","ontology_type":"Types","ontology_name":"Objects","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_92776CC2_C077_424b_8E3C_CB39996B74DB"}
{"ea_model_name":"uml:Class","ontology_node_id":"45b294ff-b6e5-450f-a0b0-
1a51b0f0b1cd","ontology_type":"types","ontology_name":"Types","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_13B8091C_0041_40d8_9DE3_6128A2320955"}
{"ea_model_name":"uml:Class","ontology_node_id":"6225c183-3c13-422c-a8d8-
1f0ec0fafa6f","ontology_type":"types","ontology_name":"tuples","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_8EF1E496_F25F_4a92_B1F0_86B3F4C4821B"}
{"ea_model_name":"uml:Class","ontology_node_id":"69c44fbd-884d-40af-b4e7-
d785ed2a280f","ontology_type":"types","ontology_name":"types-instances","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_701F6B2E_DDFE_4990_83D9_28DA16D8DF99"}
{"ea_model_name":"uml:Class","ontology_node_id":"ae9be2de-e752-4b51-9db4-
3bf3a1720a8d","ontology_type":"types","ontology_name":"placeable types","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_6A588373_3536_4209_AF5A_97A198AA6F5D"}
{"ea_model_name":"uml:Class","ontology_node_id":"54ffe79f-b0c1-4d5b-b2da-
1e85c9b9e44f","ontology_type":"types","ontology_name":"Elements Powertype","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_5A5E0C66_D6B1_43c5_AC5F_2AF3E879E8EC"}
{"ea_model_name":"uml:Class","ontology_node_id":"ff663cbd-f70b-45c3-b2c2-
e3d4c3a5b907","ontology_type":"types","ontology_name":"Elements Powertype Powertype","creationTimeStamp":"Sat Apr 25 13:58:49
BST 2015","ea_model_node_id":"EAID_C2071EDE_C507_462e_AAA9_D29B21E826B0"}
{"ea_model_name":"uml:Class","ontology_node_id":"e1256e06-6a2e-49b3-9109-
10160b0f6837","ontology_type":"types","ontology_name":"NF Common Reserved Names","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_93EFC9D1_2435_4838_808F_EDDC16DAD3AA"}
{"ea_model_name":"uml:Class","ontology_node_id":"6ba7a3dd-fd2b-4d62-87f7-
7420cded7ccd","ontology_type":"types","ontology_name":"TNFA Common Reserved Names","creationTimeStamp":"Sat Apr 25 13:58:49
BST 2015","ea_model_node_id":"EAID_47A3C853_7EAB_4b6d_A807_72769FF92E7F"}
{"ea_model_name":"uml:Class","ontology_node_id":"aea0f94d-640e-431a-a226-
87e03bf95b26","ontology_type":"types","ontology_name":"TNFA Root Naming Space Names","creationTimeStamp":"Sat Apr 25 13:58:49
BST 2015","ea_model_node_id":"EAID_A08B8FAC_FA94_49f9_851B_278D92B106B9"}
{"ea_model_name":"uml:Class","ontology_node_id":"3e98ed13-9e8d-4f39-96b6-
b1f3a33fc6e5","ontology_type":"types","ontology_name":"Names","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_1231E44F_AE18_4bcf_86BF_6461771B8E04"}
{"ea_model_name":"uml:Class","ontology_node_id":"d28b9811-df33-4aa9-9a11-
3ffc63ed901f","ontology_type":"types","ontology_name":"Names Powertype","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_D81DBD63_526C_4991_9E21_A21D5FE02152"}
{"ea_model_name":"uml:Class","ontology_node_id":"9b577360-4501-4b60-8dfc-
