Dealing with inconsistent and incomplete data in a semantic technology setting

Nwagwu Honour Chika

A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the degree of Doctor of Philosophy

August 2015

Abstract

Semantic and traditional databases are vulnerable to Inconsistent or Incomplete Data (IID). A data set stored in a traditional or semantic database is queried to retrieve record(s) in a tabular format. Such retrieved records can consist of many rows where each row contains an object and the associated fields (columns). However, a large set of records retrieved from a noisy data set may be wrongly analysed. For example, a data analyst may ascribe inconsistent data as consistent or incomplete data as complete where he did not identify the inconsistency or incompleteness in the data. Analysis on a large set of data can be undermined by the presence of IID in that data set. Reliance as a result is placed on the data analyst to identify and visualise the IID in the data set.

The IID issues are heightened in open world assumptions as evident in semantic or Resource Description Framework (RDF) databases. Unlike the closed world assumption in traditional databases where data are assumed to be complete with its own issues, in the open world assumption the data might be assumed to be unknown and IID has to be tolerated at the outset. Formal Concept Analysis (FCA) can be used to deal with IID in such databases. That is because FCA is a mathematical method that uses a lattice structure to reveal the associations among objects and attributes in a data set.

The existing FCA approaches that can be used in dealing with IID in RDF databases include fault tolerance, Dau's approach, and CUBIST approaches. The new FCA approaches include association rules, semi-automated and automated methods in FcaBedrock. These new FCA approaches were developed in the course of this study. To underpin this work, a series of empirical studies were carried out based on the single case study methodology. The case study, namely the Edinburgh Mouse Atlas Gene Expression Database (EMAGE) provided the real-life context according to that methodology. The existing and the new FCA approaches were used in identifying and visualising the IID in the EMAGE RDF data set.

The empirical studies revealed that the existing approaches used in dealing with IID in EMAGE are tedious and do not allow the IID to be easily visualised in the database. It also revealed that existing FCA approaches for dealing with IID do not exclusively visualise the IID in a data set. This is unlike the new FCA approaches, notably the semi-automated and automated FcaBedrock that can separate out and thus exclusively visualise IID in objects associated with the many value attributes that characterise such data sets. The exclusive visualisation of IID in a data set enables the data analyst to identify holistically the IID in his or her investigated data set thereby avoiding mistaken conclusions.

The aim was to discover how effective each FCA approach is in identifying and visualising IID, answering the research question: "How can FCA tools and techniques be used in identifying and visualising IID in RDF data?" The automated FcaBedrock approach emerged to be the best means for visually identifying IID in an RDF data set. The CUBIST approaches and the semi-automated approach were ranked as 2nd and 3rd, respectively, whilst Dau's approach ranked as 4th. Whilst the subject of IID in a semantic technology setting could be explored further, it can be concluded that the automated FcaBedrock approach best identifies and visualises the IID in an RDF thus semantic data set.

Acknowledgements

I am grateful to Dr. Simon Andrews and Dr. Simon Polovina for the guidance and support they have given me. I am also thankful to Mr. Constantinos Orphanides for his assistance and directions. Also, I thank the CUBIST project team particularly Dr. Frithjof Dau for all the assistance which I received in the course of this study.

Dedication

For my wife Linda and sponsors Mr. and Mrs. Afam-Anadu

Table of Contents

Tables

Figures

Candidate Statement

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university, and to my best knowledge and belief, this thesis contains no material published or written by another person, except where due reference is made in the thesis.

Abbreviations

CUBIST	Combining and Uniting Business Intelligence with Semantic Technologies
CWA	Closed World Assumption
CAEX	Computer Aided Engineering Exchange
DBMS	Database Management System
EMAGE	The Edinburgh Mouse Atlas Gene Expression Database
EMAP	The e-Mouse Atlas Project
ETL	Extract Transform and Load
FCA	Formal Concept Analysis
HTML	HyperText Mark-up Language
IID	Inconsistent or Incomplete Data
OWA	Open World Assumption
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
ST	Semantic technology
TS	Theiler Stage
URI	Universal Resource Identifier
W3C	World Wide Consortium
XML	Extended Mark-up Language
SPARQL	Simple Protocol and RDF Query Language

Chapter 1: Introduction

1.1 Background to the Research

This chapter introduces the rationale and objectives of this research. This section presents a brief technical background to the research which is focused on the need to explore novel ways for dealing with the problem of inconsistent and incomplete data in semantic databases. Section 1.2 of the chapter presents the rationale for the research. Section 1.3 states the research objectives and section 1.4 concludes with an overview of this thesis.

Semantic technologies are becoming popular and increasingly important for storing and processing data. A semantic technology (ST) uses techniques that support and exploit the semantics of information, as opposed to syntax and structural/schematic issues, to enhance existing information systems (Sheth 2005). The semantics of information refers to the meaning associated with information, whilst syntax and structural/schematic issues refer to the heterogeneity challenges existing in traditional data processing systems. STs allow the meanings of and associations between objects and attributes which constitute information in a knowledge base to be known and processed at execution time (Sheth and Ramakrishnan 2003). Resource Description Framework (RDF) data, for example, are processed in ST setting. The RDF is a framework for representing information on the Web. Unlike relational data which are stored and processed in a traditional database such as Oracle and MySQL, RDF data are stored and processed in a semantic database. The processing of RDF in ST setting is explained in details in chapter 3 of this thesis.

RDF database tolerates IID. It adopts Open World Assumption (OWA) principles which implicitly assume that a knowledge base may always be incomplete (Hitzler et al., 2011 p.131). For example, consider the following statement about Philip in an OWA database: "Philip passed an English test." Under the OWA database, there is incomplete knowledge about stored data. Statements such as "Philip failed an English test" or "Philip did not take an English test" are allowed to exist in the same database. Consequently, contradictory (inconsistent) data can exist in such databases.

Conversely, a Closed World Assumption (CWA) framework holds that all relational statements in a knowledge base are completely listed, so that what is currently not known to be true in the sense of not being included in the list is false.

Consequently, an OWA framework is much more likely to generate new statements or accommodate new meanings based on the facts in the framework.

Since RDF adopts OWA principles, it does not prevent anyone from making assertions that are nonsensical or inconsistent with other statements or the world as people see it (Klyne et al. 2004). Hence, RDF data may always be inconsistent or incomplete. Unlike an RDF triple store[1], a traditional database can adopt CWA principles, which imply that the associated knowledge base is complete. Even so, traditional databases may contain IID. CWA and OWA are discussed in details in Chapters 2 and 3, respectively.

As explained further below, this research explores novel ways for dealing with Inconsistent or Incomplete Data (IID) in RDF (hence semantic) databases, for which there is relatively fewer works compared to research on IID in traditional database settings.

1.2 Rationale for the Research

The presence of IID in semantic or traditional databases can lead to inaccurate inferences or inaccurate conclusions. This results to enormous cost on enterprises. Polovina (2013) notes that many enterprises risk business transactions based on information systems that are incomplete or misleading, given that 80-85% of all corporate information remains outside of the processing scope of such systems. Also, the Bloomberg Trading Solutions (undated) note that the complete cost of inconsistency is unquantifiable and it can include cost of accounting errors, cost of multiple feeds and all required labour to address the inconsistency.

In traditional databases, missing data is represented by the null and this remains a controversial issue as evident in (Lano 2014; Waraporn and Porkaew 2008; Zimányi and Pirotte 1997). The use of null to represent missing attribute values from various sources can lead to inconsistencies when the data are integrated. An integrated data set is evident in databases such as data warehousing, data fusion and data exchange systems. Bleiholder and Naumann (2008) identify IID as a problem in an integrated data set. Also, Decker and Martinenghi (2011) demonstrate how inconsistencies can emerge in a traditional database as a result of violations of integrity constraints. Integrity constraints and null are discussed in greater details in chapter 2.

Data set stored in a traditional or semantic database is queried to retrieve record(s) in a tabular format. Such retrieved records can contain many rows where

[1] A database that stores RDF data

each row contains an object and the associated fields (columns). However, a large set of records retrieved from a noisy data set may be wrongly analysed by a data analyst. For example, a data analyst may ascribe inconsistent data as consistent or incomplete data as complete where he did not identify the inconsistency or incompleteness in the data. The data analyst needs to visually identify the IID in such retrieved record set as to properly analyse the data. But visually identifying the IID in a large tabular record set can be difficult. This is because a record set displayed in a tabular format does not enable an easy visualisation of the IID. The data analyst visually compares the different rows in the record set to fine contradictory records or records without particular fields. There is therefore a need for an easier means to exclusively visualise the IID in a large and noisy record set.

There is a wide body of work on dealing with IIDs in traditional database settings. Among these works are (Decker and Martinenghi 2011; Grant and Hunter 2011; Bleiholder and Naumann 2008; Cortés-Calabuig et al. 2005; Hunter and Konieczny 2005; Ma et al. 2007). However, in semantic database settings there is not so much research on dealing with IIDs. Notable among the few existing works are (Dau 2013a; Dau 2013b; Melo et al. 2013). There is therefore a need to research how IID in RDF data can be dealt with.

The aim of this work is to discover ways by which Formal Concept Analysis (FCA) can be used to deal with IID in a semantic technological setting. FCA is a semantic analysis technique and has been shown in a recent European Union (EU) CUBIST[2] project to be useful and appropriate for the analysis of data in RDF triple store. It involves a mathematical method which uses a lattice structure to reveal the associations among objects and attributes in a data set. It uses the lattice as a visualisation framework to explore correlations, similarities, and anomalies in a data set (Carpineto and Romano 2004, p.15). Therefore, it can be used to explore IID in RDF data sets. Hence, this work uses an indicative case study to answer the research key question:

> "How can FCA tools and techniques be used to identify and visualise IID in RDF data?"

The research uses an RDF triple store (Owlim[3]) and FCA analysis tools such as FcaBedrock[4], Concept Explorer[5] and In-Close[6] as its semantic technological setting. It

[2] www.cubist-project.eu

[3] https://www.ontotext.com/owlim

builds on works of Dau (2013a, 2013b) and Melo et al., (2013) which deal with IID in RDF data set, in order to develop novel FCA approaches that improve upon some inadequacies in the existing ones. In other words, the research explores existing and additional approaches in which FCA tools and techniques can be used to deal with IID existing in RDF data. The research uses a single case study to evaluate the performance of the identified FCA approaches.

1.3 Research Objectives

The specific objectives of the research are as follows:

1. To understand IID issues and how they are dealt with in a traditional technology setting.
2. To understand IID issues in ST setting
3. To investigate existing approaches in dealing with IID in ST setting.
4. To propose FCA as an appropriate and effective technique for dealing with IID in ST setting.
5. To build on existing FCA approaches and develop better novel approaches.
6. To apply existing and new FCA approaches to an indicative case study
7. To compare and evaluate the usefulness and effectiveness of the different FCA approaches.

1.4 Overview of this Thesis

This thesis is presented in 10 chapters. Chapter 1 presents the rationale and objectives of the research as stated above.

Chapter 2 explains what IID is. It describes in more details, the concept of CWA and the different types of IID. Null and integrity constraints are among the identified sources of IID in traditional database. Also, the use of optional fields in data entry forms is

[4] http://sourceforge.net/projects/FcaBedrock/

[5] http://sourceforge.net/projects/conexp/

[6] http://sourceforge.net/projects/inclose/

identified as a source of incomplete data. The various approaches used in addressing IID in traditional databases are described in this chapter.

Chapter 3 provides an overview of ST, semantic database, RDF and the concept of OWA. IID in semantic database are explained. The chapter also identifies and explains existing approaches that are used in dealing with IID in a semantic database setting.

Chapter 4 describes the classical FCA approach. It explains how formal concepts are derived from formal context and how they are displayed in a lattice structure. The challenges of classical FCA approach in identifying and visualisation IID are also explained.

Chapter 5 explains the research methodology, including how FCA is used to address the issues of IID in an indicative case study - the Edinburgh Mouse Atlas Gene Expression Database (EMAGE). The chapter explains case study research with emphasis on single case study. The suitability of EMAGE as a case study is explained. The chapter also identifies FCA as a research method. The criteria used in selecting EMAGE as a case study are also explained.

Chapter 6 explains in detail the EMAGE, a database of gene expression data in the developing mouse embryo, and an accompanying suite of tools to search and analyse the data. The chapter particularly explains how the e-Mouse Atlas Project (EMAP) is used in EMAGE. In addition, it describes the EMAGE RDF data set. The causes of IID in the EMAGE data set are also identified and explained.

Chapter 7 explains how the RDF query language (SPARQL) can be used to retrieve IID from a semantic database. It provides comprehensive details about existing and new FCA approaches to IID in RDF data set. Examples are also used to illustrate how these approaches are applied. This prepares the ground for applying the approaches to the case study.

Chapter 8 therefore explains how these FCA approaches are used to identify IID existing in an EMAGE data set.

Chapter 9 compares and evaluates the effectiveness of the FCA approaches.

Chapter 10 outlines the main contributions of the research to knowledge, concludes this thesis, and discusses the challenges of the study and related future works.

Chapter 2: Inconsistent and Incomplete Data (IID) Issues in Traditional Databases

2.1 Introduction

This chapter explains what IID in a database is. It specifically discusses IID in traditional database. Section 2.2 explains the different types of IID in traditional databases. Section 2.3 discusses IID in traditional databases in which CWA is adopted. The sources of IID in traditional databases are explained in section 2.4 and the various ways of dealing with the IID in a traditional database are explained in section 2.5. Section 2.6 concludes the chapter by outlining the key messages and findings.

IID exists in a database when data do not conform to the rules governing its design. A data set in a traditional database can contain objects (G) and associated attributes (M) which have many values (W). Consequently, an object ($g \in G$) that is associated with an attribute ($m \in M$), can be inconsistent or incomplete in a traditional database. An object can be inconsistent when there is a contradiction in the values of the attribute such that $w \subseteq W$ is associated with A and $\neg A$. An object is incomplete when it has some but not all of its required values.

IID often exist in a traditional (relational) database in which data are integrated from different sources. This is because in such integrated databases, different values from different sources can be associated with an attribute of an object. Contradictory values can be associated with an attribute of an object, while some attributes of an object may not have the required values. IID are not restricted to integrated databases. A single source data set may also contain IID, for example; syntactic errors, missing values, unique value violation, out of range values and functional dependency violations are identified as different ways in which IID can exist in single source data sets (Fürber and Hepp 2010). Some examples of the different traditional databases where IID are likely to exist include; data exchange (Hernich et al. 2011; Afrati and Pavlaki 2008; Libkin 2006), data fusion (Khaleghi et al. 2013; Bleiholder and Naumann 2008; Kumar et al. 2007), and data warehousing (Kimball and Caserta 2004; Calvanese et al. 2001; Rahm and Do 2000; Chaudhuri and Dayal 1997). These domains where IID thrive in traditional databases are explained in section 2.3 below.

2.2 Types of Inconsistent or Incomplete Data (IID)

IID can occur in traditional database for different reasons; for example, it can occur because a value of an attribute is not available or because the value is integrated from a data source with a different schema. In this section, the various classifications of IID are explained.

There are different classifications of IID; for example, Codd (1986) classifies missing data in a traditional database as either 'missing and applicable' or 'missing and inapplicable'. Kim and Seo (1991) classify data conflict in multidimensional databases as 'wrong data' and 'different representations for the same data'. Bleiholder and Naumann (2008) explain that a data conflict is present in a data set if for the same real-world object e.g. a student, semantically equivalent attributes from one or more sources do not agree on its attribute value; for example source 1 reporting "23" as the student's age, and source 2 reporting "25". They describe two kinds of data conflict: (a) uncertainty about the attribute value, caused by missing information and (b) contradictions, caused by different attribute values. Other classifications of missing data in traditional databases are presented in works such as *(*Waraporn and Porkaew 2008; Zimányi and Pirotte 1997; Codd 1979; Gottlob and Zicari 1988*)*.

In this work, IID are classified as either binary or analogue. A binary form of IID exists when the same object is associated with attribute values that have opposite meanings. An example of attribute values that have opposite meanings is a gene which is 'detected' and 'not detected'. An analogue type of IID exists when an object is associated with attribute values that are slightly contradictory. An example of attribute values that are slightly contradictory is a gene which is associated with weak and medium expression levels. Finkelstein (2000) explains that "*In many cases inconsistencies reflect slips and minor errors or possibly delayed commitments which are relatively easy to resolve. Some inconsistencies however reflect serious conflicts with substantial knock-on consequences and may involve substantial negotiation."* Such minor errors can be attributed to analogue IIDs while serious conflicts can be attributed to binary IIDs. Inconsistency in gene expression data is also classified as either binary or analogue in McLeod and Burger (2011).

2.3 Close World Assumption (CWA) and Inconsistent or Incomplete Data (IID)

The underlying principles of a database can either prohibit IID such as in CWA or allow the existence of IID such as in OWA. This section describes how CWA is adopted in

traditional databases and the consequences of such implementation on the stored data.

The CWA expresses the communication agreement that an atom that does not appear in the database is false (Cortés-Calabuig et al., 2005, Denecker et al. 2010). It works on the principles that a knowledge base has complete information about every data in the domain. Even where there is no proof of a positive ground literal, the negation of that literal is assumed true (Reiter 1982).

Traditional databases adhere to the principles of CWA by representing unknown or missing values with null and resisting non-conforming data through the use of integrity constraints. The works of (Codd 1979) provides a formal treatment of missing values by null under the unknown semantics (Gottlob and Zicari 1988). Zimányi and Pirotte (1997) explain that null have been most widely used to model incomplete information under CWA. But the null is an ambiguous representation of a missing value. This is because a missing value can be unknown, not available, not existing, or not applicable. The use of the null to represent all these instances in a data set causes indefiniteness in the set of data. Consequently, inaccurate results can be drawn from a set of data in which nulls are used in representing missing attribute values.

Null cannot explicitly represent missing values where data from different sources are integrated into a data set. Its use in traditional database may result into IID when data are integrated from multiple sources. For example, an attribute value of an object from source '*A'* can be assigned the null value. Also, a different value can be assigned to the same attribute of the same object in source 'B'. This will result into an uncertainty when these data are integrated into a single database. Such uncertainty in traditional database is described in Bleiholder and Naumann (2008) as the conflict between a non-null value and one or more null values that are all used to describe the same property of an object. This type of conflict can be assessed as analogue IID (see section 2.2). Moreover, some partial knowledge available to the data user such as the inapplicability of a value in an attribute will be lost when such knowledge is represented by null (Gottlob and Zicari 1988). The works of (Waraporn and Porkaew 2008; Gottlob and Zicari 1988; Zimányi and Pirotte 1997) explain the lapses in representing missing values with null.

Also, IID can be prohibited from a traditional database through the use of integrity constraints. Integrity constraints are statements declared in the database schema which express semantic properties that are meant to be invariably satisfied by the stored data across state changes (Decker and Martinenghi 2011). The use of integrity constraints entail that the traditional database would contain partial information. This is because in traditional database, integrity constraints are used to restrict non-

conforming data from entering into the database. These restrictions imply that the traditional database does not guarantee the full correctness or completeness of its data with respect to the part of the real world that it models (Grefen 1993; Motro 1998). More so, the adoption of CWA and the use of integrity constraints do not efficiently apply to all domains. Cortés-Calabuig et al., (2005) explain that in a context of integrated data sources, CWA is inherently inappropriate since the consideration of a certain data source as a single and complete representation of the world either completely discards the other sources of information or causes contradictions among them. Consequently, IID can thrive in databases such as data exchange, data fusion, and data warehousing. These databases are briefly described below:

2.3.1 Data Exchange

Data exchange is the problem of taking data structured under a schema called the source schema, and transforming it into data structured under another schema, called the target schema (Kolaitis 2005; Fagin and Kolaitis 2005). It involves materialising the source data in a target data set. The importance of data exchange is evident in a circumstance where the transfer of data between independently created applications is required. Such independently created applications are likely to have different schemas, integrity constraints, and data format.

IID can thrive in data exchange databases. For example, in data exchange the null may be used to represent missing values of source data in the target data set. This practice as described in this chapter is likely to lead to IID. Hernich et al. (2011) and Libkin (2006) note that when the presence of incomplete information in the target source of data exchange setting is ignored then certain answers from the data exchange will be wrong. Also, certain answers from the data exchange will be wrong where CWA is not enforced (Afrati and Pavlaki 2008; Hernich et al. 2011; Libkin 2006).

2.3.2 Data Fusion

Data fusion involves the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation (Bleiholder and Naumann 2008). This involves fusing multiple records from different data sources and also resolving associated inconsistencies. The aims, definitions and techniques of data fusion vary from domain to domain (Boström et al. 2007) but, in general, data fusion faces the challenge of structuring data from multiple schemas into a data set. Bleiholder and Naumann (2008) identify the main problems in data fusion as the

detection of equivalent schema elements in different sources (schema matching) and the detection of equivalent object descriptions (duplicate detection) in different sources to integrate data into one single and consistent representation.

Data fusion systems are widely used in various areas such as sensor networks, robotics, video and image processing, and intelligent system design (Khaleghi et al. 2013). IID are identified in data fusion in works such as in (Khaleghi et al. 2013; Bleiholder and Naumann 2008; Kumar et al. 2007).

2.3.3 Data Warehousing

A Data Warehouse is a set of materialized views over the operational information sources of an organization, designed to provide support for data analysis and management's decisions (Calvanese et al. 2001). A data warehouse can include data from an organisation’s operational information sources which may span over a long period. It can involve integrating data from different sources into a database. Such database can be subject to changes where new sources are integrated and old sources may be deleted.

Basically, data warehousing involves the Extract, Transform and Load (ETL) processes. Data are extracted from identified operational information sources (internally or externally), transformed and loaded into the end target file. These extracted data can be dirty or noisy such that they are inconsistent, have missing values or other anomalies. Consequently, they will be transformed to meet the data warehouse requirements such as the integrity constraints in the data warehouse. Also, not all the extracted data might be needed for a particular managerial decision, so only the required data will be loaded into the target file for a particular analysis.

Issues of IID are obvious in ETL processes. For instance, data cleaning is an important activity in the ETL processes of a data warehouse and it deals with detecting and removing errors, missing data and inconsistencies in the data set (Kimball and Caserta 2004; Calvanese et al. 2001). Data cleaning in ETL processes aims at ensuring that the data in the data warehouse are correct, consistent and complete. For example, Kimball and Joe (2004 p. 166) point out that during the loading phase of the ETL processes, it is important to recognise the same dimensional entity across multiple source systems and resolve the conflicts in overlapping descriptions. Works such as (Kimball and Caserta 2004; Calvanese et al. 2001; Rahm and Do 2000; Chaudhuri and Dayal 1997) identified IID as a challenge in data warehouses.

2.4 Sources of IID in Traditional Databases

This section discusses null, integrity constraints, and optional fields in data entry forms as some of the sources of IID in a traditional database. These sources of IID in traditional databases are explained as follows:

2.4.1 Nulls

Incomplete data contain missing attribute value(s). The null is used in traditional databases to represent missing attribute values as explained above. A missing attribute value can be unknown, unavailable, not existing or not applicable. The representation of these values with null can be problematic in an integrated data set. Lano (2014) explains how the explicit use of nulls complicates system specification and verification by introducing indefiniteness into expressions, encouraging the use of hard-to-verify specification styles and complicating the logic used for reasoning about systems and models. This is because null can misrepresent the meanings of missing attribute values. It can also cause inconsistency in the meaning of missing value when an attribute value in a source data is compared to the represented value in an integrated data set.