53335a93abef","ontology_type":"types","ontology_name":"Naming Spaces","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_A2523529_2AF9_4291_A7FE_87CED21D4966"}
{"ea_model_name":"uml:Class","ontology_node_id":"ba5dea22-f216-4d5c-9fa8-
3bd21ff9e5d0","ontology_type":"types","ontology_name":"Representations","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_D6B28DE4_51F3_483a_A5BA_4787497D1C4B"}
{"ea_model_name":"uml:Class","ontology_node_id":"493b6367-737a-43cf-8050-
f00a17e29745","ontology_type":"types","ontology_name":"named by","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_62C7DA00_2A3B_4407_8C72_B73AB8E9FD1B"}
{"ea_model_name":"uml:Class","ontology_node_id":"c40aa5f7-d964-4916-ab6b-
1a402a8d59b5","ontology_type":"types","ontology_name":"tnfa exemplar types-instances","creationTimeStamp":"Sat Apr 25 13:58:49
BST 2015","ea_model_node_id":"EAID_153A8079_0840_4fab_B180_DA7CB086D00F"}
{"ea_model_name":"uml:Class","ontology_node_id":"a27fe4eb-00c6-4069-96e4-
cf1590e492db","ontology_type":"types","ontology_name":"This NFAgent","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_C0398330_6678_4209_A090_40F06E377C2F"}
{"ea_model_name":"uml:Class","ontology_node_id":"282df3a1-64ac-4ffd-b068-
b2566be6fd02","ontology_type":"types","ontology_name":"couples","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_AF78A21B_AB9D_4b6f_9FED_E5BE3E717B82"}
{"ea_model_name":"uml:Class","ontology_node_id":"d44fa6f2-016a-4f05-a439-
fd8a02b1ea21","ontology_type":"types","ontology_name":"super-sub-types","creationTimeStamp":"Sat Apr 25 13:58:49 BST
2015","ea_model_node_id":"EAID_6B5DC60B_DE01_4425_B533_10F6AB1836E9"}
```

References

{"ea_model_name":"uml:Class","ontology_node_id":"f2266402-008a-48eb-acb8-a8a182f2bf82","ontology_type":"types","ontology_name":"NF Alternate Names","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_BC8447ED_B4A6_480b_A71D_0B1014C54524"}
{"ea_model_name":"uml:Class","ontology_node_id":"eb0b9884-054a-41e8-80bc-610e3a2bec40","ontology_type":"types","ontology_name":"wholes-parts","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_DC772FB9_3B9F_4146_BDD8_D1238A61030C"}
{"ea_model_name":"uml:Class","ontology_node_id":"4ca93110-5e21-46eb-b9b9-5d79f458ee8a","ontology_type":"types","ontology_name":"Character Strings","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_B102B18B_C1D1_49c3_A06A_94C786DCB1A9"}
{"ea_model_name":"uml:Class","ontology_node_id":"b401b889-a85b-430e-b9f5-332f69dd5878","ontology_type":"types","ontology_name":"Signs","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_50EA419E_E494_4608_8785_8E4D3C0D03D8"}
{"ea_model_name":"uml:Class","ontology_node_id":"4f3ed4e9-5af7-4097-a55d-0aa3330e39a3","ontology_type":"types","ontology_name":"power-types-instances","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_2CEF161C_641E_4f06_861F_37B8176CC39D"}