2.4.2 Integrity constraints

Integrity constraints are statements that must always be true for the stored data and for any update to the database. Integrity constraints are usually provided by the Database Management System (DBMS). A DBMS can be defined as a program that helps to manage your database. A check for null in a primary key is an example of an integrity constraint which a DBMS provides. In some instances, third party software applications provide such integrity constraints where the constraints are not supported by the DBMS (Decker and Martinenghi 2011).

An application of integrity constraint introduces incompleteness into the traditional database. The database is left with only partial information to answer user's queries. Calì et al., (2013) explain that in a global schema containing integrity constraints, the query processing is intimately connected to the notion of querying incomplete databases. They note that "*when the global schema is expressed in the relational model with integrity constraints, even of simple types, the problem of incomplete information implicitly arises*". A global schema provides an integrated and virtual view of the different sources of data. The application of integrity constraints in traditional databases, leads to incomplete information in the database.

Furthermore, inconsistencies in a traditional database may emerge as a result of violations of integrity constraints. Decker and Martinenghi (2011) explain that integrity constraints may be violated when new constraints are added without being checked for violations by legacy data. Integrity control of a database may be turned off temporarily such as when uploading a backup for which a total check would last too long. The integrity of the data in a database may also deteriorate by migrating to the DBMS of a different vendor, since the semantics of integrity constraints tends to be proprietary. When an integrity constraint in a DBMS is violated, inconsistency prevails in the database. Also, properly designed constraints introduce incompleteness in the database by not allowing data that may violate it. IID will always exist in a traditional database when integrity constraints are applied.

2.4.3 Optional fields in data entry forms

A data entry form with an optional field can cause IID in a traditional database. Chaudhuri and Dayal (1997) identify optional fields in data entry forms as significant sources of inconsistent data. A data entry form may contain optional or mandatory fields. The asterisk '*' by the side of a field in a data entry form is often used as a distinction between optional and mandatory fields where the later is indicated by the asterisk.

An optional field enables the capture of non-mandatory data from a form user. There are issues of IID when a data entry form contains optional fields. This is because the form user may enter contradictory or even nonsensical values through an optional field. He may also leave the optional field blank which will result to missing data at the backend of the data entry form.

There are other possible causes of IID in traditional databases. For example, the lack of a consistent vocabulary in traditional database can cause IID in the database. Berners-Lee et al. (2001) explain that traditional knowledge representation systems are typically centralized, requiring everyone to share exactly the same definition of common concepts such as "parent" or "vehicle." A representation system whose data are from different sources that do not abide by a common vocabulary is likely to house inconsistent data. Also, Denecker et al. (2010) note that ignorance about the domain, lack of proper maintenance, incomplete migration, and accidental deletions of tuples are some of the reasons for the presence of incomplete knowledge in a database.

This section has outlined some sources of IID in traditional databases. Even so, the traditional database engines such as MySQL and Oracle applications are still used

to process both the single sourced and the integrated data sets. Section 2.5 explains how IID are dealt with, in such traditional database.

2.5 Approaches used to Deal with IID in Traditional Databases

There are several approaches that can be used to deal with the IID in traditional databases. These approaches can be classified as follows:

2.5.1 Resolving/Repairing IID

Resolving IID involves identifying the IID and replacing or repairing the IID. It involves applying some minimal change to IID as explained in Wijsen (2006). In data warehouse for example, data cleaning provides an avenue to detect and remove errors and inconsistencies in a data set (Rahm and Do 2000). Some of the approaches used in data cleaning are documented in (Kimball and Caserta 2004; Rahm and Do 2000).

There are also tools developed for detecting and repairing data which violate integrity constraints (Fan et al. 2008; Fazzinga et al., 2006; Raman and Hellerstein 2001). In addition, the works of (Flesca et al., 2010; Bertossi et al., 2008; Cong et al. 2007; Bravo and Bertossi 2006) demonstrate how inconsistent data can be identified and resolved through the use of relational data queries. Missing attribute values in incomplete data are resolved in relational database by either replacing them with null or with other meaningful representations. The works of (Waraporn and Porkaew 2008; Zimányi and Pirotte 1997; Gottlob and Zicari 1988; Codd 1979) demonstrate how missing values in relational databases can be replaced.

2.5.2 Preventing IID

The use of integrity constraints (see Section 2.4 above) can prevent certain IID from entering the database. For example, the primary keys in most commercial relational databases are specified as non-Null constraints by the DBMS. Such an integrity check prevents records that have missing data in the field used as the primary key from being admitted into the database. Works such as (Grefen 1993; Codd 1979) demonstrate how constraints are used to prevent IID.

2.5.3 Reasoning with IID

Sometimes, it is necessary to allow IID into a database. For example, IID should be allowed into a tax database to identify fraudulent data. On such note, Decker and Martinenghi (2011) suggest that inconsistencies are often unavoidable or even useful such as in diagnosis or mining fraudulent data. They propose ways by which some inconsistencies can be tolerated through integrity tolerance check. Hunter and Konieczny (2005) explain the importance of tolerating inconsistent data to include the avoidance of losing information about some facts in a database. It is therefore important to have good reasoning methods for dealing with a database which contains IID. An example of a good reasoning technique for retrieving information from a noisy database is consistent query answering as evident in (Fuxman et al., 2005; Chomicki et al., 2004; and Arenas et al., 1999).

2.6 Key Messages and Findings

IID may not mainly be resolved, prevented, evaluated or reasoned with. They should be managed in accordance with the requirements of the domain[7] in which they occur. For example, in circumstances in which IID is tolerated, the IID should be reasoned with rather than repaired or replaced. Also, it may not be beneficial to use traditional database as a data processing engine. A database that adheres to OWA can be more effective in dealing with IID. For example, it is acknowledged in (Bonifati et al. 2008; Halevy et al. 2003) that designing a database which can adhere to CWA is difficult or even impossible to achieve in a Peer to Peer (P2P) database system. Thus, the integrated data sets may more effectively be processed by a database system which is based on the OWA. Chapter 3 explores the semantic database, its adherence to OWA, IID in RDF data, and related concepts.

This chapter has explained the meanings and types of IID and how the integrity constraints and null in traditional databases are used, among others. It also explained the sources of IID in traditional database and the ways by which it can be dealt with. Related concepts which underpin this study such as CWA versus OWA frameworks were reviewed. The chapter noted particularly that IID is an important and challenging

[7] Some approaches, for example [Denecker et al. 2010] have different assumptions on the domain of the database (Local Closed World Assumptions) while others adhere to CWA or OWA.

concept in data processing and analysis; hence there is great need to properly manage it in order to avoid inaccurate data analysis. This informs the focus of this work in using RDF and FCA approaches to deal with IID problems in ST settings, as explored further in chapter 3 and 4.

Chapter 3: Inconsistent and Incomplete Data (IID) in Semantic Technology (ST) Setting

3.1 Background

This chapter describes how IID in Resource Description Framework (RDF) data are processed in a semantic technology (ST) setting. It begins by describing what ST is. Section 3.2 explains the OWA principles and IID in RDF triple store. It also explains the RDF data and entailment rules. Section 3.3 identifies and explains the different ways by which IID in RDF triple store are dealt with in ST settings. The chapter is concluded by outlining the key messages and findings in section 3.4.

The ST plays a huge role in the processing and analysis of RDF data. Hendler (2009) describes web 3.0 as semantic web technologies integrated into or powering large-scale web applications. He characterised the features of semantic technological applications to include applications that can integrate web data resources and have increased use of and support for the languages developed in the World Wide Consortium (W3C) Semantic Web Activity[8]. The RDF provides a base for semantic web technologies[9]. In this work, the Owlim RDF triple store, and the FCA tools and techniques namely the FcaBedrock, ConExp, and In-Close are considered.

When an RDF data set containing IID is processed in an RDF triple store, the IIDs are not prohibited from the data store unlike in a traditional database (see chapter 2). This is because an RDF triple store adheres to OWA principles. Also IID can be inferred in a triple store through the application of RDF entailment rules. Unlike the traditional data representation, RDF enables data from multiple sources to be integrated into a data set without jeopardising their meaning. Consequently, IID can be meaningfully represented, stored, and processed in RDF database. The use of RDF enables the representation of the meaning and associations in resources[10]. Section 3.2 below explains the principles behind RDF representations.

[8] www.w3.org/2001/sw

[9] http://semanticweb.org/wiki/Tool

[10] Resources can be anything which is described

3.2 Open World Assumption and IID

A database which does not assume that its data is complete and which does not use integrity constraints to exclude data that do not conform to its schema is said to adhere to OWA. This section explores OWA and IID in RDF triple store.

In OWA, the absence of information is an indication of lack of knowledge. Also, a statement cannot be inferred to be false in OWA, even when there is a failure to prove it (Sirin and Tao 2009). For example, the storing of the statement "Philip passed an English test" does not mean that "Philip failed an English test" is false. This is because Philip may have written the English test several times.

An RDF data can be inconsistent or incomplete. Unlike the traditional database where constraints might ensure the consistency of its data, RDF triple store do not prohibit IID as recommended by (Klyne et al., 2004). In addition, RDF data set may have issues such as missing schema, and evolving data, as identified in (Scheglmann et al. 2013). Nevertheless, IID can be inferred through RDF entailment rules in an RDF triple store. The application of entailment rules in a triple store may enable new knowledge to be inferred from existing knowledge. Even so, not all the missing data in a data set can be inferred by a triple store. A proper understanding of RDF data and entailment rules is needed to understand the issues of IID in an RDF triple store. RDF data and entailment rules are therefore explained in sections 3.2.1 and 3.2.2 respectively.

3.2.1 Resource Description Framework (RDF) Data

RDF is a framework for representing information in the web. Its data (RDF data) are statements made about resources in the form of triples. A triple (an RDF statement) consists of a subject, predicate (attribute) and an object. Figure 1 shows a graphical representation of 2 triples as outlined in Table 1 below.

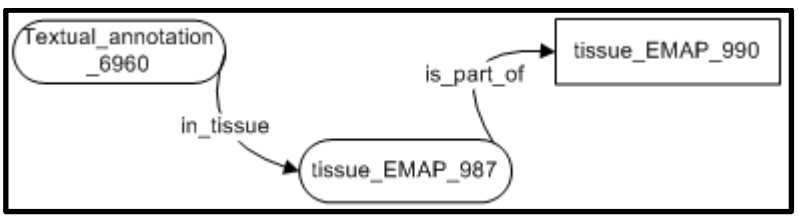

Figure 1: Graphical representation of an RDF data showing 2 triples

In Figure 1, a resource such as *textual_annotation_6960* and its predicate such as *in_tissue* are depicted. Ideally, a resource in a triple is uniquely represented by the Universal Resource Identifier (URI). For example, the resource *Textual_annotation_6960* can be uniquely identified within the project *http://www.cubist-project.eu/HWU#*. However, this work uses plain words to depict resources in a triple.

Table 1: Triples of the graph in Figure 1

Subject	Predicate	Object
textual_annotation_6960	*in_tissue*	*tissue_EMAP_987*
tissue_EMAP_987	*is_part_of*	*tissue_EMAP_990*

RDF data can contain more than 2 triples. Figure 2 shows an example of an RDF data that contains 4 triples. This is depicted in Table 2 below.

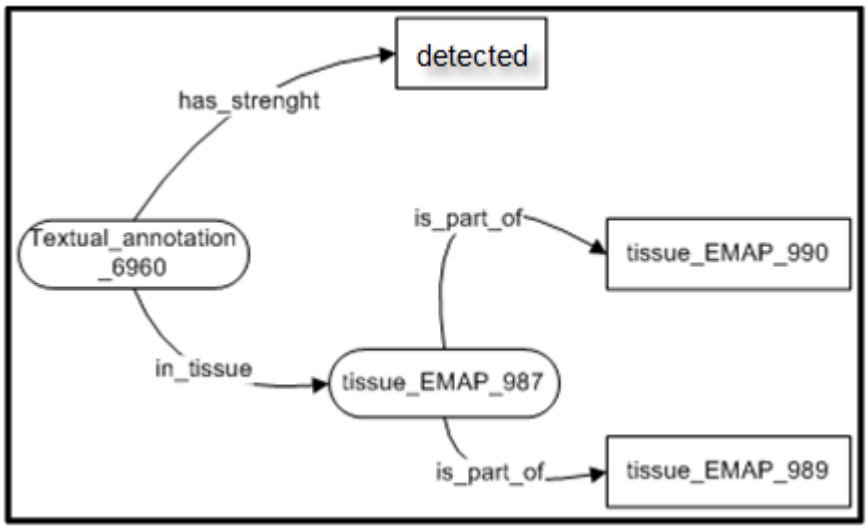

Figure 2: RDF Graph illustrating RDF data with more than 2 triples

Table 2: Triples of the graph in Figure 2

Subject	Predicate	Object
Textual_annotation_6960	in_tissue	tissue_EMAP_987
tissue_EMAP_987	Is_part_of	tissue_EMAP_990
tissue_EMAP_987	Is_part_of	tissue_EMAP_989
Textual_annotation_6960	has_strength	level_detected

A comparison of Table 1 and Table 2 reveals that the RDF data in Table 1 is incomplete. It can be seen that the object *tissue_EMAP_987* is only associated to 1 attribute (predicate) value in Table 1 such that the attribute '*is_part_of'* is associated to the value '*tissue_EMAP_990*'. This is unlike in Table 2 where the attribute '*is_part_of*' of the object *tissue_EMAP_987* is associated to *tissue_EMAP_990* and *tissue_EMAP_989*. These observations are easier to identify in the associated figures. Also it is possible to have an RDF data which has contradictory attribute values such as detected and not detected. These are examples of instances of IID in an RDF data set.

RDF is a framework which can be used as a modelling language and also as a data representation format. It uses Resource Description Schema (RDFS)[11] or Web Ontology Language (OWL)[12] as its modelling languages. RDFS, for example, provides data-modelling vocabulary such as *rdfs:Class*, *rdfs:subClassOf*, *rdfs:label*, and *rdfs:domain* which are used to describe the properties of a subject or an object of a triple. Both the RDF data and its metadata (such as RDFS data) can be stored as triples in the same data set. An example of how a modelling language can be used in an RDF data set is illustrated in Table 3 below.

Table 3: A Triple illustrating how RDFS may be used

Subject	Predicate	Object
Textual_annotation_6960	rdfs:label	assay no 6960

In Table 3, the description of *textual_annotation_6960* is achieved through the use of RDFS construct such as *rdfs:label* (see Table 3 above). More descriptions can be

11 http://www.w3.org/TR/rdf-schema/

12 http://www.w3.org/TR/owl-semantics/

added to the data where necessary. This may be achieved through the use of *rdfs:Class* or *rdfs:subClassOf* constructs in RDFS.

The use of RDF and its modelling languages (RDFS or OWL) in data representation enhances data integration (combining data from different sources). This is because RDF uses URI which retains the uniqueness and semantics of the represented data even when they are integrated with other data. Also, the use of RDF does not result in issues of interoperability of data as evident in Extended Mark-up Language (XML), Hyper Text Mark-up Language (HTML) or relational data format (Berners-Lee et al., 2001; Bizer et al., 2009). Consequently, RDF data from various sources can be integrated into a single data set without losing the semantics.

RDF is used in representing web data as evident in projects such as DBpedia[13] and CUBIST. However, a fundamental issue with RDF data is its tolerance of IID. An object in RDF data can be associated with many attribute values; this can result in IID in the data. For instance, there is inconsistency in an object which is associated with contradictory attribute values. On the other hand, incompleteness will exist in an object which does not have a required value. Nevertheless, RDF triple stores can use the entailment rules to infer missing attribute values or deduce inconsistency as a means of addressing its IID challenges. Entailment and entailment rules are explained in section 3.2.2 below.

3.2.2 Entailment

A triple store is a semantic technological setting that stores RDF data (triples) and also reasons over the stored triples. It can also infer new triples into its RDF data set through entailment rule(s). Entailment describes two graphs which are equal in all aspects in that every assertion made about one RDF graph may be made with equal truth about the other graph (Powers 2003). As explained in Hayes and McBride (2004)

> *If A entails B, then any interpretation that makes A true also makes B true, so that an assertion of A already contains the same "meaning" as an assertion of B; one could say that the meaning of B is somehow contained in, or subsumed by, that of A. If A and B entail each other, then they both "mean" the same thing, in the sense that asserting either of them makes the same claim about the world.*

13 http://dbpedia.org/About

Table 4 shows an example of an RDFS entailment rule "*rdfs11"* as outlined in Hayes and McBride (2004).

Table 4: An example of RDFS entailment rule

Rule Name	If E contains:	then add:
rdfs11	uuu *rdfs:subClassOf* vvv vvv *rdfs:subClassOf* xxx	uuu *rdfs:subClassOf* xxx

A triple store adds an entailed triple to a graph when the graph entails any larger graph. This is because a graph entails any larger graph that is obtained by applying entailment rule(s) to the original graph. In Table 4, the presence of a graph whose pattern is "*uuu rdfs:subClassOf vvv*" and "*vvv rdfs:subClassOf xxx*" results to the addition of the triple "*uuu rdfs:subClassOf xxx*" to the graph. This provides a rule by which graphs of this pattern are entitled. The W3C RDF* recommendations include a list of standard entailment rules. For example Hayes and McBride (2004) outline a standard set of simple entailment rules for RDFS. Also, a standard set of entailment rules in OWL is outlined in W3C website[14].

RDF triple stores such as Owlim, Jena[15], Oracle Semantic database and Sesame adhere to OWA. A semantic database provides the semantic technological setting that can process the inconsistent or incomplete information in an RDF data set. As explained in Klyne et al. (2004), "*RDF does not prevent anyone from making assertions that are nonsensical or inconsistent with other statements or the world as people see it. Designers of applications that use RDF should be aware of this and may design their applications to tolerate incomplete or inconsistent sources of information*". Unlike in traditional databases (see chapter 2), where data are accepted only if they do not violate the integrity constraints, RDF triple stores do not resist IID. Also, the entailment rules can be used to tolerate, infer or even identify the IID in RDF database as explained in section 3.3 below. Even so, there is still a need to identify other means of dealing with the tolerated IID in a triple store. Section 3.3 explores the existing approaches for dealing with the IID in a semantic technological setting.

14 http://www.w3.org/TR/owl-semantics/rdfs.html

15 http://jena.sourceforge.net

3.3 Existing Approaches to IID in ST Setting

The existing approaches for dealing with IID in an RDF triple store can involve identifying, evaluating, visualising, analysing or reasoning with the IID. These measures will empower the information analyst with the accurate knowledge about the nature of the IID. This knowledge will enable the information user to avoid inaccurate conclusions.

IID can be identified and measured in an investigated data set. The works of (Fürber and Hepp 2010; Melo et al., 2013; Péron et al., 2011; Drumond et al., 2012; Sertkaya 2009) explain different ways of using SPARQL to identify IID in RDF data set. Also Nwagwu (2013) explains how inconsistent data in RDF data set can be evaluated through the use of SPARQL queries. But the identification of IID through SPARQL queries involves presenting the results in a table and such presentations may be difficult to analyse. In fact, voluminous tables can be difficult to perceive or visualise. Nevertheless, Dau 2013a; Dau 2013b; and Andrews and McLeod 2013 present approaches that enable the visual analysis of the IIDs in RDF triple store. These works uses FCA techniques as a means of dealing with IID in a data set.

Another means of dealing with IID in a triple store is through the use of rules. Hayes and McBride (2004) explain entailment rules as a means of inferring RDF data and also identifying its inconsistencies. RDF processing engine such as Owlim uses rule based approach to deal with IID (Stoilov and Bishop 2012).

This section identifies the rule based approach, the query based approach, and the combination of the query based approach with FCA techniques as three main documented approaches for dealing with IID existing in RDF data. The section below, explains these approaches.

3.3.1 Rule Based Approach used to Deal with IID in RDF Data

A triple store adds an entailed triple to a graph when the graph entails any larger graph. RDF data processing engines use entailment rules to deal with IID existing in its data set. Stoilov and Bishop (2012) explain how Owlim-SE infers missing data and how it identifies inconsistent data through entailment rules.

In Owlim-SE, entailment rules provide a means of adding triples to an entailed graph. Also, entailment rules provide a means of detecting inconsistency in the triple store. Rules that are not supposed to add any triple to a graph indicate inconsistency when there are triples that conform to the pattern exhibited by the rule. For example, whenever there are triples that conform to the pattern exhibited in an Owlim

consistency checking rule, an error message is sent to a standard output such as a log file (Stoilov and Bishop 2012). Table 5 below, shows a rule that is designed to identify inconsistent data.

Table 5: Owlim consistency check

Rule Name	**If Entailment rule contains:**	**Standard output:**
consistency Check	x owl:sameAs y x owl:differentFrom y	Error message

As depicted in Table 5, whenever the reasoner identifies a graph that contains the triples "*x owl:sameAs* y" and "*x owl:differentFrom y*", an error message is sent to a standard output notably an error file.

Rule based detection approach to IID automatically identifies IID existing in an RDF data set but the following lapses can be associated with the approach:

- The error file will be difficult to visualise where there are many errors.
- Analysing the IID in the error messages can be difficult especially since the identified inconsistent data are mostly stored as log (text) files as explained in (Stoilov and Bishop 2012).
- There are semantic and syntactic differences among data in different data sets. As a result, differences exist in the rules of the different data sets for identifying their IID. It will be impossible to articulate and stipulate all the rules of all data sets as entailment rules in a reasoner.
- The IID approach in Owlim triple stores demands that inferred data are integrated automatically into the data set. Also, inconsistent data are generated as error messages in a log file (Stoilov and Bishop 2012). This approach makes the identification, visualisation, evaluation and analysis of IID difficult to achieve.

3.3.2 Query Based Approach used to Deal with IID in RDF Data

SPARQL is a query language for RDF data. The use of SPARQL can enable the retrieval of IID from an RDF triple store. For more details about SPARQL, the works of Quilitz and Leser (2008), Power (2003), and DuCharme (2011) are recommended.

The query based approach used in dealing with IID provides a means by which the information analyst can interact with the data set and address issues of IID such as identifying and evaluating the IID in a data set. This approach enables the information analyst to creatively analyse the data through SPARQL features and display results in a tabular format.

Abele et al. (2013) explain how SPARQL queries can be used to validate Computer Aided Engineering Exchange (CAEX) data set. This validation process involves using SPARQL queries to check for inconsistent data in CAEX data set. Quilitz and Leser (2008) and Dau (2013a) explain how the *optional* and the *union* keywords can be used to retrieve IID from an RDF data set. Fürber and Hepp (2010) explain how to identify data quality problems in single source scenarios through SPARQL queries. Their focus is on syntactic errors, missing values, unique value violation, out of range values and functional dependency violations. Also, Nwagwu (2013) shows how SPARQL queries can be used in evaluating the data from an RDF triple store.