{"ea_model_name":"uml:Class","ontology_node_id":"81eb2110-e616-4f8b-a2ae-6891aabadbc7","ontology_type":"types","ontology_name":"temporal stages of","creationTimeStamp":"Sat Apr 25 13:58:49 BST 2015","ea_model_node_id":"EAID_E9478A66_BB8F_41c5_8410_5F00A9127228"}

Foundational Relationship types (asserted via a single graph edge):

POWER_TYPES_INSTANCES
TYPES_INSTANCES
NON_WELL_FOUNDED_TYPES_INSTANCES
TNFA_REMOTE_TYPES_INSTANCES
SUPER_SUB_TYPES

Property keys

ea_model_rel_id (generated by EA UML design tool)
ea_model_node_id (generated by EA UML design tool)
ontology_rel_id (generated by 4D-SETL)
ea_model_name: UML model name i.e. class
ont_name: ontology element name i.e. Postcodes
ont_type_name: ontology type i.e. Type
ontology_name: domain ontology in which the element is defined
ont_rel_id: (generated by 4D-SETL)
ontology_node_id (generated by 4D-SETL)
creation_time_stamp: (time set for complete import batch to identify all element generated by a load operation)

Indexes

ont_id_index lucene {"type":"exact"}
foundation_ont_type_index lucene {"type":"exact"}
foundation_name_index lucene{"type":"exact"}
ea_model_node_id_indexlucene {"type":"exact"}
ont_rel_id_index lucene {"type":"exact"}
ea_model_rel_id_index lucene {"type":"exact"}

Appendix B: Company Node Listing

```
Start warm cache query - new code
****************************Start new path traversal1
Node[58145710]: JACEK SLUSARCZYK LTD
Node[58145585]: JACEK MUSIAL LTD
Node[58145553]: JACEK MNICH LTD
Node[58145521]: JACEK LOMNICKI LTD
Node[58145365]: JACEK KOPERA LTD
Node[58145272]: JACEK GOSCINSKI LTD
Node[58145240]: JACEK GORGON LTD
Node[57843217]: J&K BALANCE LTD
Node[57339929]: ITTC  BUSINESS LTD
Node[57306811]: ITI GLOBAL TRADE LTD
Node[57275896]: ITALY AND HOME INTERIORS LTD
Node[57180984]: ISOVENT (UK) LIMITED
Node[57068921]: ISAGO LTD
Node[56683178]: INTERPRO-WEB LTD
Node[56145475]: INSDA CONSULTING LTD
Node[55797484]: INEO TRAVEL LTD
Node[55498930]: IMS-TECHNICAL SERVICES LTD
Node[55377934]: IMMS WEST LTD
Node[55349531]: IMG ELITE CONSTRUCTION LTD
Node[54997888]: IDEA4U LTD
Node[54655209]: I.C.F-INTER CURRENCY FAIR LTD
Node[54117847]: HUBERT DROZDZAK LTD
Node[54054368]: HS PROFILE LIMITED
Node[53624516]: HOOQ STUDIO LTD
Node[52831628]: HIGHSTYLE CONSULTING LTD
Node[50637961]: HAJTO PROJECT MANAGEMENT LTD
Node[50173712]: GUTEK STEEL LTD
Node[49956443]: GRZEGORZ SZYMSKI LTD
Node[49956411]: GRZEGORZ STARY LTD
Node[49956255]: GRZEGORZ REMISZEWSKI LTD
Node[49956161]: GRZEGORZ MIELEC LTD
Node[49956005]: GRZEGORZ LANGE LTD
Node[49955787]: GRZEGORZ JAGODZINSKI LTD
Node[49951916]: GRYF GRAPHIC CENTER LIMITED
Node[49821683]: GROSS GLOBAL LTD
Node[49416779]: GREEN LOGIC ENTERPRISES LTD
Node[49396723]: GREEN HEAVEN ADVISERS LTD
Node[49353967]: GREEN COUNTY LTD
Node[49281847]: GREAT OPPORTUNITY LTD
Node[48796263]: GOSTUDENT LTD
Node[48620140]: GOMONEY365 LIMITED
Node[48556251]: GOLDMETRO LIMITED
Node[48458000]: GOLDCAST LTD
Node[48444267]: GOLD STANDARD (UK) LTD
Node[48416172]: GOINGSUPER LTD