But there are challenges with visually analysing the query results displayed in a table, especially where there are many rows and columns. One of such challenges is the inability to visualise at a glance, the relationships existing in IID across many rows of a table. As a result, identifying and visualising objects with similar contradictions or similar incompleteness becomes a difficult task. Also, voluminous tables may be difficult to perceive.

3.3.3 Combining Query Based approach with FCA Techniques

FCA techniques such as fault tolerance and interactive exploration enable the analysis and visualisation of the data in a database. These data are retrieved through the use of SPARQL queries from the RDF database. However, such data must first be transformed into a formal context and subsequently into a formal concept, to visualise the inconsistency or incompleteness in it. Formal context and the concept lattice are explained in chapter 4.

Dau (2013a, 2013b) and Melo et al. (2013) provide different approaches that involve retrieving a result set through SPARQL queries and visualising IID in the result set through FCA tools and techniques. Dau (2013a) explains how to identify and visualise IID through the use of his developed SPARQL to formal context tool ‘SPARQLcontext creator’. Dau (2013b) explains how to reason with a noisy and incomplete data set through the application of fault tolerance technique on a retrieved result set.

Also, the CUBIST project developed many FCA techniques for dealing with IID. CUBIST is an acronym for Combining and Uniting Business Intelligence with Semantic Technologies. This project was funded by the European Commission under the 7th Framework Programme of ICT, Topic 4.3: Intelligent Information Management. It ran from 1st October 2010 to 31st September 2013. Among CUBIST's achievements are the development of a tool (CUBIST) and approaches for dealing with inconsistent and incomplete RDF data. The CUBIST approach is based on "the querying of ontology data which is then converted to a formal context in a process transparent to the user" (Melo et al. 2013). Melo et al. (2013) explain the IID processing capabilities of CUBIST. CUBIST applies fault tolerance, and interactive exploration techniques when dealing with IID in an RDF data set.

Another FCA technique for identifying incomplete data is attribute exploration. Attribute exploration has been ingeniously applied in distinctive areas such as ontology completion, security checks, and web data. For example, through attribute exploration, the missing data of an incomplete ontology can be deduced by querying the domain expert. Sertkaya (2009) explains how attribute exploration is applied by OntoComp application to complete ontology attributes. However, Baader et al. (2007) note that attribute exploration does not adhere to the semantics of the OWA. Chapter 7 of this work examines the attribute exploration in greater details.

Combining Query based approach with FCA techniques addresses the problems associated with the Rule based approach by providing the means to identify and visualise IID in an RDF database. The application of FCA techniques as evident in Dau (2013a, 2013b) and Melo et al. (2013) empower the data analyst with the ability to visualise the IID existing in his investigated data set. But these FCA techniques do not separate out the IID from its data set, hence, they do not exclusively visualise IID existing in a large data set. Work such as (Hunter and Konieczny 2005; Grant and Hunter 2011; Finkelstein 2000; Dau 2013b) demonstrate the need to identify, and analyse IID existing in a data set. There is also need to exclusively visualise the IID in a data set especially when dealing with a large and noisy data set. This is to avoid erroneous conclusions. It is explained in Nwagwu and Orphanides (2015) and Nwagwu (2014) that an FCA approach that visualises all the data in a data set might not clearly display the IID in the associated lattice. This may hinder the ability of the data analyst to visualise the IID in the lattice where the investigated record set is large. Consequently, the data analyst may assess an inconsistent data as consistent when he did not visualise any contradiction associated with the data. This problem is further described with illustrating examples in Chapter 4.

3.4 Key Messages and Findings

RDF is a semantic framework whose data are processed in ST setting. It adopts the OWA principles. Unlike traditional data model, RDF preserves the meaning associated with its represented data even when the data are integrated from multiple data sources. Similar to traditional database, RDF database faces IID challenges.

This chapter has explored issues of IID in RDF data and how it is presently dealt with in a semantic technological setting. It identified the lapses in the current approaches used in dealing with IID in ST setting. It identifies that the rule base, query based, and the combination of the rule based with FCA approaches are the existing methods of dealing with IID in RDF database. It further recognises that combining query based approach with FCA techniques provides a more robust approach that can be used to deal with IID in RDF database than the rule or query based approaches. Nevertheless, combining query based approach with FCA techniques does not exclusively identify or visualise the IID in a large and noisy data set.

Chapter 4 formally introduces the FCA. Also, it further explains the challenges associated with the use of FCA in dealing with IID in RDF data set.

Chapter 4: Formal Concept Analysis (FCA)

4.1 Introduction

This chapter introduces Formal Concept Analysis (FCA) which underpins this research. Following this introduction, section 4.2 examines the classical FCA approach and its challenges in dealing with IID. Examples are used to illustrate the classical FCA approach.

Formal Concept Analysis (FCA) was introduced by Rudolf Wille in (Wille 1982). It is a mathematical method which uses the concept lattice as a formalism to explore correlations, similarities, refinements, anomalies, or even inconsistencies (Carpineto and Romano 2004). It provides a data processing approach by which data are analysed by conceptually clustering objects with respect to a given set of attributes and visualising the set of clusters in a lattice structure.

FCA is used in this work to analyse and visualise IID in RDF data. It provides better analysis and visualisation approaches than traditional data analysis techniques such as (x, y) plots, linear and bar-charts, histogram, and pie charts. For instance, traditional data analysis and visualisation techniques are rendered ineffective when a data set contains tens, hundreds or thousands of dimensions and when the data set does not have natural mapping to the display space (Keim et al. 2008; Keim 2001). This is unlike in FCA which provides data analysis and visualisation approaches for exploring correlations, similarities, refinements, anomalies, and inconsistencies.

FCA presents more advanced data analysis and visualisation techniques which have been shown in the CUBIST project to be useful and appropriate for the analysis of data in RDF triple store. Also, its tools and techniques can be used to analyse large data set. For example, Andrews and Orphanides (2010) describe how the FcaBedrock and the In-Close can be used to analyse a formal context which contains more than 220,000 formal concepts. Stumme et al. (2002) show how the algorithm '*tatanic*' is used in the computation and visualisation of 32,086 formal concepts. The formal context and formal concept are explained in section 4.2. FCA tools and techniques have been used to identify IID existing in an RDF data set. For example, in (Dau 2013b; Andrews and McLeod 2013; and Melo et al. 2013), different FCA tools and techniques were used to identify and visualise IID existing in RDF data. Even so, it can still be challenging to explore IID especially when dealing with a large and noisy data set.

4.2 The Classical FCA Approach

FCA begins with a formal context. A formal context can be described as a single-valued context (*G, M, I*). *G* represents a set of objects represented along the rows of a table, *M* is a set of attributes represented along the columns of the same table and *I* is a binary relation between *G* and *M* ($I \subseteq G \times M$) represented by a cross on the cell intersecting a particular object with its corresponding attribute. The symbols *G*, *M*, and *I* are used in this work in accordance with how it is used by Rudolf Wille in (Wille 1982) where FCA was first introduced. These symbols are also used in most FCA literatures such as (Carpineto and Romano 2004; Wolff 1993; Burmeister and Holzer 2005).

Table 6 below, describes some gene expressions in some tissues of a mouse. A cross in a cell is used to depict a gene expressed in the corresponding tissue of the mouse. For example, Otx2 is detected in the node, organ system, neural ectoderm, and future brain. The absence of a cross in a cell depicts that the corresponding tissue does not express the corresponding gene expression. It can also mean that it is not known whether this object has the attribute or not (Wolff 1993; Burmeister and Holzer 2005). Table 6 describes a set of attributes (Otx2_detected, Otx2_not_detected, Hoxb1_detected, and Hoxb1_not_detected) and a set of objects (node, mesoderm, organ system, neural ectoderm, future brain and limb).

Table 6: Formal context of gene expressions in tissues

	Otx2_detected	**Otx2_not_detected**	**Hoxb1_detected**	**Hoxb1_not_detected**
node	X			
mesoderm				X
organ system	X		X	
neural ectoderm	X	X		
future brain	X			
limb				

A formal context as depicted in Table 6 describes a set of objects and a set of attributes. If $x \in A$ and $y \in B$ (where *A* is a set of objects in a formal context *K* and *B* is a set of attributes in *K*) then xIy holds for the objects and attributes in *K* implying that the object *x* has the attribute *y* and the attribute *y* is a feature of the object *x.* Any of this pair (A, B) in a formal context is called formal concept. The set 'A' is called the *extent* while the set 'B' is called the *intent* of the formal concept. Ganter et al. (2002) explain that "*A (formal) concept of a formal context (G, M, I) is a pair (A, B) with* $A \subseteq G$, $B \subseteq M$,

A' = B and B' = A. The sets A and B are called the extent and the intent of the formal concept (A, B), respectively." The extent of a formal concept contains all objects that have the attributes of the intent while the intent contains all attributes shared by the objects of the extent. An example of such a pair in Table 6 is {(*mesoderm*), (*Hoxb1_not_detected*)}.

Mathematically

A formal context $K = (G, M, I)$

Where

G = a set of objects

M = a set of attributes

I = a binary relation between G and $M \rightarrow I \subseteq G \times M$

The extent 'A**'** (set of objects) and the intent 'B' (set of attributes) of each formal concept in a formal context K can be defined as follows:

(A, B) with $A \subseteq G$, $B \subseteq M$, $A = B'$, and $B = A'$ where

$A' = \{y \in Y \mid \text{for each } x \in A \text{ then } (x, y) \in I\}$,

$B' = \{x \in X \mid \text{for each } y \in B \text{ then } (x, y) \in I\}$,

A and *B* are called the extent and the intent of the formal concept (A, B) and they define a formal concept in a formal context. Some of the formal concepts identified from Table 6 are as listed below:

a. {(*mesoderm*), (*Hoxb1_not_detected*)}
b. {(neural ectoderm), (*Otx2_not_detected*, *Otx2_detected*)}
c. {(*organ system*), (*Otx2_detected*, *Hoxb1_detected*)}
d. {(node, organ system, future brain, neural ectoderm,), (*Otx2_not_detected*)}

When a set of objects are described by a set of attributes, it becomes easy to identify which attribute is associated with which object or which object is associated with which attribute. On this note, IID can easily be identified by noting the objects which are associated with contradictory attributes or the objects which are not associated with the set of attributes examined. In Table 6, *neural ectoderm* is associated with contradictory attributes- (*Otx2_not_detected*, *Otx2_detected*). The tissue 'l*imb'* is not associated with any attribute. The neural ectoderm is inconsistent while the limb is incomplete.

The identification of IID from a large formal context can be easier when the concept lattice is built from the formal context. A concept lattice enables the visualisation of objects with similar attributes, objects without attributes, and other associations such as their hierarchical order. Section 4.2.1 below describes how a concept lattice is built from a formal context.

4.2.1 Building a Concept Lattice from a Formal Context

A concept lattice is a structured diagram which is composed of one or many nodes. A concept lattice can be built from a formal context through the use of FCA tools such as ToscanaJ, Toscana, or ConExp. Building a concept lattice involves mining formal concepts in a formal context and displaying them hierarchically in a lattice structure.

As noted above, a concept (A, B) is evident in a formal context when

$A \subseteq G, B \subseteq M, A = B', \text{ and } B = A'$

A and *B* are called the extent and the intent respectively of the formal concept (*A, B*). For a context (G, M, I), a concept X = (A, B) is less general than a concept Y = (C, D) (or $X \leq Y$) if $A \subseteq C$ or equivalently, $D \subseteq B$. i.e $X \leq Y$. Also, if there is no other concept Z such that $Z \neq X$, $Z \neq Y$, $X \leq Z \leq Y$, then X is called a lower neighbour (or subconcept) of Y, and Y is called an upper neighbour (or superconcept) of X (Kuznetsov and Obiedkov 2001). The set of all identified formal concepts of a formal context K existing in such a hierarchical order, forms a complete lattice $\underline{B}(K)$.

The concept lattice displays objects with similar attributes (concepts) in a lattice structure. This enables easier identification and visualisation of such objects with similar attributes. Figure 3, (see below) is a concept lattice built from Table 6 (see above) through the use of ConExp application. A concept lattice is easy to read. The simple reading rule of a concept lattice as described in (Wolff 1993) is that an object g has an attribute m if and only if there is an upward leading path from the circle labelled by “g” to the circle labelled by “m”. Consequently, it can easily be visualised from Figure 3 that *Otx2* is detected in *organ system*, *future brain*, *neural ectoderm* and *node*.

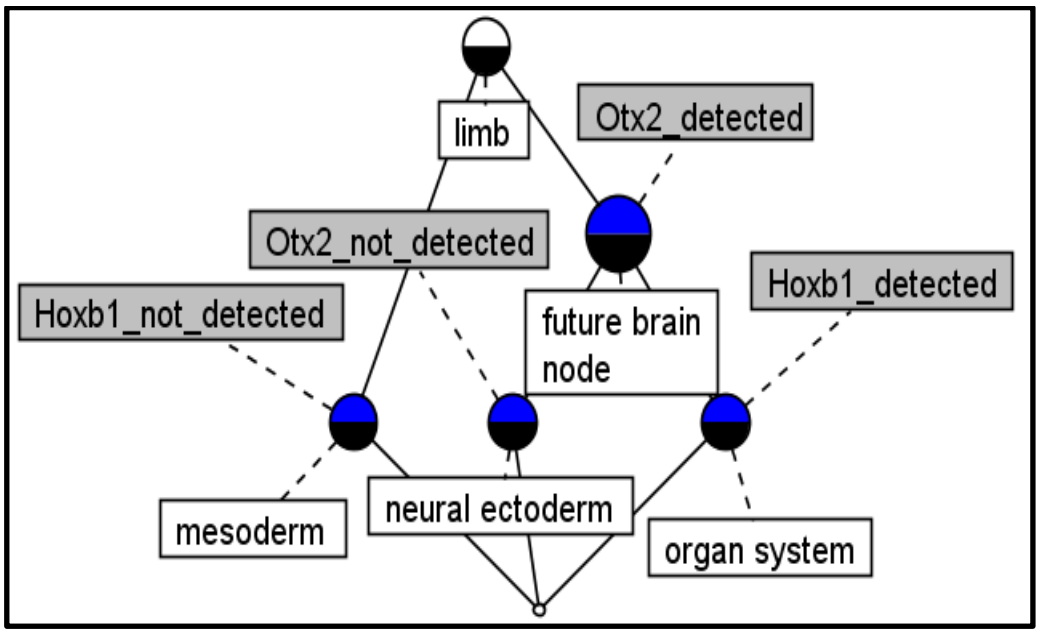

Figure 3: Concept lattice showing gene expressions in tissues of a Mouse

FCA enables the visualisation of IID data in a lattice structure. For example, Otx2 is both *detected* and *not detected* in *neural ectoderm* hence *neural ectoderm* is inconsistent. The incomplete data can easily be identified in extents which are not associated with any attribute or intents which are not associated with any object. Incomplete data can be displayed at the bottom or top of the lattice. In Figure 3 above, the incomplete datum is displayed at the topmost node. The object *limb* is incomplete because it is not associated with any attribute.

Concept lattice provides an easy means of identifying IID but there are challenges with its readability when it displays concepts from a large formal context. A description of such challenges is provided in section 4.2.2 below. Also, a description of how data from a many-valued context such as data retrieved from querying a database can be transformed to a formal context is provided in the same section.

4.2.2 Conceptual Scaling and Visualisation Challenges

As earlier noted, FCA begins with a single-valued context (*G, M, I*) but data can be represented as a many-valued context (*G*, *M*, *W*, *I*) where *G* is a set of objects, *M* is a set of attributes, *W* is a set of attribute values, and *I* is a ternary relation between *G*, *M* and *W*. In a many-valued context (*g*, *m*, *w*) ∈ *I*, the attribute m takes the value w for the object g (Messai et al., 2008). Record sets retrieved from a database such as RDF databases can often exist as many-valued context. In FCA, a many-valued context is

transformed to a single-valued context (formal context) through the use of conceptual scaling.

Conceptual scaling is used to transform a many-valued context (*G, M, W, I*), to a single-valued context (*G*, *M*, *I*). This is often achieved by replacing every many-valued attribute in the recordset by the corresponding attribute-value pairs, with each object being described by one attribute-value pair, per many valued attribute (Carpineto and Romano 2004; Annoni and Brüggemann 2008). An example of such transformation is shown in Figure 4 below.

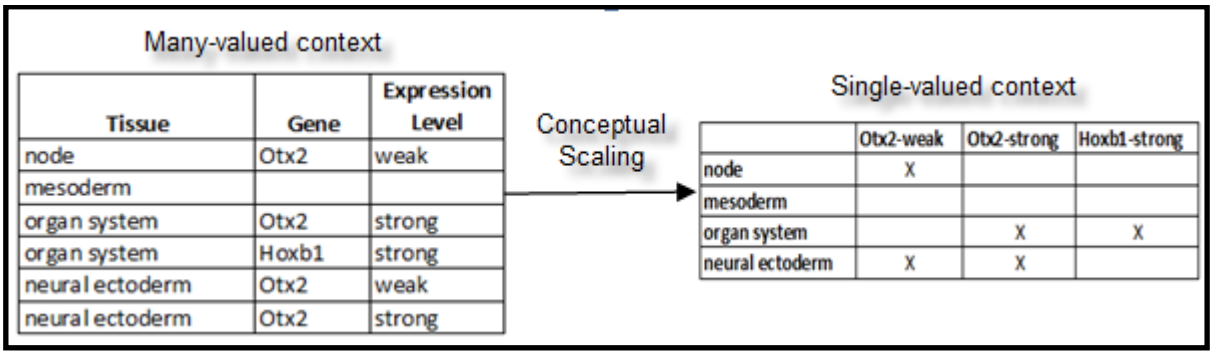

Figure 4: An example of a transformation from many-valued to a single-valued context

Conceptual scaling provides a global view of the object-attribute-value relations in a many-valued context by replacing every many-valued attribute in the context with the corresponding attribute-value pairs. In Figure 4, the single-valued context is derived by describing each object of the many-valued context with the corresponding attribute-value pair. For example, the attribute *Otx2* is associated with the values *weak* and *strong* in the many-valued context of Figure 4. A transformation of this many-valued attribute to single-valued attribute will result to the attribute-value pair '*Otx2-weak'* and '*Otx2-strong'*. Conceptual scaling has the advantage of presenting a global view of its represented data. However, when conceptual scaling is applied to large data sets, the resulting formal contexts are more than often unmanageable. A complex and unreadable concept lattice is built when such context is visualised in a lattice structure.

A many-valued context of only a modest size can result into a formal context containing hundreds or even thousands of formal concepts when conceptual scaling is applied. The corresponding concept lattice will be difficult to visualise and also unreadable. Figure 5 below, is composed of 27 concepts. A view of the concept lattice can help us to understand how unreadable a concept lattice can become, when they are built from a large formal context. Figure 5 is very unreadable as compared with Figure 3 which has only 6 concepts.

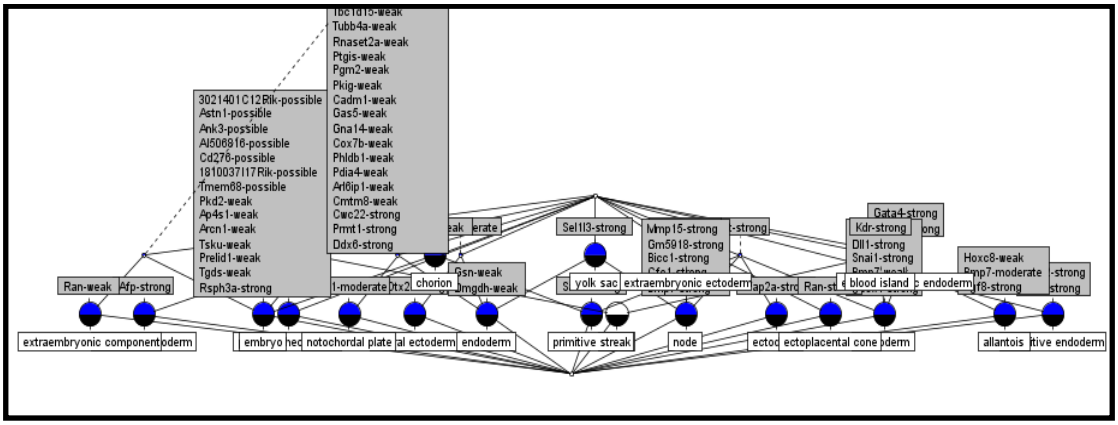

Figure 5: Formal concept illustrating visualisation issues

Carpineto and Romano (2004 p. 26) and Dau (2013b) note that, an increase in the size of formal context could result in an exponential rise in number of formal concepts. A concept lattice which has many formal concepts will also have many crossing edges. These factors (many formal concepts and many crossing edges) will hinder the identification or visualisation of attributes of interest by the data analyst. Hence, visualising IID through the use of the classical FCA approach is challenging when dealing with a large formal context.

4.3 Key Messages and Findings

This chapter described the classical FCA approach and how it is used in building a concept lattice from a formal context. A formal context or a single-value context is noted as the starting point of FCA and concept lattice can be built through the application of context visualisation tool such as ConExp on a formal context. It is explained in this chapter that the display of formal concepts of a formal context in a concept lattice can enable the identification and visualisation of the IID existing in the formal context. Also, most data sets such as the query results from an RDF database exist in many-valued form and conceptual scaling is used in transforming such many-value data set to a formal context. However, the use of conceptual scaling can produce a large context from even a modest sized many-valued context. This results to an unreadable concept lattice when such many-valued contexts are transformed. Consequently, identifying and visualising the IID in the corresponding concept lattice can be challenging.

Nevertheless, there are some FCA approaches specifically developed to deal with IID. These approaches are classified in this work as the existing and new FCA techniques. They are presented in chapter 7. Chapter 5 presents the research methods used in this work.

Chapter 5: Research Methodology

5.1 Introduction

Empirical studies were conducted on issues of IID in the Edinburgh Mouse Atlas Gene Expression Database (EMAGE). The studies looked into the causes of IID in EMAGE and how IID in EMAGE can be dealt with through the use of FCA tools, FCA techniques, and Owlim semantic database. The empirical studies were undertaken to meet the research objectives which are outlined in section 1.3 of this thesis. These research objectives include understanding existing ways by which IID in RDF data set are dealt with in a semantic technological setting and proposing novel approaches for dealing with IID, among other things. This chapter explores the approaches adopted in these studies. It focuses on how the research methods were applied.

The empirical studies show the existing FCA and semantic database approaches for dealing with IID do not exclusively identify or visualise IID existing in an RDF data set. This is explained in chapters 3, 4 and 7. An exclusive visualisation of IID in a data set is very important especially when dealing with a large and noisy data set. For example, when a data set contains thousands of attributes and attribute values, a data analyst using a non-exclusive IID visualization approach may ascribe an inconsistent data as consistent where he did not visualise the contradictory data. The exclusive visualisation of IID involves separating out the IID from the noisy data set and visualising it in a concept lattice. This reduces the complexity associated with visualising IID when the entire formal concepts from a large data set are depicted in a lattice structure. This is because fewer concepts would be presented in the lattice unlike when dealing with the whole formal concepts from the noisy data set. Stumme et al. (2002) recommend that strategies (other than arbitrarily reducing the context) for dealing with large concept lattices should be considered.

Again, the main research question of this work is *"How can FCA tools and techniques be used to identify and visualise IID in RDF data?"* Accordingly, FCA research approaches are used in this work. Also, in chapters 2 and 3, it is seen that IID is more likely to exist in an integrated data set. Consequently, an investigation into how IID can be dealt with should focus on an integrated data set. Also, investigating IID in a real life context will ensure that genuine issues relating to the IIDs in the investigated context are identified. As a result, the Edinburgh Mouse Atlas Gene Expression Database (EMAGE) is investigated in this work. It is the only 'case' studied in this work.

Yin (2009) explains that descriptive or explanatory questions, studying a phenomenon within its real-world context and evaluating a phenomenon create relevant opportunities for applying the case study method as a research method. The suitability of EMAGE as a single case study is explained in section 5.2. The case study and FCA research approaches are the different research methods used in this work. The use of these methods to examine particular phenomena (IID) produced mixed results sets-qualitative and quantitative results. The results from the FCA methods notably the automated and semi-automated FcaBedRock methods were validated by verifying them through the EMAGE search options. The EMAGE search options are described in section 6.4

The subsequent sections of this chapter explore how these research approaches were used in this work. Section 5.2 explores the single case study and the suitability of EMAGE as a single case study. Section 5.3 explain the FCA research approach and section 5.4 explains some alternatives to the approaches adopted in this work, the challenges of the applied research approaches and the various ethics considered in this work are discussed. Section 5.5 outlines the key messages and findings of this chapter

5.2 The Single Case Study Research Approach

This section briefly defines the case study research approach and the suitability of EMAGE as the single case study for this work.

A case study research is an empirical enquiry about a contemporary phenomenon i.e. a case set within its real world context, especially when the boundaries between phenomenon and context are not clearly evident (Yin 2009 p.18). Schell (1992) considers case study as the most flexible of all research designs, allowing a study to retain the holistic characteristics of real-life events while investigating empirical events.

On the other hand, the case study research approach has been criticised for being of less value, impossible to generalise from, being biased by researchers, and so on (Runeson and Höst 2009). However, Yin (2009 p.38-39, p.136-141), Runeson and Höst (2009), Zainal (2007), Kelliher (2011), and Flyvbjerg (2007) explain that the application of proper research methodological practices can enable the validation and generalisation of the results from a case study. Yin (2009 p.38-39, p.136-141) identifies analytic generalisation as a useful means of generalising case study findings. Zainal (2007) and Kelliher (2011) identify triangulation as a means of confirming the

validity of the single case study process. Flyvbjerg (2007) clarified most of the misunderstandings associated with case-study research approach as untrue. The case study is used in this work as a means to understand how IID can be dealt with, within a semantic technological setting.

A case study can involve a single or multiple cases. In this work, the EMAGE is used as a single case study. The use of a single case study in this work is justified by the fact that EMAGE provides the needed data for the critical test of the various novel FCA approaches presented in this work. Yin (2009 p. 52) explains that the single case is eminently justifiable under certain conditions in which the case represents (a) a critical test of existing theory, (b) a rare or unique circumstance, or (c) a representative or typical case, where the case serves a (d) revelatory or (e) longitudinal purpose. Flyvbjerg (2007) describes a critical case as having strategic importance in relation to the general problem. He recommends that when a researcher is looking for critical cases, it is a good idea to look for either the 'most likely' or 'least likely' cases, i.e. cases likely to either clearly confirm or irrefutably contradicts propositions and hypotheses. An integrated database is an example of a critical case in which IID are most likely to exist. Works such as (Fürber and Hepp 2010; Kimball and Caserta 2004; Rahm and Do 2000; McLeod and Burger, 2011; McLeod and Burger 2007; Bleiholder and Naumann 2008) assess the integrated database as susceptible to IID. EMAGE is an integrated database (McLeod and Burger 2011). The data set in EMAGE is integrated from unstructured and heterogeneous sources such as journal publications that often precede an experiment being published in EMAGE (McLeod and Burger, 2011). The EMAGE also integrates data from the gene expression database (GXD), and laboratory reports among others. Dau (2013a) and Melo (2013) identified IID in EMAGE RDF database.

Moreover, the use of the EMAGE as a use case is a unique opportunity because the CUBIST project used EMAGE as a case study in the investigation and development of FCA methods in a platform that combines essential features of Semantic Technologies and Business Intelligence. This provides additional resources illustrating how FCA tools and techniques are used to deal with IID in RDF data. It also provides a platform to compare approaches developed in this work with other FCA approaches while using the same data set.

In summary, the EMAGE is a good representative of an ideal case. It provides a context for the investigation of IID and also provides the context for evaluating the existing and new IID approaches. In this work, it is used as a single case study to address the research question.

5.2.1 Case Study Research Methods

EMAGE is investigated in this work, as a means to answer the research question. This was achieved by investigating EMAGE documentary sources, EMAGE RDF data set and personal communications with EMAGE researchers. Yin (2009 p. 101-114) lists documentary information and interviews as some of the sources of evidence in a case study. Runeson and Höst (2009) note that a case study may contain, elements of other research methods such as archival analyses, literature search, and observation. The investigations of the EMAGE involved the use of different methods by which evidences about the investigated phenomena (IID) were collected. These methods include archival analyses, literature search, personal communications, and analysis of EMAGE RDF data set. They were used as follows:

- **Archival analyses:** EMAGE and CUBIST documentary archives were explored in this work. The EMAGE archive[16] includes publications which relates to the processing of EMAGE gene expressions while CUBIST archive[17] includes scientific publications on how to semantically process RDF data. These archives are scrutinised to identify how IID thrive in EMAGE database, the different ways by which FCA can be used to deal with IID in RDF data set and the associated effect of such approaches.
- **Literature search:** Apart from CUBIST archive, other FCA publications were consulted as to understand how FCA can be used to deal with IID in a data set. Also, database literatures such as relational and semantic databases were explored. The essence is to properly understand how IID can be dealt with, in these databases.
- **Personal Communications:** They were communications between the researcher and other professionals such as CUBIST researchers. These communications held intermittently during this case study. For example, electronic mails were used to ask questions as to clarify concepts that were not properly understood. CUBIST researchers were mostly consulted for such clarifications where necessary. This is because CUBIST partnered with Heriot-Watt University to create the RDF version of the EMAGE data. This collaboration empowered CUBIST researchers with enormous knowledge about EMAGE and the EMAGE RDF data set. Various consultations with CUBIST researchers were made by means of telephone enquiry,

[16] http://www.emouseatlas.org/emage/about/publications.html

[17] http://www.cubist-project.eu/index.php?id=442

Skype video calls, electronic mails and informal chats in the course of this work. Some extracts from the electronic mails shared in the course of this work are presented in appendix A.

- **Analysis of EMAGE RDF data set:** A copy of EMAGE RDF data set was gotten from CUBIST and stored in a triple store. This data set was queried and the corresponding result sets were analysed through the use of the new and existing FCA approaches. The results of these analyses are depicted in chapter 8. Section 5.3 explains some FCA research methods used in the analysis of EMAGE RDF data set.

These research methods provided ample evidence about how IID are dealt with in EMAGE and how IID in EMAGE can be more effectively dealt with. The results (data) obtained from these research methods were used in evaluating the FCA approaches as narrated in chapter 8.

5.3 Formal Concept Analysis (FCA) Research Approaches

To identify and visualise the IID in an EMAGE RDF data set, an EMAGE RDF data was stored in Owlim triple store. This stored data set consists of over a million triples. Different SPARQL queries (see Chapter 8) were used to retrieve objects associated with many-value attributes from the stored data. The essence of each of the query is to retrieve objects whose attributes are many values. Subsequently, the semi-automated and automated FcaBedrock approaches (as comprehensively explained in chapter 7) were used to identify and visualise the IID in each of the retrieved record set. This section explains how the data analysis and visualisation approaches in FCA were used to exclusively identify and visualise IID, as evident in the semi-automated and automated FcaBedrock approaches.

5.3.1 Data Analysis and Visualisation Techniques in FCA

In FCA, conceptual scaling is used to transform a record set which has many-valued context to a single-valued context. Certainly, not all objects in a many-valued context are consistent or have complete attribute values. Some objects are inconsistent or incomplete. If a context is restricted to single, scaled, many-valued attribute, M is the set of n attribute values and G is a set of objects that have as a property, the many-valued attribute from which M is scaled, then the standard binary relation in FCA:

$I \subseteq G \times M$ can be formed to show which object has which attribute value, giving rise to the standard formal context $K = (G, M, I)$.

Given mutually exclusive many valued attributes, the context is defined to be consistent if each of its objects has only a single attribute value. If an object has two or more attribute values, then the context is inconsistent.

If the set of possible attribute values $M = \{v_1, v_2, \ldots, v_n\}$, then let the set of objects with value v_1 be G_{v_1} and the set of objects with value v_2 be G_{v_2} and so on. A consistent context can be defined thus:

$$\forall v_i, v_j \in M \bullet v_i \neq v_j \Rightarrow G_{v_i} \cap G_{v_j} = \emptyset \quad (1)$$

where "$\bullet$" implies "then".

The context is defined to be complete if every object in *G* has a value:

$$G = \cup\{G_{v_1}, G_{v_2}, \ldots, G_{v_n}\} \quad (2)$$

Once such a formal context has been created, the concept lattice visualises consistency and completeness in a very clear manner. For example, if $M = \{v_1, v_2, v_3\}$ and $G = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9\}$ then examples of consistent and inconsistent lattices are shown in Figure 6(a-d). 6a illustrates consistent data, 6b illustrates inconsistent data where two objects have two values, 6c illustrates inconsistent data where one object has all three values and 6d illustrates incomplete data which are clearly labelled at the topmost node of the lattice. For easy identification of IID in a concept lattice, it should be noted that an extent is identified by a lower filled semicircle labelled below the node while the associated intent is identified by an upper filled semicircle in the same node or in an ascending path to the node. Extents which have attributes with more than one value are inconsistent and an extent which is not associated to an attribute value is incomplete.

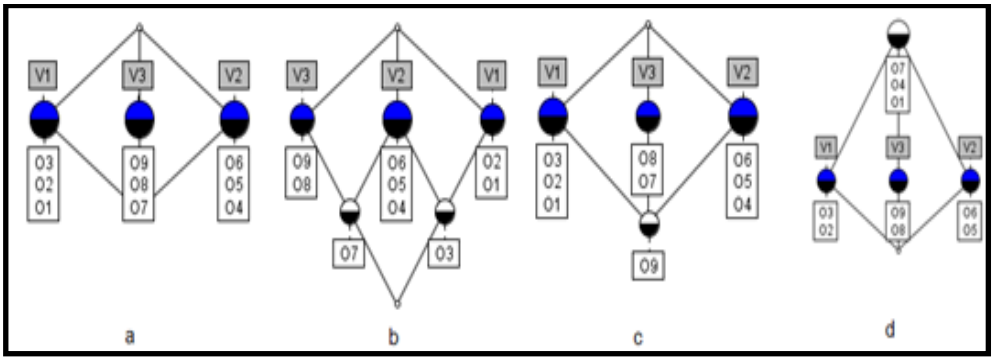

Figure 6: Examples of consistent (a), inconsistent (b and c), and incomplete (d) concept lattices

Also, objects which have mutually exclusive many-valued attributes can be separated out from a noisy context and exclusively visualised in a concept lattice through the use of the restrictive functionality in the FcaBedrock. The FcaBedrock application is an FCA tool that is used to transform a many valued record set to a context file. It was chosen in this research unlike its rivals such as ToscanaJ[18] because it allows its users to optionally select which attributes to convert to a context file. It is used in this research to restrict single value attributes when transforming a many-value record set to a formal context. Once the formal context of objects associated with mutually exclusive many-value attributes is created, the concept lattice can be built to exclusively visualise the IID in the formal context. For example, if *M* = {a-1, a-2, a-3, a-4} and *G* = {o1, o2, o3, o4} then let the set of objects with value $a-1$ be $o1_{a-1}$ and the set of objects with value $a-2$ be $o2_{a-2}$ and so on, then examples of inconsistent and incomplete lattices are shown in Figure 7(a-d). 7a and 7b illustrates inconsistent and incomplete data, 7c illustrates inconsistent data where three objects have contradictory values, and 7d illustrates incomplete data where attribute values without an associated object are displayed. Again, extents which have attributes with more than one value are inconsistent. An intent which is not associated to an object is incomplete.

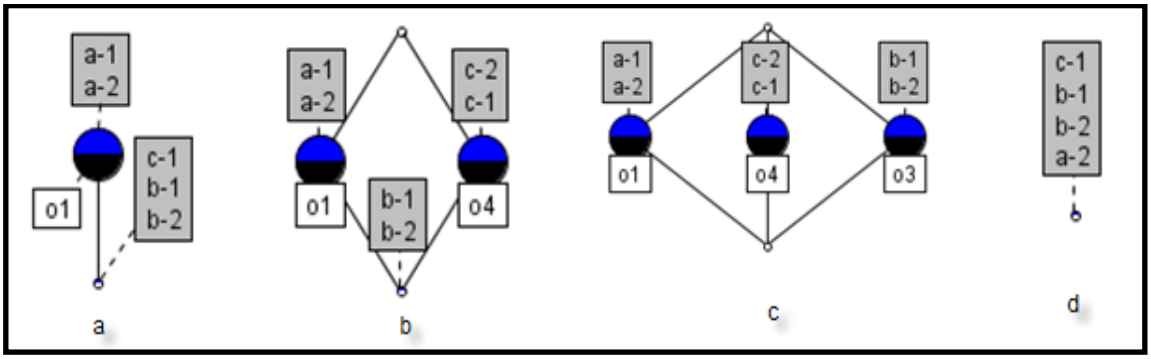

Figure 7: Examples of (a, b) inconsistent and incomplete, (c) inconsistent, and (d) incomplete concept lattices

The semi-automated and the automated FcaBedrock approaches are among the novel approaches implemented in this work. These approaches use FcaBedrock application to restrict the single-valued attributes as to exclusively visualise IID. The semi-automated FcaBedrock approach uses a manual system to restrict the single-valued attributes while the automated FcaBedrock approach uses an inconsistency mode to automatically restrict the single-valued attributes (see chapter 7).

[18] http://toscanaj.sourceforge.net/

The semi-automated FcaBedrock approach semi-automatically restricts single-valued attributes, when transforming a many-valued file to a formal context file. This is achieved by using the FcaBedrock application to manually restrict single-value attributes to a context file and subsequently building, visualising and editing concept lattice through the ConExp application. However, the approach can be painstaking, especially when there are many attributes to be restricted in the FcaBedrock. Hence the automated FcaBedrock approach was developed.

The automated FcaBedrock approach involves automatically transforming objects and their associated many-value attributes to a context file. This approach is implemented by extending the FcaBedrock application. This extension enables the FcaBedrock to automatically transform only objects associated with many valued attribute to a context file. The agile software development method was adopted to extend the FcaBedrock application. This is made possible through the collaboration with the FcaBedrock developer's team through an agile software development approach.

FCA provides data analysis and visualisation techniques which can be used to analyse and exclusively visualise IID in an RDF data set. Through the application of data analysis and visualisation techniques such as fault tolerance, interactive exploration, and conceptual clustering, among others, the data analyst can discover hidden knowledge in a massive data set. Data analysis and visualisation as used in this work involve the use of mixed design approach. Onwuegbuzie and Leech (2006) noted that *"Conducting mixed methods research involves collecting, analyzing, and interpreting quantitative and qualitative data in a single study or in a series of studies that investigate the same underlying phenomenon."* Both the quantitative and the qualitative data were analysed in this study. For example, some of the record sets retrieved by querying the stored EMAGE data set contain qualitative and quantitative data. Numeric (quantitative) data were retrieved from EMAGE RDF data set during the evaluation of the amount of IID present in a record set as evident in chapter 8. Also, most of the data collected by interview or archival analysis are descriptive qualitative data. Both qualitative and quantitative data can be visualised in a concept lattice.

5.4 Alternative Approaches, Challenges and Ethical Considerations

5.4.1 Alternative Approaches

A quantitative or qualitative research design could have been used in this work. But quantitative research is criticised as not being comprehensive because it excludes the meanings which participants give to events (Becker 1996; Sandelowski 2000). Qualitative research on the other hand, is criticised for its subjectivity and lack of generalizability (Sandelowski 2000). However, a mixed research design as implemented in this work, use quantitative research to reinforce the results from the qualitative research.

Also, a simulated data set could have been used instead of case study research data. But a simulated data set will have the short coming of not providing relevant documentations that explains the sources or causes of any identified IID.

5.4.2 Challenges

A challenge faced in this work is the inability to get licences for some of the applications which were evaluated. Dau and Melo approaches (see chapter 7 and 8) were not experimentally investigated in this work because of the inability to get their application licences. Nevertheless, the approaches were assessed by scrutinising their functionalities and results in their associated articles.

5.4.3 Ethical Considerations

This work obtained the consents to include personal communications of the CUBIST researchers whose electronic mails are attached in appendix A. The consents are included in the appendix A. The consent to include personal communications of the FcaBedRock developer whose electronic mails are attached in appendix B is also included in appendix B. Appendix C is the licence for the use of Owlim-SE application. This research was done in accordance with the research ethics of Sheffield Hallam University (SHU) and it was approved by the ethics committee of the university.

5.5 Key Messages and Findings

This chapter describes the various research methods used in this work. EMAGE is the single case study investigated for IID in this work through the use of single case study methodology. It is explained that case study methods such as data analysis techniques, and literature search were used in investigating the EMAGE. The automated FcaBedrock and the semi-automated FcaBedrock approaches are among the data analysis techniques used in investigating IID in EMAGE RDF data set. The research ethics and challenges encountered during this study are also outlined in this chapter.

Chapter 6: The Edinburgh Mouse Atlas Gene Expression Database (EMAGE)

6.1 Introduction

The Edinburgh Mouse Atlas Gene Expression Database (EMAGE) is a database of gene expression data in the developing mouse embryo and an accompanying suite of tools to search and analyse the data[19]. The EMAGE is a free online database. It provides an avenue for biologists to identify genes and their level of expressions in the tissues of the different Theiler Stages of the mouse. This knowledge is essential as to determine the possible causes of ill health in an organism. A knowledge of the level of gene expressions in tissues or organs of an organism at a particular developmental stage can help a biologist to determine the cause of ill health (mal-function of the cell, tissue or organ) in the organism at that particular stage (McLeod and Burger 2011). This can be done by comparing the genes expressed in the cells or tissues of a healthy organism with the genes expressed in the cells or tissues of an unhealthy organism at the same developmental stage.

The developmental stages of the house mouse as classified in Theiler (1989) are used in EMAGE as a means of storing gene expression data. Theiler (1989) classified the house mouse developmental stages into 28 different Theiler Stages (TS) namely TS*01 to* TS*28*. TS01 to TS26 identify the unborn or developing mouse while TS27 and TS28 identify the new born and the postnatal adult mouse respectively. These developmental stages are listed in appendix D. EMAGE stores data from TS01 to TS26 for the developing house mouse.

The EMAGE database uses resources from the e-Mouse Atlas Project[20] (EMAP). It also integrates data from various biological experimental reports such as journal publications, screening projects, and laboratory reports. Consequently, IID is likely to be evident in EMAGE database. Also, EMAGE seeks to identify the gene expressed in every tissue of every developmental stage of the mouse. But some methods used in EMAGE database such as its annotation and propagation methods can introduce inconsistency and incompleteness in the gene expression database.

[19] http://www.eMouseatlas.org/emage/

[20] http://www.emouseatlas.org/emap/about/what_is_emap.html#emap

Section 6.2 of this chapter, describes the methods used by the EMAP in EMAGE. Section 6.3 outlines and explains the causes of IID in EMAGE database. Section 6.4 and 6.5 describe the EMAGE search options and the EMAGE RDF data set respectively. The chapter is concluded by outlining its key messages and findings in section 6.6

6.2 The EMAP

The EMAP anatomy ontology and anatomy structures are used in EMAGE to enable the visualisation and interpretation of the genes expressed in the various tissues of the different Theiler Stages in the EMAGE database. This section explains how this is achieved in EMAGE database.

6.2.1 Visualisation of Gene Expression

EMAP contains a hierarchically organised ontology of anatomical terms for each Theiler Stage in mouse development and a set of 2D and 3D virtual mouse embryo models for post implantation stages of Theiler Stage of development. Figure 8 (see below) shows a subset of the EMAP Anatomy Ontology of Theiler Stage 11 while a mapping of gene expression on an EMAP virtual mouse embryo at Theiler Stage 17 is shown in Figure 9 (see below).

In Figure 8 shows a subset of EMAP anatomy ontology in a tree like structure. This arrangement enables an easy visualisation of the tissues and also the interpretation of genes expressed in the tissues. The *endoderm* for example has been ‘opened’ in Figure 8. This is indicated by the ‘-‘ symbol which reveals its two children (*primitive endoderm* and *definitive endoderm*). It is therefore easy to visualise that the *primitive endoderm is part of* the *endoderm* and that the *definitive endoderm* is part of the endoderm. This interpretation can also be applied to any open branch of the tree. The *extraembryonic component* has been ‘closed’ in Figure 8. This is indicated by the ‘+’ symbol. This indicates that *extraembryonic component* has children that can be viewed should that branch be ‘opened’.

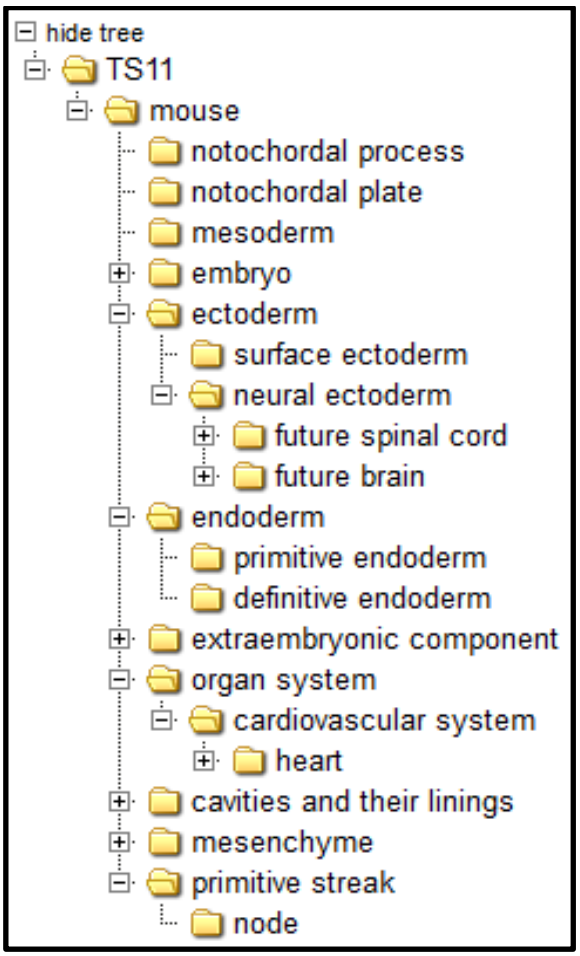

Figure 8: A part of the EMAP Anatomy Ontology of Theiler Stage 11 available in http://www.eMouseatlas.org/emap/ema/DAOAnatomyJSP/anatomy.html?stage=TS11 last accessed on 24th March, 2015

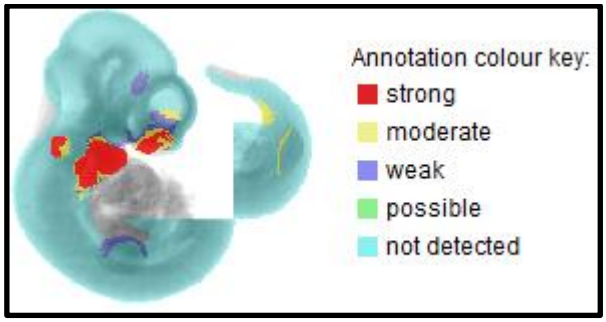

Figure 9: A whole-mount mapping of spatially annotated Mouse embryo showing the expression of distal-less homeobox at TS17 available at http://www.eMouseatlas.org/gxdb/dbImage/segment1/1444/detail_1444.html last accessed on 24th March, 2015

In EMAGE, the whole-mount view or 3D embryo model enables spatially annotated gene expressions to be visualised. For example, Figure 9 (see above) shows a whole-mount view of *distal-less homebox* expressions in a virtual *embryo* at Theiler Stage 17. The essence of EMAP anatomical ontology or virtual mouse embryo is to enable the visualisation and interpretation of the EMAGE stored genes expressions. This is made possible either through textual annotation or spatial annotation as described in section 6.2.2 below.