Node[48415671]: GOING UP RENOVATIONS (UK) LIMITED
Node[48406037]: GOFUTURE LTD
Node[48334115]: GO INVENTORY LIMITED
Node[48269414]: GMH HOLDING LIMITED
Node[47988415]: GLOBAL EXPANDER LTD
Node[47727182]: GLAMOUR EMPIRE LIMITED
Node[47696339]: GLA CONTRACTORS LIMITED
Node[47086210]: GENERAL TRENDS UK LTD
Node[46785301]: GAVANNO LTD
Node[46161193]: G. BUCKINGHAM & COMPANY LIMITED
Node[44261388]: FLIS AND FLIS LIMITED
Node[43111081]: FELIX ACCOMMODATION (EALING) LIMITED
Node[43066003]: FEBE DESIGN LTD
Node[42911137]: FAST LOGISTICS LTD
Node[42877775]: FASHION EAST LIMITED
Node[42629028]: FAIRWIND INSTALLATION LTD
Node[42481067]: FACADES SOLUTIONS LTD
Node[42109204]: EXPERT CONNECTION LTD
Node[42058647]: EXIMIA CREATIVE LIMITED
Node[41735405]: EVEREST SOLUTIONS UK LIMITED
```

Node[41563930]: EUROPEAN MORTGAGES LTD
Node[41499595]: EUROMEDIATOR LIMITED
Node[41437491]: EURO SIMCON LTD
Node[41397564]: EURO ADV LTD
Node[40407618]: ENDOSCOPY SERVISAND SALES LTD
Node[40199066]: EMIL CHODACKI LTD
Node[40139056]: EMD CONSTRUCTION LIMITED
Node[40057027]: ELVENA LTD
Node[39369149]: EG-DESIGN (UK) LIMITED
Node[38929143]: ECO REDWOOD WINDOWS LIMITED
Node[38926407]: ECO PLANT SALES LIMITED
Node[38926376]: ECO PLANT LABORATORIES LTD
Node[38719308]: EASY- INVEST&CONSULTING LTD
Node[38589513]: EAST SHEEN LIMITED
Node[38439810]: EALING'S REAL ESTATE LTD
Node[38312370]: E.D.R. (UK) LIMITED
Node[38305876]: E.C. INTERIORS LTD
Node[37504020]: DREI- KA LTD
Node[37436961]: DRATEX LTD
Node[36929072]: DOMICA LTD
Node[36928917]: DOMI EUROPE LTD
Node[36695112]: DM WINDOW SOLUTIONS LTD
Node[36694367]: DM TRADE LTD
Node[36625833]: DK CHEM ORGANIC SYNTHESIS LTD
Node[35149010]: DEE SENSE LTD
Node[34793095]: DAYLI SERVICE LTD
Node[34759841]: DAWO LTD
Node[34743358]: DAWID SLUSARCZYK LTD
Node[34743295]: DAWID PUKALA LTD
Node[34259007]: DARIUSZ KOWALSKI LTD
Node[34258851]: DARIUSZ DEC LTD
Node[34258789]: DARIUSZ BUILDING SERVICES LIMITED
Node[34234921]: DARACON LTD
Node[33659305]: D&P LOGISTICS UK LTD.
Node[32817920]: CRUSTY BLOOMERS LIMITED
Node[32373574]: CREATIVE TECHNOLOGIES GLOBAL CONSULTING & SERVICES LTD
Node[32268753]: CRE8 NEW MEDIA LIMITED
Node[31395048]: COPERNICUS VENTURE CAPITAL LTD
Node[31192549]: CONTINENTAL STYLE LIMITED
Node[30437585]: COMFI-DENTAL CARE LIMITED
Node[14437544]: 4PICTURE LTD
Node[15643002]: A1 ORGANIC FOODS LTD
Node[15957532]: ABC ACTIVE LTD
Node[15974908]: ABC PROSPECT LTD
Node[16550267]: ACMA TRADE LTD
Node[16830855]: ADAM CZARNOTA LIMITED
Node[16840348]: ADAM MACIASZCZYK LTD
Node[16850226]: ADAM ZAK LTD
Node[17094969]: ADRIAN JAJKO LTD
Node[17387709]: AETHER DISTRIBUTION LTD
Node[17388178]: AETHER MINING LTD
Node[17454737]: AFORKLIFT LTD
Node[17527965]: AGA-LEP LIMITED
Node[17626750]: AGO CONSTRUCTION LIMITED
Node[17795684]: AIR COMANDOR LIMITED
Node[18003683]: AJT BUILDING CONTRACTORS LIMITED
Node[18517280]: ALFAMEDIO LTD
Node[18880102]: ALLS SCHOOL LTD
Node[18957555]: ALOECAMP LTD