6.2.2 Annotation of Gene Expression

EMAGE gene expressions are gotten from experimental results of journal publications, screening projects, and laboratory reports among others. Such gene expressions are annotated with terms (standardised ontology) from Gene Ontology[21]. Gene expressions which are stored in EMAGE can be gotten from images that are either an entire mouse embryo (whole-mount), or a slice/section of the mouse. They can also include textual descriptions of the gene expressions. The images of whole-mount or slice of a mouse show the gene expression patterns and the associated tissues. For example, Boyl et al., (2001) outline the expression patterns and descriptions of *Otx2* in *Forebrain* and *midbrain* of the developing mouse. His experimental report (Journal publication) contains details of his experiment and associated diagram that was analysed. Essential details relating to gene expressions for instance, the expression patterns of *Otx2* in the various investigated tissues, the materials used such as information of the specimen, and the procedures, can therefore be annotated and stored in EMAGE database.

Spatial or textual annotation techniques can be used to retrieve gene expression data from different experimental sources and stored in EMAGE. In spatial annotation, gene expressions are mapped to their associated region of space and developmental stage (Theiler stage) in the virtual embryos. This is achieved in EMAGE by the use of the 2D or 3D EMAP embryonic models **(**Richardson et al., 2014). As a result, EMAGE gene expressions can be stored in 2D or 3D imagery that is also available for query. Figure 10 (see above), is an illustration of a spatially annotated mouse embryo which shows the expression of the gene '*distal-less homeobox 5*' at TS17.

[21] http://geneontology.org/

Spatial annotation is performed independently of any text annotation that may accompany it in an EMAGE entry[22]. Basically, EMAGE spatial annotation can involve any of the following:

- Spatial annotation of sectioned and whole-mount stained embryos: This involves the mapping of gene expressions in sectioned or whole-mount stained embryos on EMAP 2D virtual embryo. The displayed gene expressions are usually retrieved from photographs of whole-mount or sectioned stained embryo. The investigated image is first read by an EMAGE application such as MAPaint, then expression patterns in the image are warped onto a whole-mount or sectioned image of one of the standard EMAP model embryos.
- Spatial annotation of 3D data: 3D gene expressions are retrieved from EMAGE data sources through technologies such as high resolution optical Microscopy, magnetic resonance microscopy (MRM) or Optical Projection Tomography (OPT). EMAGE curators use OPT to capture 3D images of whole-mount embryos assayed through colourimetric *in situ* protocols. AMIRA, WlzWarp or MAPaint programs (EMAGE in-house image warping applications) are subsequently used to spatially map the captured OPT data onto stage matched models (Richardson et al., 2014).

Textual annotation is performed manually by the EMAGE annotator. Textual descriptions in biological reports are annotated and mapped to their associated EMAP anatomical terms. Textual annotation involves the manual mapping of genes expressed in EMAGE experimental report to their associated EMAP anatomy ontology. Figure 10 (see below), illustrates the linking of contents of an experimental report to corresponding EMAP anatomy ontologies. The textual descriptions written at the right side of Figure 10 represents textual explanations about the gene expression of the investigated tissue. The picture of the mouse in the right corner of Figure 10 represents the whole-mount view of the mouse.

[22] http://www.eMouseatlas.org/emage/

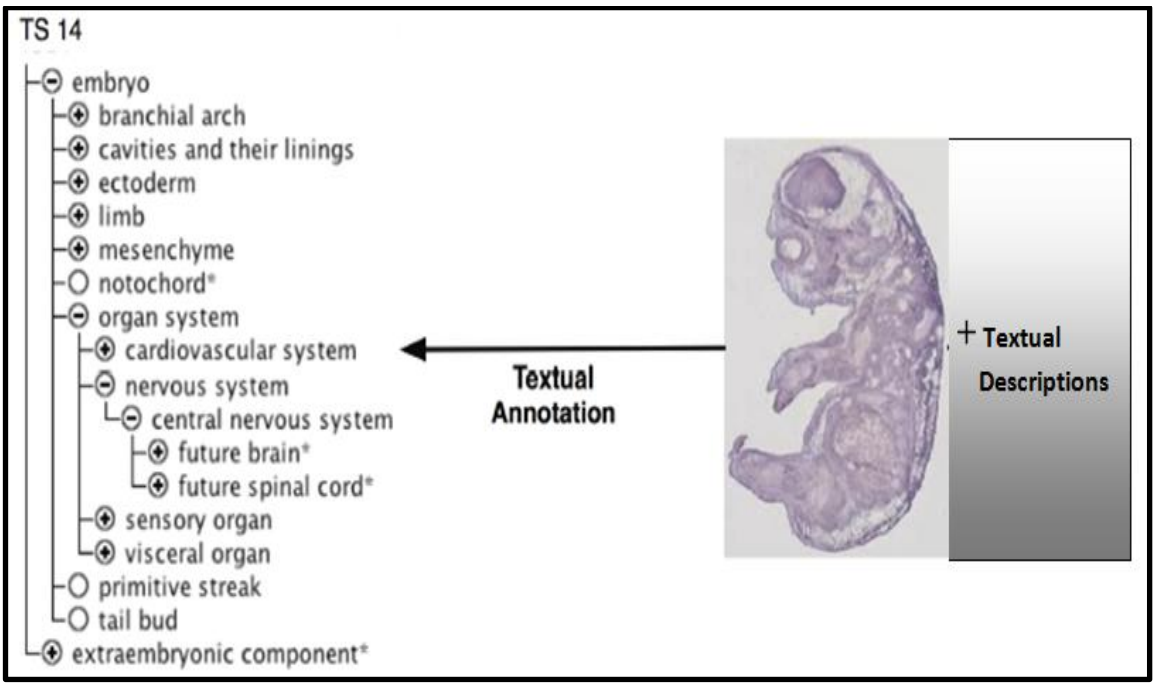

Figure 10: A modification of Figure 5 in (McLeod and Burger 2011), illustrating the textual annotation process

The processes and technologies involved in EMAGE spatial and textual annotation are articulated in EMAGE publications[23] such as in (Christiansen et al., 2006**;** Richardson et al. 2014; Venkataraman et al. 2008).

6.2.3 Propagation of Gene expression

EMAGE database propagates its gene expressions as to ensure the identification of the gene expressions in every gene in every tissue of every Theiler Stage. The propagation of gene expression in EMAGE can be understood through the EMAP anatomy ontology. A tissue is related to another tissue and is described in EMAP anatomy ontology with the *is_part_of* relationship. For example, it can be seen from Figure 9 above, that the *heart is_part_of* the *cardiovascular system*, the *cardiovascular system is_part_of* the *organ system and* the *organ system is_part_of* the *mouse*. The *is_part_of* relationship between tissues as depicted in the EMAP hierarchical tree structure is used in the propagation of gene expressions in EMAGE.

In EMAGE, a gene expression can either be propagated up (positively) or down (negatively) depending on its expression. In positive propagation, *detected* gene expression levels associated with a lower granularity tissue (child tissue) is assigned to

[23] http://www.emouseatlas.org/emap/about/publications.html

related higher granularity tissues. *Cardiovascular system* is a lower granularity tissue to *organ system* as shown in Figure 9. If *Otx2* is identified as *strong* in *cardiovascular system*, then a positive propagation will assign *strong* to *Otx2* in *organ system*.

In negative propagation, a gene which is not detected in a higher granularity tissue is assigned as 'not detected' to related lower granularity tissues. *Organ system* is a higher granularity tissue to the *cardiovascular system*. It also has an *is_part_of* relationship with *cardiovascular system*. If *Otx2* is *not detected* in the *organ system* then a negative propagation will assign '*not detected'* to *Otx2* in the *cardiovascular system*.

In theory, propagation of gene expression provides an ideal way to complete the tissues without an associated gene expression in EMAGE. However, the existence of IID prior to propagation of gene expressions in the database, the annotation process, and the data integration issues are among the factors that can cause IID when the gene expressions are propagated. These are explained in section 6.3 below.

6.3 Causes of IID in EMAGE Gene Expression Database

This section explains the causes of IID in EMAGE database and how the EMAGE search interface can be used to identify and visualise the IIDs existing in EMAGE. The following factors are identified in this work as the cause of IID in the EMAGE database.

6.3.1 Data integration

EMAGE data includes annotated gene expressions. These gene expressions are from different biologist experimental reports. EMAGE curators retrieve details relating to gene expressions in these experimental reports and store same in the EMAGE database. But the lack of standardised ontology for representing gene expressions in the experimental reports which were presented by biologists can affect the consistency of the data in EMAGE. For example, a gene expression can be described as *weak* in a biologist's experiment while the same degree of gene expression may be described as *moderate* in another biologist's experiment.

Also, the non-standardised image technology used in capturing the experimental reports can affect the consistency and completeness of the data in EMAGE. For example, the non-standardisation of technology for photographing biological images will result in IID in EMAGE data. More so, different precision microscopes may be used by different biologists to capture the analysed image. The retrieved data from the

various reports will be incomplete or inconsistent when they are integrated into EMAGE database.

Other sources of IID that results from data integration in EMAGE database can include experimental error or variation in experimental conditions which can lead to differences in results produced by different biologists on the same test as identified in Burger (2007). It is also possible that there is no information about some genes in some tissues from a particular stage of the developing mouse.

6.3.2 Propagation

If there are IID in EMAGE database which may have resulted from the EMAGE data integration processes (as explained in section 6.3.1 above), then the propagation of gene expressions in EMAGE will cause additional IID in the database.

Ideally, propagating gene expressions in EMAGE database should ensure a consistent and complete gene expression database. Nevertheless, some inconsistent or incomplete gene expression data in the EMAGE database will contradict other EMAGE data when the expressions are propagated. For instance, let us assume that the gene *smad1* is identified at a *strong* expression level in the *cardiovascular system* in Theiler Stage 11 (see Figure 8 above). The *organ system* and the *mouse* will then be assigned *strong* expression level when the EMAGE dataset is positively propagated. However, assigning *strong* expression level to *smad1* in the mouse will result in an inconsistent expression in EMAGE where *smad1* in the *cardiovascular system* or in the *mouse* is associated with *moderate* expression level. Also, let us assume a situation where in Theiler Stage 11, the gene *otx2* is *not detected* in the *neural ectoderm*. This will imply that *otx2* in *future spinal cord* and in *future brain* will *not be detected* because *future spinal cord* and *future brain* are part of *neural ectoderm*. But assigning *not detected* to *otx2* in *future brain* will result in inconsistent expression if another experiment has already detected *otx2* in *future brain* in the same Theiler Stage.

6.3.3 Textual Annotation

Textual descriptions of experimental reports may produce incomplete gene expression information in the EMAGE. This is because, the EMAGE annotator will only capture whatever information the researcher/biologist wishes to present in his experimental report. As explained in (Taylor et al., 2013), textual annotations may be incomplete if for instance, the researcher is only interested in the *heart* then he will not create textual

annotations for the *brain*. If for instance the research is documented at a high granularity, the textual annotation will report the gene as being expressed in the *heart* rather than the sub-component in which it is actually found. Also, textual annotation is manually processed by the EMAGE annotators and this may result in errors of wrong tabulation or slight omissions.

6.3.4 Data Processing Technologies

Spatial annotation in EMAGE enables the retrieval of gene expressions from images in biological reports as explained in section 6.2.2. IID can exist in EMAGE as a result of spatial data processing technologies. This is because there are different technologies for sourcing spatial data in EMAGE. Software applications such as AMIRA, WIzViewer or MAPaint are different programs for producing spatial data in EMAGE. These programs are not automated application and their spatially mapped data can vary depending on the proficiency of the program user. Such variations can introduce IID in an EMAGE database.

A summary of the various causes and examples of IID in EMAGE database is outlined in Table 7 (see below).

Table 7: Causes and examples of IID in EMAGE

S/No	Cause of IID	Reason for IID	Example
1.	Data integration	• Lack of standardised ontology • Non-standard image capturing technology • Experimental errors, and different precision microscopes • Results from multiple experiments	Different precision microscopes may be used by different biologists to capture their analysed images, thereby resulting in IID when this data is integrated in EMAGE database
2.	EMAGE propagation	• Propagating existing IID	Negatively propagating not_*detected* to *otx2* in *future brain* where another

			experiment has already detected *otx2* as *strong* in *future brain*.
3.	Textual annotation	• Level of granularity, wrong tabulation, and slight omission	The annotation of *Organ system* without annotating *Cardiovascular system*
4.	Data processing technologies	• Level of proficiency of EMAGE program user	When different people use EMAGE application such as AMIRA or WlzViewer for the same data set, there can be variations in their produced results.

6.4 The EMAGE Search Options

EMAGE users can access non-spatially mapped data via an EMAGE repository which is accessible through the EMAGE website (Stevenson et al. 2011; Venkataraman et al. 2008). EMAGE provides its users with search options (interfaces) through its website, for the search and analysis of its data. These search options in EMAGE website uses key words such as the gene/protein name, symbol or ID as a query term, embryonic region as query term, Biomart interface, and the name of an anatomical structure, among others. Basically, EMAGE captures a query when its user searches for a term through any of the search interface. It processes this query and displays the result set in a tabular structure. Each EMAGE entry is associated with the details of the search result as shown in Figure 11.

The EMAGE search options can be used to identify and visualise IID as evident in Figure 11. The result from searching for a key word in EMAGE website can include different rows with differences in expression levels for the same gene in the same tissue of the same Theiler Stage. IID can be identified in such results when scrolling down the web page(s) displayed by the search engine. The colour on the images displayed on the web page as evident in Figure 11, provide some indication of the level of expression. Two or more different expression levels (different colours) from the same gene, of related tissues or the same tissue in the same Theiler Stage will indicate

inconsistent expression. This will imply that the corresponding tissues have inconsistent gene expression. But such inconsistency is identified by comparing different rows in the results displayed by the EMAGE search engine. This process of identifying inconsistent gene expression is tedious. A better method for visualising IID will enable an easier means of identifying the IID in the results displayed by the search engine.

The EMAGE search options do not separate out the IID from its result set as to exclusively visualise the IIDs. Talyor et al. (2013) explain that "*EMAGE does not provide any visualisation to summarise the expression information across time or between multiple genes*." Consequently, the binary or analogue inconsistency in a tissue at particular TS cannot be exclusively visualised in the EMAGE website. There is a need to improve the EMAGE search options as to incorporate techniques that can exclusively analyse the IID in the EMAGE database.

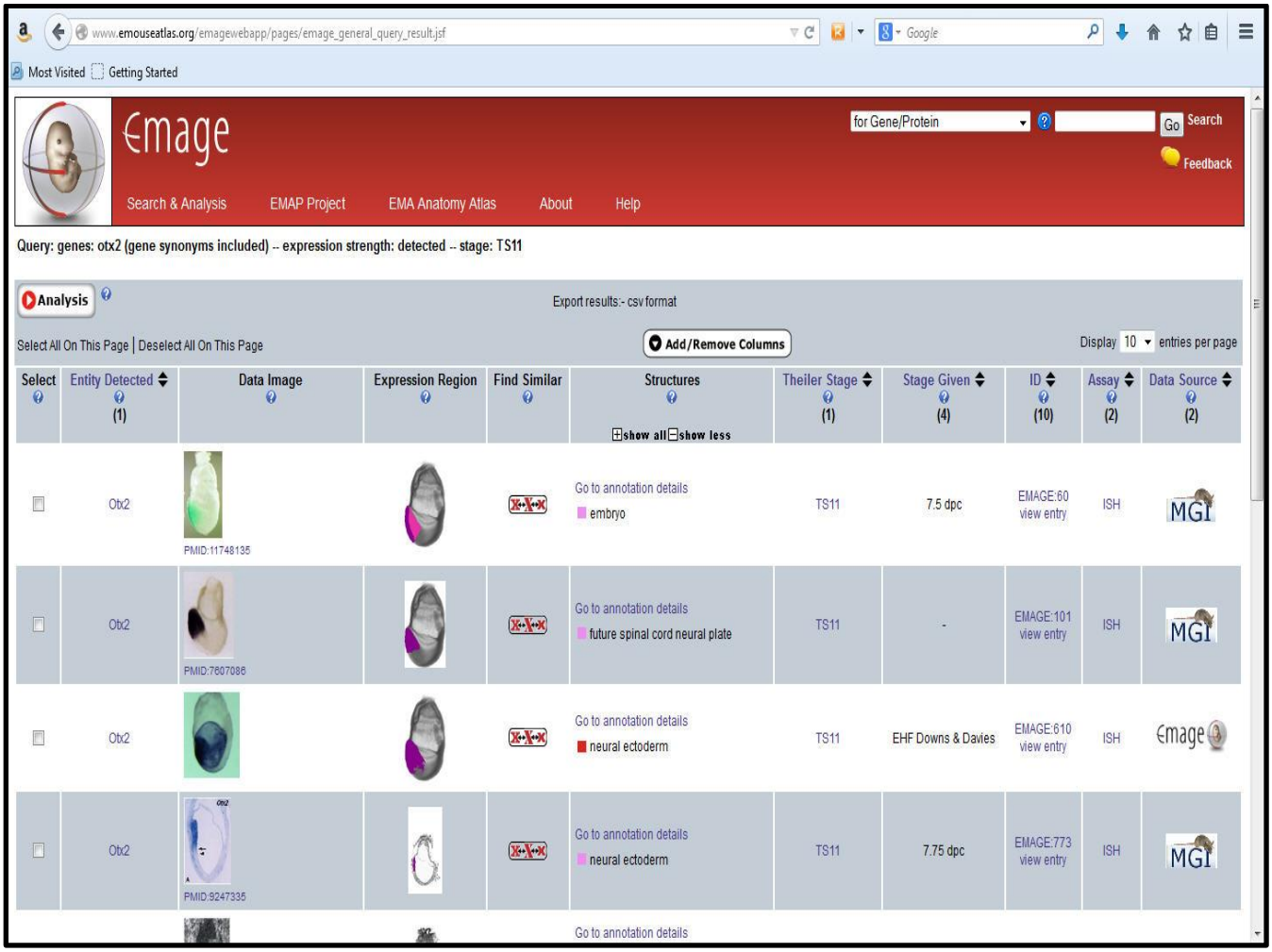

Figure 11: Result of asking where the gene Otx2 is detected in TS11 through the gene/protein search option of EMAGE website, available at http://www.emouseatlas.org/emagewebapp/pages/emage_general_query_result.jsf, last accessed on the 23rd March 2015.

6.5 EMAGE RDF Data Set

The EMAGE database stores the EMAGE data set. An EMAGE user can request for the EMAGE data set or a subset of the EMAGE in different formats such as relational or XML format through the EMAGE website.

The EMAGE RDF dataset is explored in this work. A sub data set of the EMAGE data in RDF form was provided by CUBIST project team and analysed for IID in this work. The ontology for the EMAGE RDF data set is presented in Dau (2013a) and reconstructed in Figure 12 (see below).

As described in Figure 12, the EMAGE RDF data set contains different classes which include Theiler Stage, Tissue, Gene, Strength, and Experiment. These classes have properties and derived properties. The relationships among the classes, properties and derived properties in the EMAGE RDF data are depicted in Figure 12 below. For example, the Gene has *g_intextual_annotation* as its derived property. Also, a textual annotation has *has_involved_gene* and *in_tissue* as its properties. A textual annotation can be associated with different expression levels – *strong*, *moderate*, *weak*, or *not_detected*. Also, detected is the derived property for *strong*, *moderate*, or *weak* expressions.

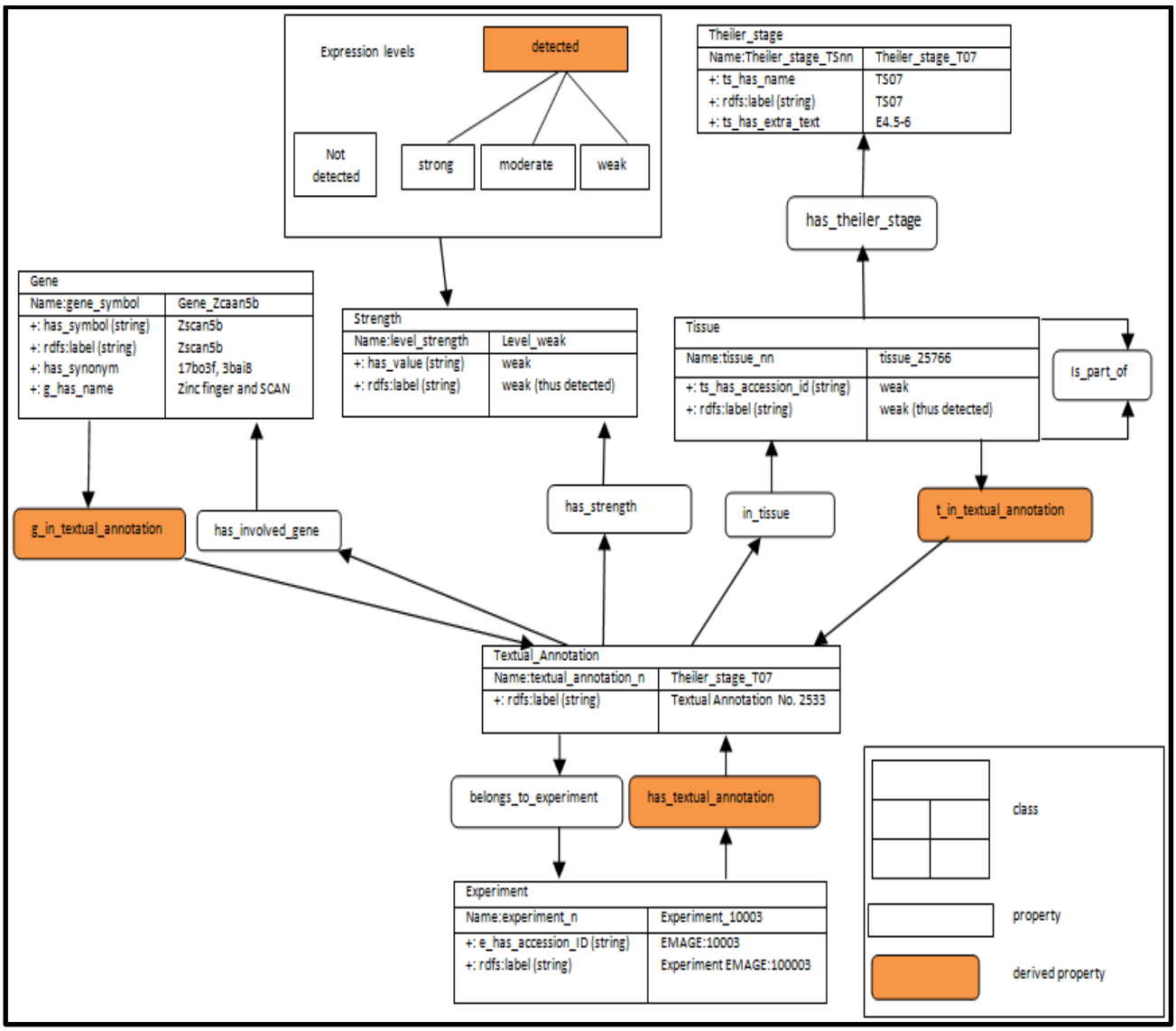

Figure 12: The ontology for the EMAGE RDF data set as adapted from Dau (2013a)

6.6 Key Messages and Findings

This chapter has explained the EMAGE, the EMAP, and the causes of IID in EMAGE. It explained the various ways by which the EMAGE uses EMAP anatomy ontology and anatomy structures. It also described the EMAGE RDF data set. This chapter identified that the search options used in EMAGE are unable to exclusively visualise IID existing in the database. Consequently, the FCA approaches for dealing with IID in RDF data set are explored in chapter 7 below.

Chapter 7: FCA Approaches for dealing with IID in RDF Data

7.1 Introduction

The IID in a data set can be dealt with through the use of FCA approaches such as Dau's approach, CUBIST approaches, fault tolerance approach, attribute exploration, the semi-automated FcaBedrock approach or the automated FcaBedrock approach. This chapter provides comprehensive details on how these approaches are used in dealing with the IID in an RDF data set.

The FCA approaches for dealing with IID in a data set is divided into two in this Chapter namely: the existing FCA approaches and the new FCA approaches. The existing FCA approaches contain approaches which were developed by other FCA researchers. It includes attribute exploration, fault tolerance, Dau and CUBIST approaches. The new FCA approaches are approaches which were developed in the course of this work. It includes the association rule, semi-automated and the automated FcaBedrock approaches.

RDF data set tolerates IID as explained in chapters 1 and 3 of this work. Such IIDs can exist in record sets retrieved from the RDF database. In chapter 3, it is noted that there is need for robust approaches which can be used to identify and visualise the IID existing in RDF database. This will enable the data users to make informed decisions or conclusions as explained in chapter 1.

This chapter begins by briefly explaining in section 7.2, the basic SPARQL keywords that can be used in retrieving IID from a noisy data set. Section 7.3 discusses existing FCA approaches for dealing with IID. Section 7.4 discusses the new FCA approaches proposed in the course of this work.

7.2 Retrieving IID with SPARQL

SPARQL is a query language which is used in retrieving matching subgraphs from an RDF triple store. RDF subgraph which have contradictory attribute values (inconsistent data) or which do not have some required attribute-values (incomplete data) can be retrieved through SPARQL queries.

A basic graph pattern matches a subgraph of the RDF data when RDF terms from that subgraph can be substituted for the variables and the result is an RDF graph equivalent to the subgraph (Harris and Seaborne 2013). Some examples of the RDF

graphs are presented in Chapter 3 of this work. The *optional* keyword can be used in SPARQL to check for bindings to contradictory attribute values in the data set. Quilitz and Leser (2008) explain that in an optional match, either the optional graph pattern matches a graph, thereby defining and adding bindings to one or more solutions or it leaves a solution unchanged without adding any additional bindings. Similarly, the *union* keyword can be used to combine graph patterns so that one of several alternative graph patterns may match (Quilitz and Leser 2008), thereby providing a means of retrieving inconsistent data from contradictory graph patterns. Consequently, the use of *optional or union* key words in a SPARQL query can enable a match of subgraphs containing IID in the RDF data store. The *optional* and *union* keywords are the two basic SPARQL keywords for retrieving IID. Dau (2013a) explains how the optional and the union keywords are used in retrieving IID from an RDF data set in a triple store.

SPARQL also has other keywords which are relevant to retrieving inconsistent or incomplete subgraphs from RDF graphs. These keywords include *filter*, *bind*, *sameTerm*, and *GroupBy*. For more details about writing SPARQL query, this work recommends its readers to consult publications such as Quilitz and Leser (2008), Power (2003), DuCharme (2011), Dau (2013a) and Nwagwu (2013). These documentations provide good examples of how IID can be retrieved from an RDF data set in a triple store.

7.3 Existing FCA approaches for dealing with IID in RDF data set

This work identifies the Dau, CUBIST, fault tolerance and attribute exploration approaches as existing techniques for dealing with IID. These approaches are discussed in this section.

7.3.1 Dau's Approach-SPARQL2context creator

Dau developed the SPARQL2context creator as documented in (Dau 2013a) where it is explained how IID can be retrieved from a triple store and visualised in a lattice structure. The SPARQL2context creator is an application that can retrieve and transform the result of the SPARQL-query into a formal context. It functions by using the names of the query output variables which begins with 'o' to generate objects of the context. Variables which begin with 'a' are used to generate attributes of the formal context while other variables have no impact on the generated context. Also, more than

one variable that begins with an ‘o’ or an ‘a’ are simply concatenated to generate an object or attribute name respectively.

Dau’s approach involves retrieving a record set that contains IID, transforming all the data in the record set to a formal context, and visualising this single-valued context in a lattice structure. This is illustrated in Table 8 and Figure 13 below. Table 8 provides an example of a formal context created from a SPARQL-query result set by the SPARQL2Context creator, as reproduced from Dau (2013a). In the generated formal context (see Table 8), there are some rows and columns of cells which do not contain any data. Such rows or columns depict incomplete data. Examples of such empty row or column of cells are o1, o1-o2, o3, A1, A1-A2, and A2.

Table 8: Transforming SPAQRL-query-results to formal contexts as evident in Dau (2013a)

The result of a SPARQL-query

Obj1	Obj2	Att1	Att2
O1			
O1	O2		
	O3		
		A1	
		A1	A2
			A2
O4		A3	
O5	O6	A4	
O5	O7	A5	A6
O5	O7	A4	

The generated formal context

	A1	A1–A2	A2	A3	A4	A5–A6
O1						
O1–O2						
O3						
O4				X		
O5–O6					X	
O5–O7					X	X

Dau notes that SPARQL query can be written as object restricted, attribute restricted, objects unrestricted and attributes unrestricted queries. A query that is designed to retrieve a set of attributes and unrestricted objects from a database is likely to retrieve a record set containing some empty rows. Similarly, a query that is designed to retrieve a set of objects and unrestricted attributes is likely to retrieve a record set containing some empty columns. It can therefore be said that a query that is designed to retrieve a set of objects and unrestricted attributes, a set of attributes and unrestricted objects, or unrestricted objects and unrestricted attributes, is likely to contain IID in its result set.

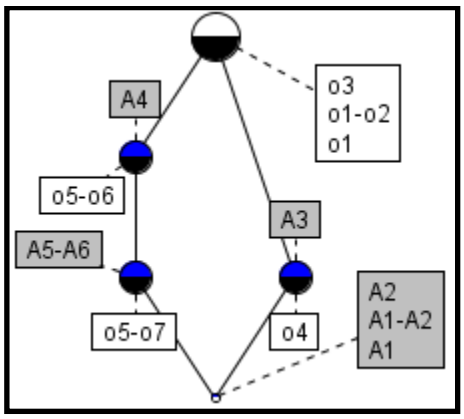

Figure 13: Concept lattice built from Table 8

Figure 13 is built from the formal context in Table 8. It is used to illustrate how IID can be identified, as suggested in Dau (2013a). Dau notes that instances of contradictory data can be apparent where an object is associated with at least two pairs of attributes. As a result, o5-o7 in Figure 13 should be given a closer look to check if the attributes A5-A6 and A4 are contradictory. Empty columns and empty rows depict incomplete attributes and incomplete objects respectively, as shown in Table 8 (see above). In Figure 13, o3, o1-o2, and o1 are incomplete objects while A2, A1-A2, and A1 are incomplete attributes. Chapter 8 demonstrates the application of Dau's approach on EMAGE data set.

7.3.2 CUBIST Approaches

A general approach of the CUBIST application is to query a triple store and convert the result into a formal context, which can be simplified to make it manageable, before visualising it as a concept lattice and associated charts (Melo et al., 2013). CUBIST uses a set of pre-defined queries to query the ontology of an RDF data set, which are then converted to a formal context. It deals with IID existing in its explored data set through utilizing distinct colour, interactive exploration and fault tolerance.

Data conflict is emphasised in CUBIST by associating distinct colours to the inconsistency type. For example, a red colour can be used to indicate that a gene is both detected and not detected (binary inconsistency) in the same tissue at the same Theiler Stage. CUBIST users can visualise the inconsistency associated to data through an interactive exploration of the data. Also, CUBIST applies fault tolerance as a means of inferring missing data (fault tolerance is discussed in section 7.3.3 below).

Generally, CUBIST combines approaches developed in works such as Andrews (2011) and Dau (2013a, 2013b), in order to deal with IID in an RDF data set. Its approaches for dealing with IID in EMAGE data set is further described in Chapter 8.

7.3.3 Fault tolerance

Fault tolerance is another approach of dealing with IID in RDF data set. It is noted in Dau (2013b) that if all the information in the formal context is preserved in a concept lattice (*G, M, I*), the concept lattice might exponentially grow in size such that it has $2^{\max\{|G|, |M|\}}$ many formal concepts; hence the introduction of a fault tolerance approach. Fault tolerance in FCA can be described as the substitution of a certain amount of missing data as true values during the computation of formal concepts (Andrews and McLeod 2013). Generally, the application of fault tolerance in FCA involves the introduction of softness (user defined tolerance) to the constraint implemented in FCA.

Fault tolerance approach as used in FCA, provides a means of reasoning with an incomplete and noisy data set. It can enable the visualisation and also enhance the readability of the concept lattice from a large data set. When fault tolerance is applied on a noisy data set, it produces interesting and manageable lattices which enables the inference of missing values as demonstrated in (Andrews and McLeod 2013; Dau 2013b).

Table 9 is an illustration of a formal context in binary format. The objects in Table 9 includes a, b, c, d, and e while the attributes of the context includes 1, 2, 3 and 4. An application of fault tolerance can involve allowing a certain amount of missing attributes. For example, a tolerance of one will involve changing the "0" value in attribute 2 to "1". Similarly, a fault tolerance of two will involve changing the two "0"s in attribute 1 and 3 to "1"s as explained in Andrews and McLeod (2013).

Table 9: An illustration of a formal context in binary format as adapted from Andrews and McLeod (2013)

	1	2	3	4
a	0	1	0	1
b	1	1	0	0
c	1	1	1	0
d	0	0	1	0
e	1	1	1	0

Also, in Dau (2013b), the incidence relation I in a formal context (*G*, *M*, *I*) is extended so that the derived lattice becomes smaller. He achieved this by measuring the incidence relations (associating them with numerical values) and applying a threshold as a fault tolerance. A total of 6 different ways of measuring the *I* between *g* and *m* is presented in (Dau 2013b) namely, global similarity:objects only, global similarity: attributes only, global similarity: objets and attributes, local similarity:objects only, local similarity:attributes only, and local similarity:objects and attributes. These measures are represented with *GObj, GAtt, GObj,Att , LObj, LAtt ,* and *LObj,Att* respectively. Table 10 and Figure 14 (see below) illustrate how the mathematical measure of *GObj* is implemented

Table 10 is an example of the global object-based measurement (*Gobj*) derived from a formal context. For a violation of *I* by exactly 9 objects (o1 to o10), an incidence measure of 0.1 is recorded. It should be noted that when the '*I'* is not associated with an 'X', it means that a violation has occurred. Also, for a violation by 8 objects, the incidence measure of 0.2 is recorded. In Figure 14, a threshold of 1 identifies all the information in the formal context. A reduction in the threshold, such as a threshold of 0.6, and 0.4, presents a concept lattice that tolerates that amount of missing information as apparent in the incidence measure. The proof of these derivations is presented in Dau (2013b).

Table 10: Example of a global object-based measurement (Gobj) incidence measure (right) from a formal context (left) as adapted from Dau (2013b)

obj	A1	A2	A3	A4
o1	X			
o2		X	X	
o3		X	X	X
o4				
o5				X
o6			X	X
o7				
o8				X
o9			X	
o10				X

Formal Context

o1	1.00	0.20	0.60	0.50
o2	0.10	1.00	1.00	0.50
o3	0.10	1.00	1.00	1.00
o4	0.10	0.20	0.60	0.50
o5	0.10	0.20	0.60	1.00
o6	0.10	0.20	1.00	1.00
o7	0.10	0.20	0.60	0.50
o8	0.10	0.20	0.60	1.00
o9	0.10	0.20	1.00	0.50
o10	0.10	0.20	0.60	1.00

Incidence measure: = Gobj

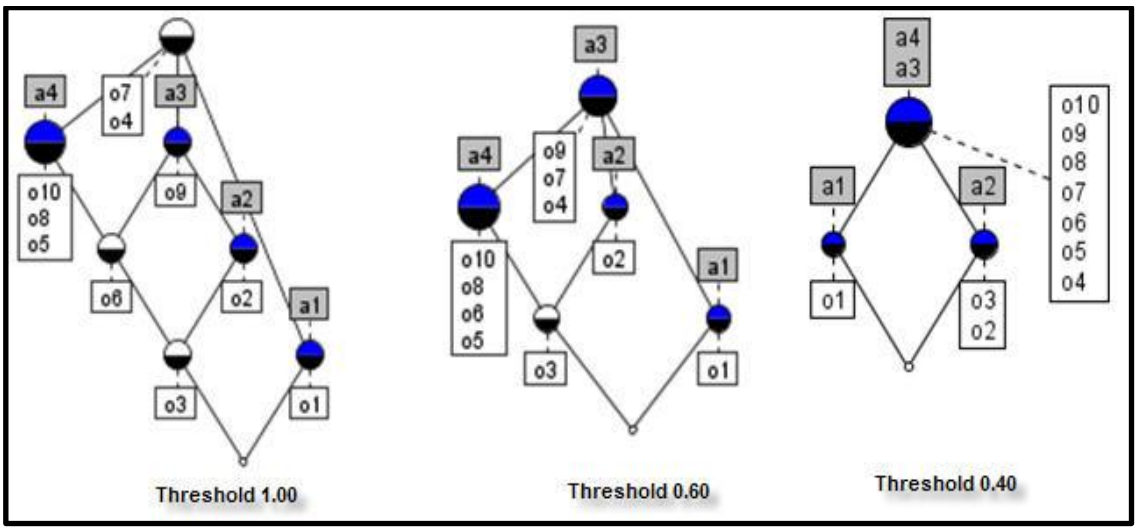

Figure 14: Examples of concept lattices derived from GObj as build from the formal context in Table 10

Figure 14 above, depicts examples of concept lattices derived from GObj as build from the formal context in Table 10. The fault tolerance approach can be used to infer missing data. For example, in the concept lattice with the threshold of 0.60 (see Figure 14), the attribute 'a3' is inferred for the objects o7 and o4. In the 0.40 threshold of Figure 14, a4 is inferred for the objects (o9, o7, o4) while a3 is inferred for the objects (o7, o4, o10, o8, o5, o4).

Fault tolerance is applied in noisy data sets to reduce the combinatorial explosion of the number of formal concepts and also to compute interesting result sets. Other works on fault tolerance and how it can be used to reason with noisy data set include Pensa and Boulicaut (2005a, 2005b). Having noted that fault tolerance as presented in (Dau 2013b; Andrews and McLeod 2013) provides a useful means of reasoning with a noisy and incomplete data set, there are some challenges associated with this approach in this work. The application developed in Dau (2013b) is not publicly available. It was not possible to obtain the implemented software in this work. Also, there are no publications that show how the 6 different incident measures are used in dealing with IID of an EMAGE data set. Consequently, evaluating this approach was not realisable. The fault tolerance approach is not evaluated further in subsequent chapters of this work.

7.3.4 Attribute Exploration

Attribute exploration is a knowledge acquisition method of FCA that is used to acquire complete knowledge about an application domain by asking successive questions to a domain expert (Sertkaya 2009). In so doing, missing attributes of objects are identified and added to the formal context. Attribute exploration enables the identification of incomplete data. It ensures that implications are computed for a given formal context (*G*, *M*, *I*). In a formal context, an implication between two subsets of attributes Q and R means that if a set of objects is described by the attributes contained in Q then it is necessarily described by the attributes contained in R. This is mathematically represented as follows:

A context (*G*, *M*, *I*) satisfies the implication $Q \rightarrow R$, with $Q, R \subset M$, if for all $g \in G$, gIq for all $q \in Q$ implies gIr for all $r \in R$ (Carpineto and Romano 2004 p. 141).

In FCA, attribute exploration enables the computation of implications in which objects are confirmed to have all attributes of the implications by a domain expert. The objectives of attribute exploration in FCA are to identify all objects that implication rules apply to and to provide counter examples where the rules are not applicable as identified by a domain expert. The classical method of attribute exploration is explained in (Ganter 1999; Ganter 2010 p.322- 340). It is based on implications and counter examples. OntoComP[24] and conExp are some of the open source applications that implement attribute exploration. Some of the applications of Attribute exploration include ontology completion Sertkaya (2009), security checks Obiedkov et al., (2009), and web data Jäschke and Rudolph (2013). Nonetheless, the use of attribute exploration is not appropriate in every domain.

In attribute exploration, knowledge is assumed to be completed by the domain expert. However, it has been noted that the semantics whereby knowledge is completed by a domain expert as adopted by attribute exploration does not agree with the semantics of the Open World Assumption (OWA) (Baader et al., 2007). OWA follows open world semantics which implicitly assume that a knowledge base may always be incomplete (Hitzler et al., 2011 p.131). The OWA is discussed in Chapter 3 of this work. The open and always incomplete semantic web, RDF(S) and OWL adhere to the OWA as discussed in Chapter 3 (see also Hitzler et al., 2011 p.372). As a result, attribute exploration is noted in this work as inappropriate for RDF and OWA knowledge bases.

[24] http://ontocomp.googlecode.com

7.4 New FCA approaches used to deal with IID in RDF data

This work also identifies new FCA approaches for identifying and visualising the IID existing in an RDF data set. These approaches are proposed in the course of this work and explained in this section. They include the association rule, the semi-automated and the automated FcaBedrock approaches.

7.4.1 Association Rule

The application of the association rule in FCA can provide a means of visualising missing or incomplete data when two concept lattices are compared. Stumme et al., (2002) explain that an association rule is a pair $X \rightarrow Y$ with $X, Y \subseteq M$. For $X, Y \subseteq M$, the implication $X \Rightarrow Y$ holds in the context, if each object having all attributes in X also has all attributes in Y. An implication can be read directly in the line diagram of a concept lattice in which the largest concept have intent M which contains X and Y. For example, the implication {Att1-v1, Att1-v2} ⇒ {Att4-v2, Att3-v1} holds in Figure 15a.

The association rule as visualised in concept lattice can be used to identify IID. This can be achieved by comparing the association rule in the concept lattice of a subunit data set with that of the super or master data set. A master data set refers to a dataset that incorporates data from sub units. An example of a master and its subunits data sets are the data set held by the central administrative office of an organisation and its departmental data sets. In principle, incomplete data in the data set from a central administrative office can be identified by comparing the association rule from its concept lattice with the departmental data sets.

The application of the association rule in FCA can provide a means to easily visualise missing or incomplete data. Figures 15a and 15b illustrate how IID can be visualised given the concept lattices from a sub data set (departmental data set) and the master data set. Evidently, the implication {Att1-v1, Att1-v2} ⇒ {Att4-v2, Att3-v1} of Figure 15a does not hold in Figure 15b. A visual analysis of the implication {Att1-v1, Att1-v2} ⇒ {Att4-v2, Att3-v1} in Figure 15a and the implication {Att1-v1, Att1-v2} ⇒ {Att3-v1} holding in Figure 15b reveals that Att4-v2 is missing in the central dataset. Att4-v2 is identified as the missing attribute while Obj3 is identified as the incomplete object.

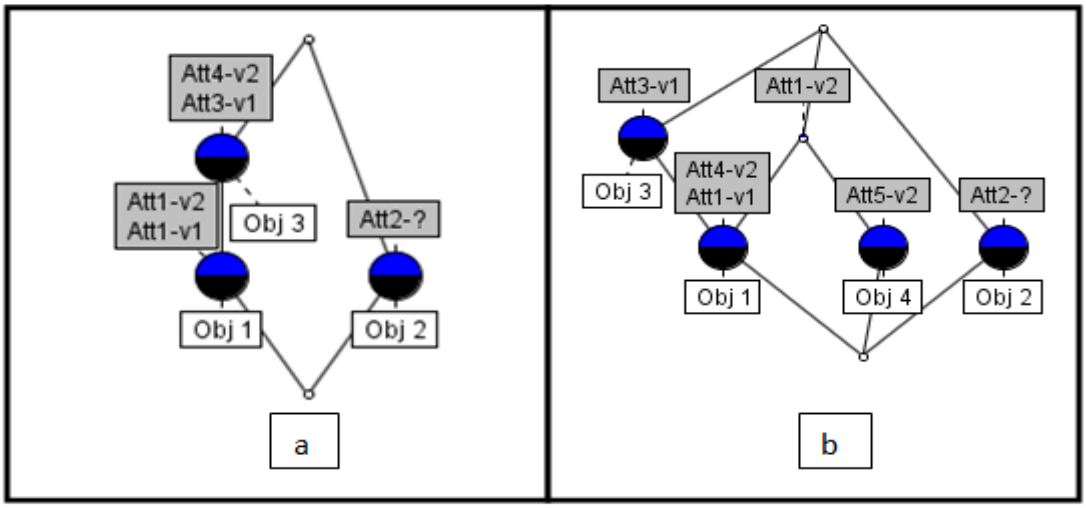

Figure 15: Concept lattice of a formal context from a dummy departmental data set (a) and concept lattice of a formal context from a dummy administrative data set (b)

The application of the association rule in identifying incomplete attributes or objects as explained in this section is yet to be explored in FCA literature. Unlike attribute exploration which depends on the knowledge of the domain expert to identify the incomplete data, the association rule approach provides a mechanism for comparing different data sets. This process of comparing concept diagrams from the master data store with the concept diagrams from the subunit data set is in accordance with OWA as there are no claims of the existence of complete data in the investigated or central dataset. The association rule will not be considered in subsequent chapters of this work because EMAGE do not have a master data set. Nevertheless, this approach provides further opportunity to explore how FCA can be used in the identification and visualisation of IID existing in a noisy data set.