Node[19088629]: ALSERVICES LTD
Node[19105864]: ALTAIR SOLUTIONS (UK) LTD
Node[19288783]: BBAIR LTD
Node[19728402]: BEEFLOW LTD
Node[19942268]: BELOPA LTD
Node[20132142]: BER SOLUTION LIMITED
Node[20298623]: BESPOKE WINDOWS LONDON LTD
Node[20312299]: BEST CHOICE (UK) LTD
Node[20322219]: BEST FORMATIONS LIMITED
Node[20359627]: BESTFURNITURE FOR YOU LTD
Node[20522533]: BGD SOLUTIONS (UK) LTD

```
Node[20635372]: BIG BASEMENT COMPANY LTD
Node[21979918]: BOGDAN KURSKI LIMITED
Node[21984931]: BOGUSLAW STEPKOWICZ LTD
Node[22294378]: BOTANIQUE LTD
Node[22586109]: BRADO LTD
Node[22760469]: BRAVE BROS LTD
Node[23506135]: BRONISLAW MUZYK LTD
Node[23985960]: BUCKINGHAM MORTGAGE SERVICES LIMITED
Node[24425969]: BUSINESS STAR LIMITED
Node[24487544]: BUTTERFLY BUSINESS LTD
Node[24858966]: C&B PARTNERS LTD
Node[24879467]: C&T SOLUTIONS LIMITED
Node[25557739]: CAMERON INVESTMENTS VENTURE LTD
Node[25825437]: CAPITAL BUSINESS LINKS LIMITED
Node[26651095]: CASTOR FIBER LTD
Node[26862864]: CBLOX LTD
Node[27151388]: CENTRAL BYTES LTD
Node[27547612]: CHANGE FOR BETTER LTD
Node[29858268]: CMJL INVESTMENT LTD
60.725302 milliseconds - warm cache test
```

Appendix C: Director Node Listing

```
==> | Node[58803769]{ontology_name:"18c1ad11-ad4b-448d-aff5-aaf20e8595a2",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"18c1ad11-ad4b-448d-aff5-aaf20e8595a2",label:"D00631496"} |
==> | Node[58803766]{ontology_name:"244e4270-77aa-443e-8057-16752b01006d",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"244e4270-77aa-443e-8057-16752b01006d",label:"D00631496"} |
==> | Node[58803743]{ontology_name:"eefb37da-4e5f-4a6a-a310-c8ecf70f9837",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"eefb37da-4e5f-4a6a-a310-c8ecf70f9837",label:"D00631428"} |
==> | Node[58803740]{ontology_name:"0d2f705f-c0ea-4d92-b9eb-c8b5cda451a2",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"0d2f705f-c0ea-4d92-b9eb-c8b5cda451a2",label:"D00631428"} |
==> | Node[58803717]{ontology_name:"efe9808e-6561-4eb1-8ab4-4731d86a9402",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"efe9808e-6561-4eb1-8ab4-4731d86a9402",label:"D00631273"} |
==> | Node[58803714]{ontology_name:"0409b2aa-82c8-420b-b0f3-81438d1aeda9",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"0409b2aa-82c8-420b-b0f3-81438d1aeda9",label:"D00631273"} |
==> | Node[58803690]{ontology_name:"07b923db-e3af-4547-bf56-9b72db875ef4",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"07b923db-e3af-4547-bf56-9b72db875ef4",label:"D00631771"} |
==> | Node[58803687]{ontology_name:"ee0984f4-c1f5-4e86-88e4-60ff0556bfb1",ontology_type:"company - officer
tuple",description:"This node represents a company officer temporal stage
tuple",ontology_domain:"directors",ontology_creation_date:"Tue Apr 14 10:22:00 BST 2015",utterance_date:"Tue Apr 14 10:22:00 BST
2015",ontology_unique_name:"ee0984f4-c1f5-4e86-88e4-60ff0556bfb1",label:"D00631771"} |
==> 8 rows
==> 37 ms
```