Similarly, the In-Close (an FCA tool) will not be used in this work. This is because it does not meaningfully deal with the IID existing in a noisy data set. In-Close can be used to reduce the number of formal concepts in a context file. It functions by requesting from its user, a minimum number of objects and a minimum number of attributes. It uses this information to mine for formal concepts that have this minimum support. Andrews et al. (2011) demonstrate how In-Close can be applied in a large data set. In-Close can be used to produce a manageable number of concepts from a large formal context. However, the approach used in In-Close is not considered in this work as appropriate to deal with IID. This is because, the approach in In-Close uses a trial and error approach where the user inputs any minimum number of objects or attributes. Such approach is not considered in this work as a semantic approach of

dealing with IID. As a result, it will not be considered in subsequent Chapters of this work.

7.4.2 Semi-automated and automated FcaBedrock Approach

The semi-automated and the automated FcaBedrock approaches are used to identify and visualise the IID existing in a noisy data set. They are based on mutually exclusive attribute value principle. An object in a data set is inconsistent when it is associated with an attribute which has mutually exclusive attribute values. Also, an object is incomplete when it is not associated with any of the mutually exclusive attribute values.

An object can contain many-valued attribute. For example, a *tissue* (object) can be associated with a *gene* (attribute) which can be *detected* or *not detected* (values); a *person* (object) can be addressed with a *title* (attribute) such as '*Mr.*', '*Dr.*', or '*Prof*'(values); a *student* (object) has a *grade* (attribute) which is either a '*pass*' or a '*fail*' (values). Some of these attribute values are mutually exclusive. For example, *pass* and *fail*. An object will be described as inconsistent when it is associated with many-valued attribute which are mutually exclusive. For example, there is inconsistency in a student's result whose grade is assigned a 'pass' and a 'fail'. Also, a tissue is inconsistent when it is associated with a gene that is both *detected* and *not detected*. Such objects will be incomplete when they are not associated with any of the mutually exclusive attribute values. This work recalls part of section 5.3.1 in this section for easy follow through. Also Figure 6 is depicted here as Figure 16.

The context of objects whose many-valued attributes are mutually exclusive can be defined to be consistent if each of its objects has only a single-value attribute from a mutually exclusive attribute values. If an object in such a context has two or more attribute values, then the context is inconsistent. For example, if the set of possible attribute values $M = \{v_1, v_2, \dots, v_n\}$, then let the set of objects with value v_1 be G_{v_1} and the set of objects with value v_2 be G_{v_2} and so on. A consistent context can be defined thus:

$$\forall v_i, v_j \in M \bullet v_i \neq v_j \Rightarrow G_{v_i} \cap G_{v_j} = \emptyset \quad (1)$$

where "$\bullet$" implies "then".

Consequently, an object which is associated with more than one value from an attribute is inconsistent.

The context is defined to be complete if every object in *G* has a value:

$$G = \cup\{G_{v_1}, G_{v_2}, \ldots, G_{v_n}\} \quad (2)$$

An object without an associated attribute value is therefore incomplete.

A concept lattice can be used to visualise consistency and completeness in a very clear manner. For example, if $M = \{v_1, v_2, v_3\}$ and $G = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9\}$ then examples of consistent and inconsistent lattices are shown in Figure 16(a-d). 16a illustrates consistent data, 16b illustrates inconsistent data where two objects have two values, 16c illustrates inconsistent data where one object has all three values and 16d illustrates incomplete data (objects without a value), which are clearly labelled at the topmost node of the lattice. For easy identification of IID in a concept lattice, it should be noted that an extent is identified by a lower filled semicircle labelled below the node, while the associated intent is identified by an upper filled semicircle in the same node or in an ascending path to the node. Extents which have attributes with more than one value are inconsistent and an extent which is not associated with an attribute value is incomplete.

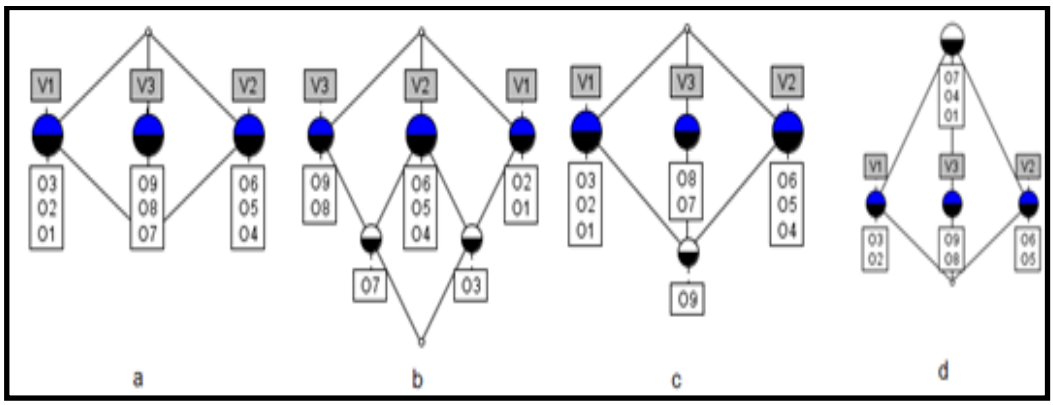

Figure 16: Examples of consistent (a), inconsistent (b and c), and incomplete (d) concept lattices

This work implemented the above concept by using free FCA tools to separate out and exclusively visualise IID in objects whose attribute values are mutually exclusive. This is implemented through the semi-automated FcaBedrock or the automated FcaBedrock approach as explained below.

Semi-automated FcaBedrock Approach

The semi-automated FcaBedrock approach semi-automatically separates out (restricts) attributes associated with single-value when transforming a record set to a context file. In the semi-automated FcaBedrock approach, IID are identified in a data set through the use of FcaBedrock and ConExp. This is done by reading a many-valued attribute into an FcaBedrock application and manually changing the 'y' to 'n' of those attributes with one category value in the FcaBedrock's editing environment. The essence of this editing is to restrict single-valued attributes, thereby converting other attributes to a formal context. Concept Explorer is subsequently used in building, visualising and also editing the concept lattice from the context file**.** Figure 17 illustrates this approach.

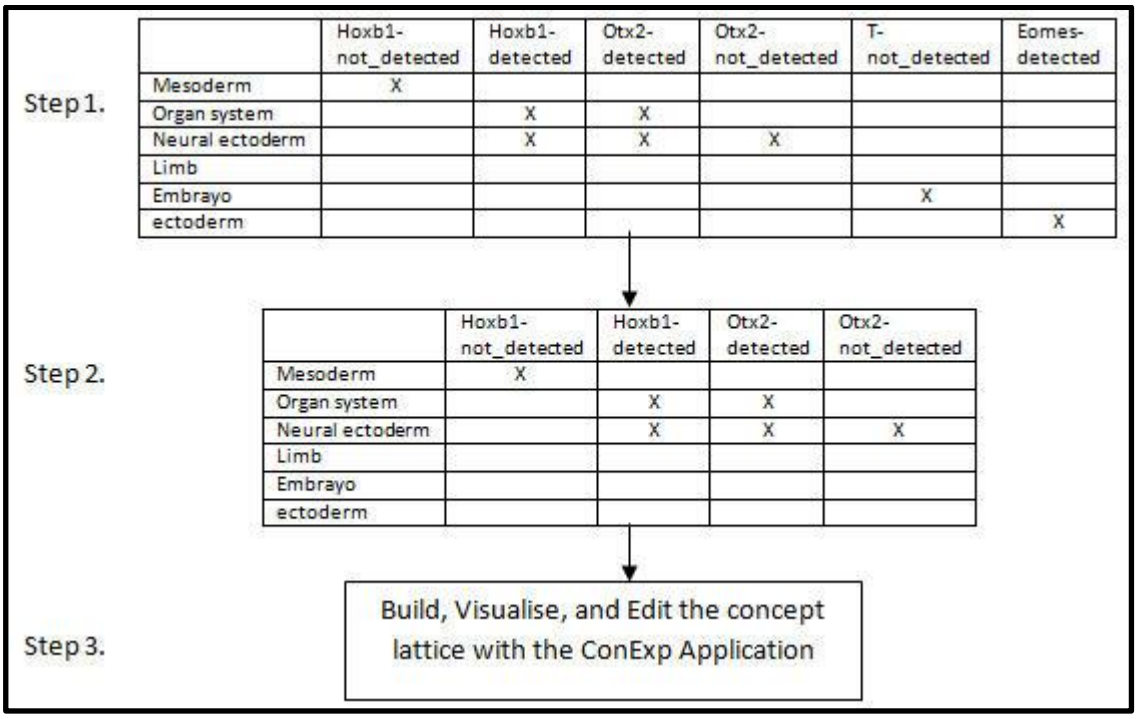

Figure 17: An illustration of the semi-automated FcaBedrock approach

The step 1 of Figure 17 is the corresponding formal context of an FcaBedrock input file. Ideally, the context file retrieved from an RDF database is often in a many-value context but can be converted to a single-value context as described in chapter 4. The many-value context is read and processed by the FcaBedrock application. It is presented in a single-value context in Figure 17 for easy follow-through. Step 2 is the formal context of the output file from the FcaBedrock application. Step 3 informs the reader that the ConExp (a context visualisation tool) will be used to further edit the output file from the FcaBedrock.

It can be noted that not all the attributes in step 1, are converted to the final context in step 2 as depicted in Figure 17. FcaBedrock provides an approach that enables the restriction of single-valued attributes when converting a many valued-record set to a context file. Figure 18 (see below) shows how this was done in FcaBedrock. The 'convert' attributes of *T* and *Eomes* in Figure 18 are changed to 'n'. This change has the effect of not transforming these single-value attributes to a formal context. Consequently, single valued attributes are limited from the converted context.

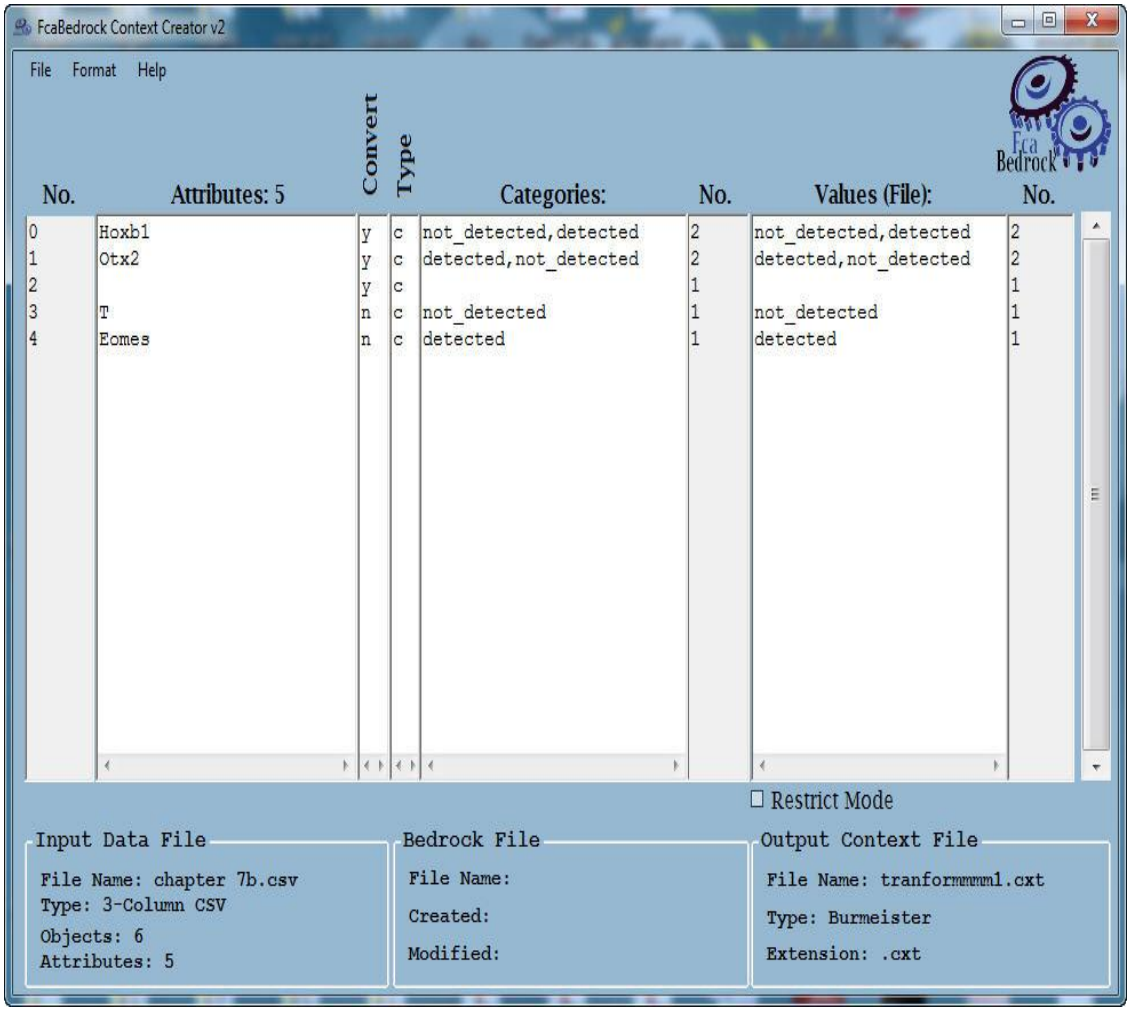

Figure 18: Semi-automated FcaBedrock processing approach

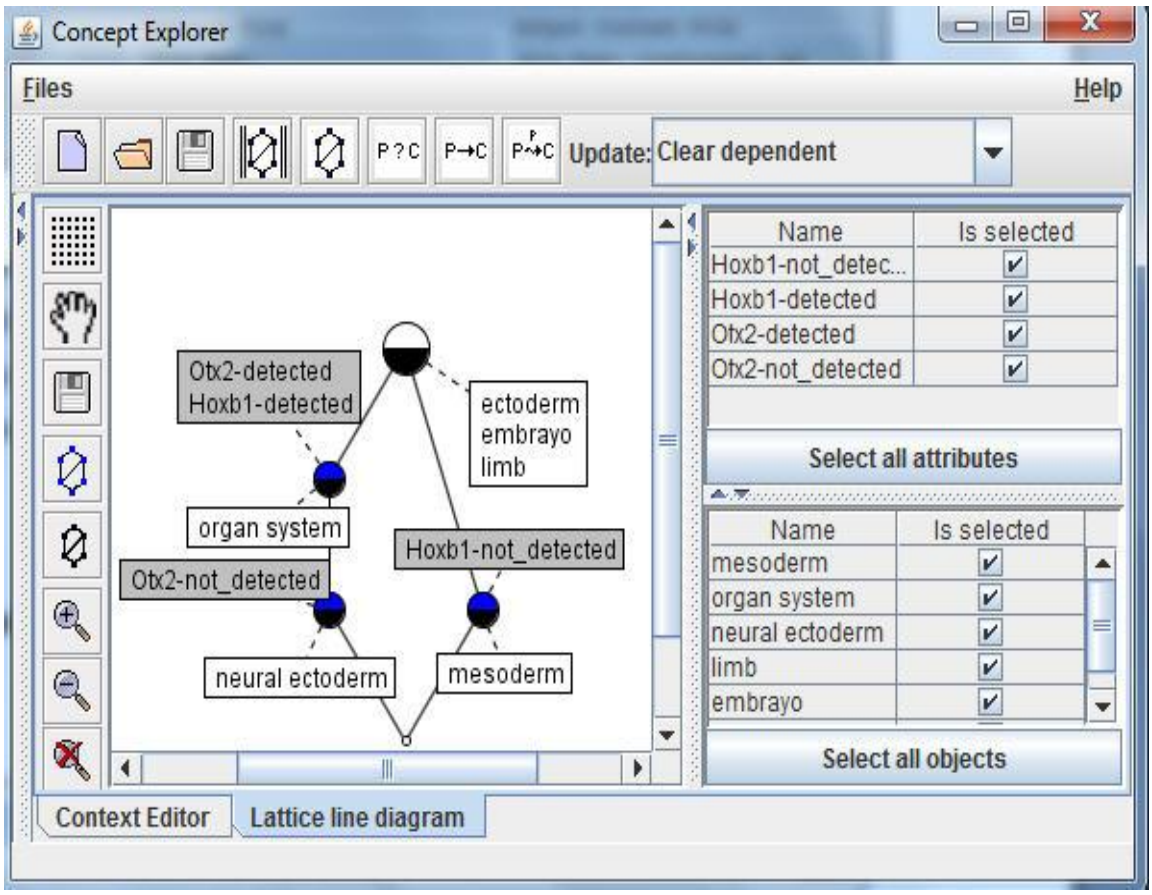

Figure 19:Concept lattice built from the output file from the semi-automatic FcaBedrock approach

Even so, there is still a need to edit the concept lattice built from the FcaBedrock's output file (see step 3 of Figure 17). This is because not every many-valued attribute is inconsistent. In Figure 19, all the objects which are not associated with any attributes as depicted at the topmost node are incomplete. It is easy to visualise the incomplete data in Figure 19 but the inconsistent datum is not clearly depicted. The data analyst therefore, deselects attribute values which do not have contradictory attribute-value. Consequently, the data analyst can visualise only the IID in the data set when the concept lattice is edited through deselecting the appropriate boxes as shown in the left hand side of the ConExp (see Figure 20). Figure 20 presents an exclusive view of IID in the investigated data set.

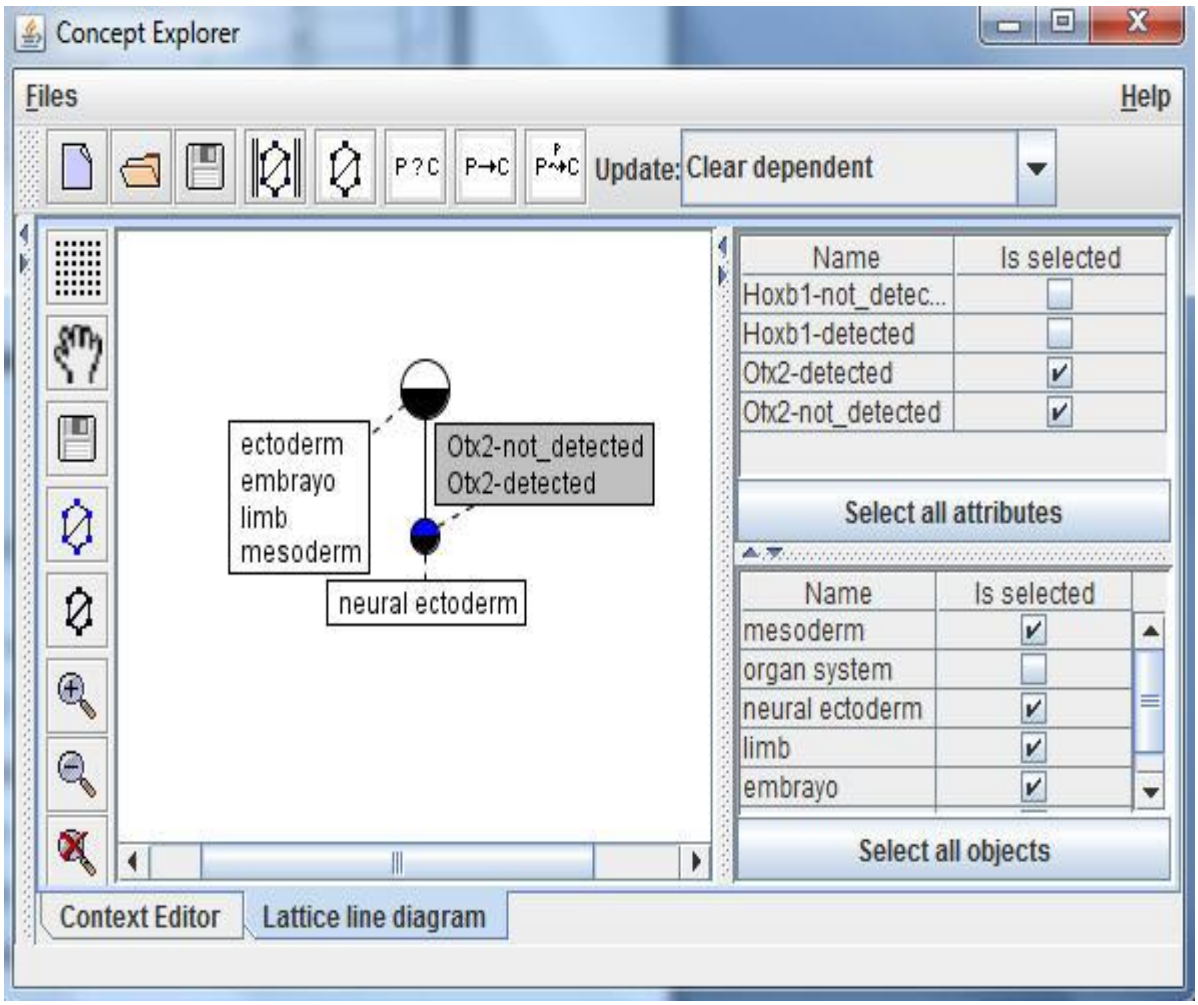

Figure 20: Editing a Concept lattice in the ConExp

Similar to the semi-automated FcaBedrock approach is the approach presented in (Andrews and Orphanides 2010), where it is shown how a data analyst can use FcaBedrock to analyse a data set by restricting the formal context conversion to only the data attributes of interest for a particular analysis. Also, Jiang et al., (2009) used a node without a label for its own object in a concept lattice to show anonymous node. An 'anonymous node' means that an own object label is missing from the node (Jiang et al., 2009).

The use of the semi-automated approach as described in this section is painstaking and time consuming. It is implemented by manually excluding single attribute values in FcaBedrock and also editing the concept lattice to separate out and exclusively visualise the IID in the data set. This approach exclusively identifies IID but it might not be appropriate for identifying IID in a large dataset. Hence there is a need for an automated FcaBedrock approach.

Automated FcaBedrock Approach

The automated FcaBedrock approach is an IID identification and visualisation approach. It involves an automatic transformation of IID from a record set to a context file and a subsequent visualisation of the context file through a context visualisation tool. Similar to the semi-automated FcaBedrock approach is the automated FcaBedrock approach. The automated FcaBedrock approach is applied to separate out and exclusively visualise IID in objects whose attribute values are mutually exclusive. Again, this work recalls part of section 5.3.1 in this section for easy follow through. Figure 7 is depicted here as Figure 21.

The automated FcaBedrock approach is implemented by extending the FcaBedrock application. This extension enables the FcaBedrock to automatically convert only objects and their associated many valued attribute to a context file. As explained earlier, a many-valued context containing mutually exclusive attribute-values which do not adhere to equation 1 and 2 above is either inconsistent or incomplete. IID can be separated out and exclusively visualised from a noisy data set through the use of the automated FcaBedrock approach. For example, if *M* = {a-1, a-2, a-3, a-4} and *G* = {o1, o2, o3, o4}, then let the set of objects with value $a-1$ be $o1_{a-1}$ and the set of objects with value $a-2$ be $o2_{a-2}$ and so on. Examples of inconsistent and incomplete lattices are shown in Figure 21(a-d). 21a and 21b illustrates inconsistent and incomplete data; 21c illustrates inconsistent data where each of the three objects has contradictory values; and 21d illustrates incomplete data where attribute values without an associated object are displayed. Again, extents which have attributes with more than one value are inconsistent. An intent which is not associated with an object is incomplete.

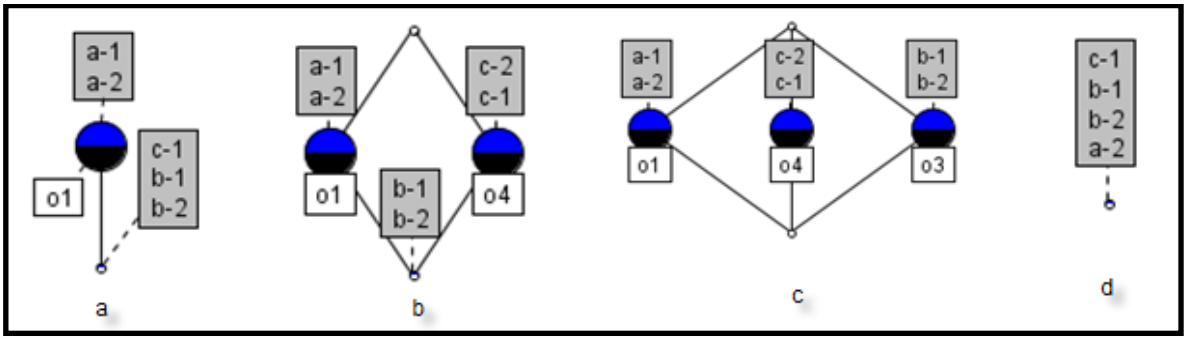

Figure 21: Examples of (a, b) inconsistent and incomplete, (c) inconsistent, and (d) incomplete concept lattices

Figure 22 (see below), illustrates how the automated FcaBedrock approach is applied. In Figure 22, the formal context (see step 1) is the corresponding context file of a CSV or 3 column file. Again, the context file retrieved from RDF database is usually in a many-value context but it is represented with the corresponding single-value attribute context for easy follow-through. This file is read and processed by the FcaBedrock application. The second formal context (see step 2) is the corresponding context of the output file from the FcaBedrock. The context is actualised if the inconsistency mode was selected in the FcaBedrock during its many to single context conversion process as evident in Figure 23. It can be observed from the step 2 of Figure 22, that the object '*neural ectoderm'* and associated attribute values were converted to a formal context. This is because '*neural ectoderm'* is the only object whose attribute is associated with many values. The other objects such as *Mesoderm*, *Organ system*, *Limb*, *Embrayo*, and *ectoderm* are not associated with a many-value attributes. Also, it can be noted that all the attribute-values in the input context are transformed to the single-value context.

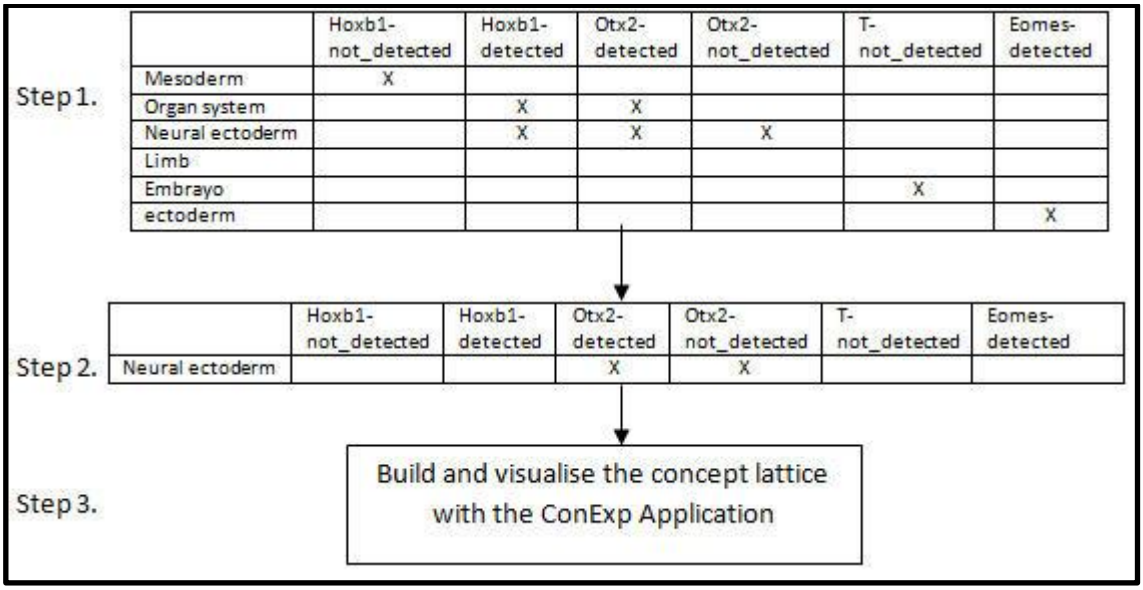

Figure 22: An illustration of the automated FcaBedrock approach

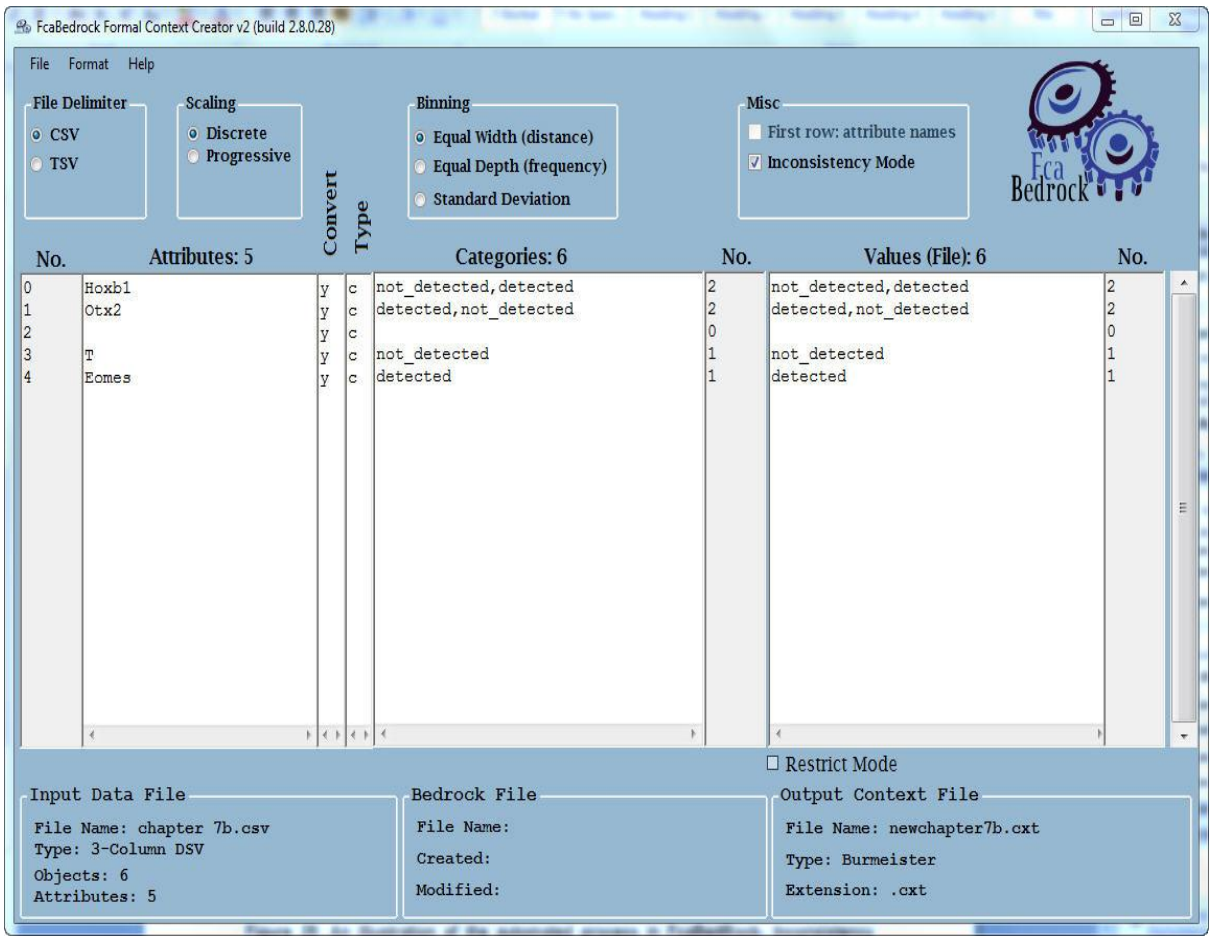

Figure 23: Automated FcaBedrock processing approach

Figure 24 shows the concept lattice built from the output file of the FcaBedrock. This is done through the use of the ConExp application. In the concept lattice, all the attributes that are not associated with any objects i.e. attributes depicted at the bottom node are incomplete while the object associated with contradictory attribute-values is inconsistent.

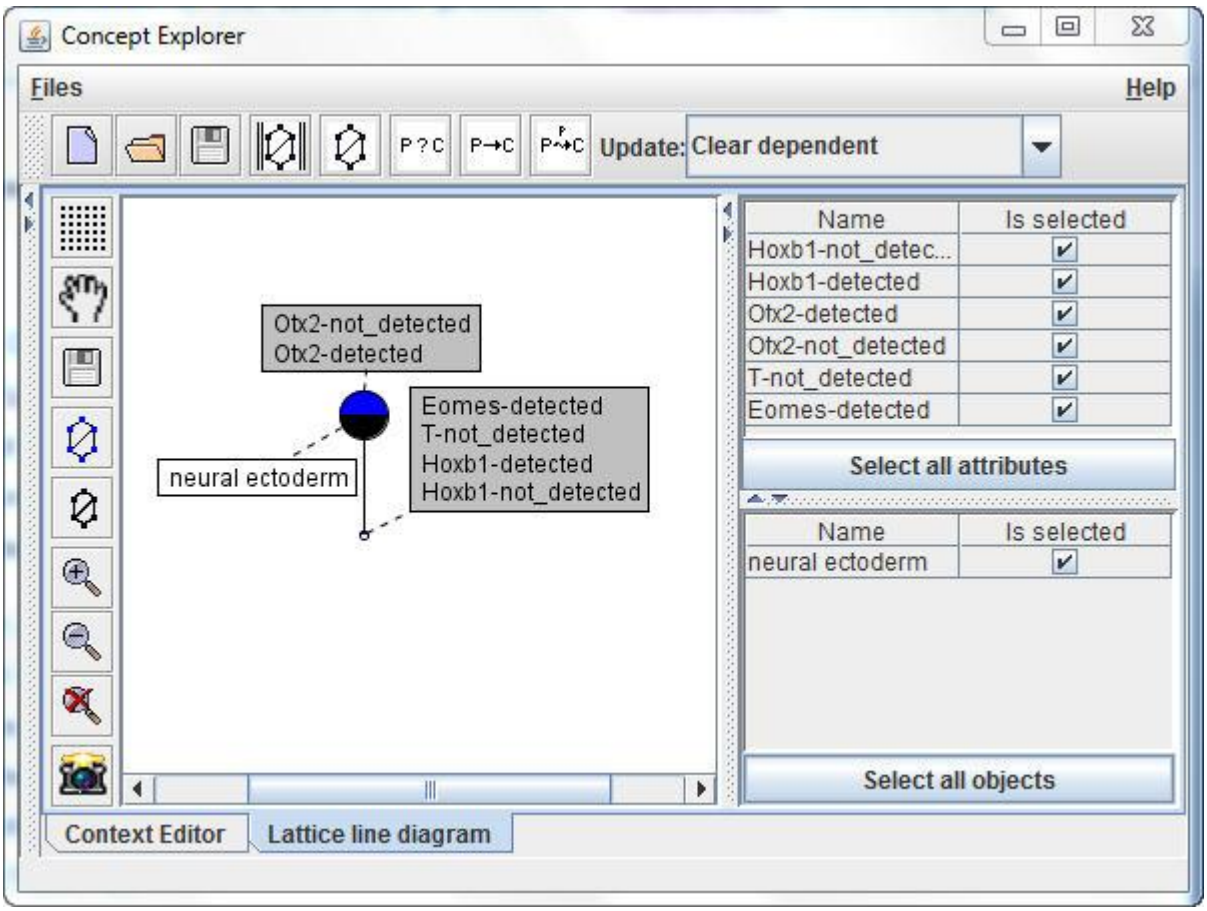

Figure 24: Concept lattice in the ConExp application

The automated FcaBedrock approach provides an exclusive view of IID existing in a data set. This work collaborated with the FcaBedrock developer's team to extend the FcaBedrock with the inconsistency mode. The pseudocode for this extension is depicted in Figure 25 below. This pseudocode was implemented by an FcaBedrock developer.

```
1:    K= (G, M, I)
2:    Let G = all objects
3:    Let M = all attributes
4:    Let O(q,w) = empty list where q = inconsistent object index and w = inconsistent attribute values
5:    Let counter = counter to count attribute-values //(i.e. all values of EACH attribute) for each object
6:    For every object i in G
7:       counter = 0
8:       For every attribute j in M
9:             For every attribute-value x in M[j]
10:                  If G[i] contains x then counter = counter + 1
11:            End For
12:      End For
13:      If counter > 1 then
               //inconsistency right here – object G[i] has more than one value
               //for attribute M[j], so save it's index and the attribute which renders
               //this object inconsistent in O
14:            O.add(G[i], M[j])
15:      End If
16: End for
//So now O contains all objects along with the attributes which render each object inconsistent.
```

Figure 25: Pseudocode for the Inconsistency Mode in FcaBedrock

7.5 Key Messages and Findings

This chapter explained how a record set containing IID can be retrieved from an RDF triple store. It identified attribute exploration, association rule, fault tolerance, Dau, CUBIST, the semi-automated FcaBedrock, and the automated FcaBedrock approaches as means of dealing with IID. It explained that the use of an attribute exploration does not conform to the OWA principles; hence it will not be discussed further as an FCA approach for dealing with IID in RDF data set. Fault tolerance provides a useful means of reasoning with a noisy and incomplete data set. This work could not secure licence of a fault tolerance FCA application. Consequently, fault tolerance will not be evaluated further in this work. Similarly, association rule will not be evaluated further in this work because there are no master data in the EMAGE use case.

This chapter explained how Dau approach, CUBIST approaches, the semi-automated FcaBedrock approach and automated FcaBedrock approach are used in dealing with IID in an RDF data set. These approaches were investigated further through the use of EMAGE RDF data as documented in chapter 8.

Chapter 8: Experiments

8.1 Introduction

This Chapter presents the results of the different FCA approaches which were used to identify and visualise the IID existing in EMAGE RDF data set. An EMAGE RDF data set was examined for instances of IID. The FCA approaches that were used in investigating the data set include: Dau's approach (Dau 2013a), CUBIST approaches (Melo et al. 2013), the semi-automatic FcaBedrock approach (Nwagwu 2014), and the automated FcaBedrock approach (Nwagwu and Orphanides 2015).

This chapter describes the experiments carried out on EMAGE RDF data set under 4 different heading namely, Introduction which describes the approach of the experiment; Application which outlines the software application packages used in the experiment; Queries and experimental results which presents the queries applied and the results obtained; and Summary which provides concluding statements about the approach. The use of Dau's and CUBIST approaches to visually identify the IID in the EMAGE RDF data set are presented in section 8.2 and 8.3 respectively while how the semi-automated and the automated FcaBedrock approaches are used in identifying and visualising the IID in the EMAGE RDF data set are presented in section 8.4 and 8.5 respectively.

8.2 Dau's Approach- SPARQL2context Creator

8.2.1 Introduction

Dau's approaches for dealing with IID are documented in (Dau 2013a; Dau 2013b). This section focuses on Dau (2013a) which explains how the SPARQL2Context creator tool is used in retrieving IID from EMAGE RDF data set. Dau (2013a) explains how *optional-clause* and *union-clause* can be used in retrieving IID from a noisy dataset. It describes the SPARQL2Context creator tool, the use of the optional and *union* keywords as detailed in chapter 7 of this work.

8.2.2 Application

The applications used in this approach include the SPARQL2Context creator, Owlim-SE and a concept lattice builder

8.2.3 Queries and experimental results

Finding and visualising incomplete data in EMAGE RDF data set

Table 11 is a SPARQL query used in retrieving from EMAGE RDF data set, record set containing incomplete data as documented in Dau (2013a).

Table 11: A reproduction of the query in Table 4 of (Dau 2013a) showing how the union keyword is used in identifying IID

```
Select Distinct ?obj ?att where {
{?x1 rdf:type :Tissue ; rdfs:label ?obj .
?x1:has_theiler_stage :Theiler_stage_TS07 . }
UNION
{?x1 rdf:type :Tissue ; rdfs:label ?obj .
?x1 :has_theiler_stage :Theiler_stage_TS07 .
?x3 rdf:type :Gene ; rdfs:label ?att .
?x2 rdf:type :Textual_Annotation .
?x2 :in_tissue ?x1 .
?x2 :has_involved_gene ?x3 .
?x2 :has_strength :level_detected_derived . }
}
Order by ?obj ?att
```

The term *level_detected_derived* is a derived property from detected, strong, moderate, or weak. Any gene expression that is associated with *level_detected_derived* can be detected, strong, or weak. Table 11 lists all the tissues with certain attributes such as Theiler Stage 07 and *level_detected_derived*. It utilises the keyword '*union*' to retrieve its record set from EMAGE data set. The query is an example of object-unrestricted query (see section 7.3.1). Figure 26 shows the corresponding concept lattice built from EMAGE record set retrieved with the query.

It can be seen from Figure 26 that the incomplete data are depicted as objects without an associated attribute. These objects are displayed at the topmost node of the lattice. It should be noted that each of the EMAP:IDs such as EMAP:56, as depicted in Figure 26 is a unique tissue in EMAGE data set.

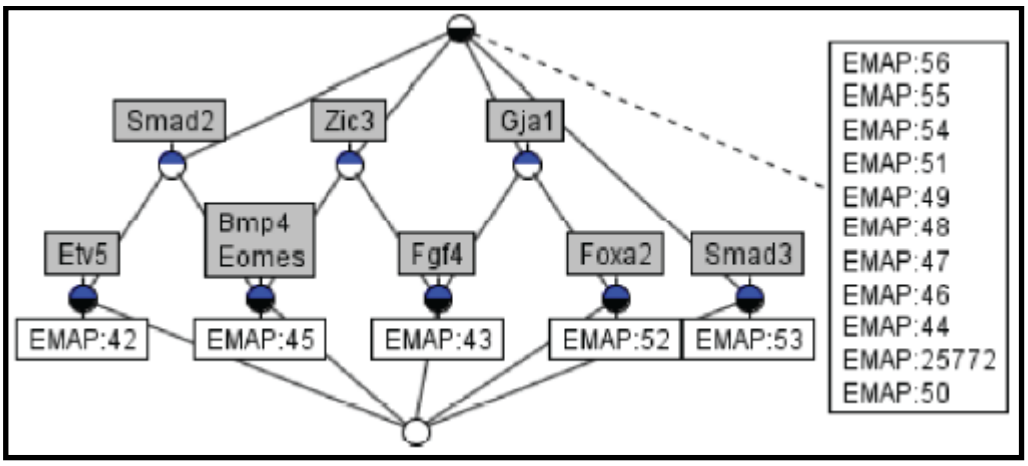

Figure 26: Concept lattice of results retrieved by the query in Table 16 as depicted in Figure 5 in (Dau 2013a)

Table 11 can be modified to depict the information about the strength of the tissues. Figure 27 shows the corresponding concept lattice of a record set retrieved from the EMAGE data set when the query in Table 11 includes patterns depicting the strength of the tissues. Again, the incomplete data in Figure 27 are depicted at the topmost node of the lattice, as objects without an associated attribute.

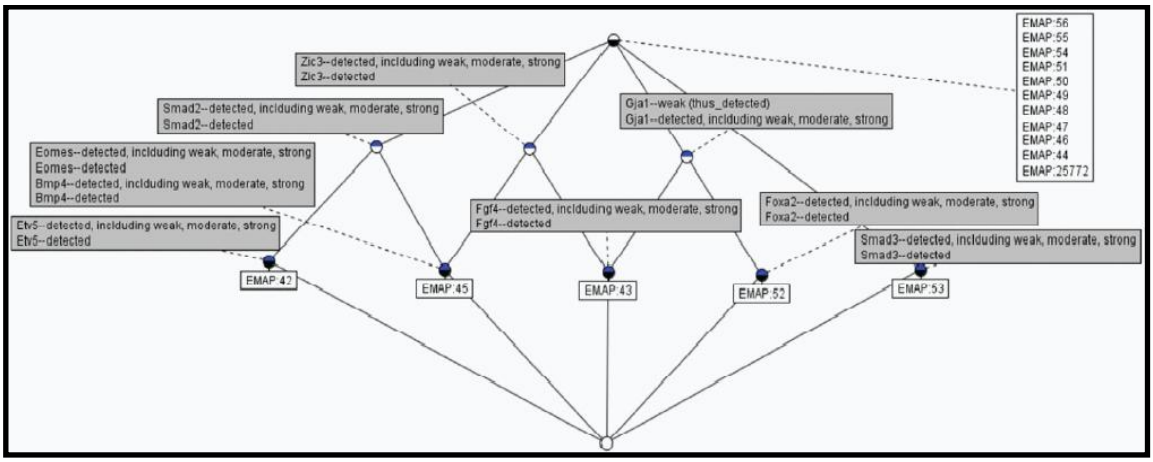

Figure 27: Concept lattice showing incomplete data in Tissues and associated gene expressions as depicted in Figure 6 in (Dau 2013a)

Finding and visualising contradictions in EMAGE RDF data set

The query in Table 12 (see below) is used to retrieve a record set which contain contradictory data from EMAGE RDF data set.

Table 12: A reproduction of the query in Table 7 from Dau (2013a) designed to retrieve contradictory data from EMAGE RDF data set

```
Select distinct ?o0 ?a0 where {
  ?x0 rdf:type :Tissue ; rdfs:label ?o0 .
  ?x1 rdf:type :Tissue ; rdfs:label ?a0 .
  ?x2 rdf:type :Gene ; rdfs:label ?o1 .
  ?ta1 :in_tissue ?x0 ; :has_involved_gene ?x2 ; :has_strength :level_detected_derived .
  ?ta2 :in_tissue ?x1 ; :has_involved_gene ?x2 ; :has_strength :level_not_detected .
{
{?x0 :is_part_of ?x1 . Filter (!sameTerm(?x1, ?x0)) }
UNION
{Filter(sameTerm(?x0, ?x1))}}}
```

The query in Table 12 retrieves pairs of tissues that are limited to particular attributes and also have '*is_part_of*' association. The query propagates the gene expressions in tissues. This implies that the retrieved record set may contain tissues whose gene expression contradicts its related tissues. Figure 28 below, shows the corresponding concept lattice of a record set retrieved from the EMAGE data set when the query in Table 12 is applied to EMAGE RDF database.

In Figure 28, it is easy to visualise the association of related tissues. A tissue associated with more than one tissue could possibly imply that the tissues have contradictory expression levels. This can be visualised in Figure 29 where the gene associated to each tissue is displayed.

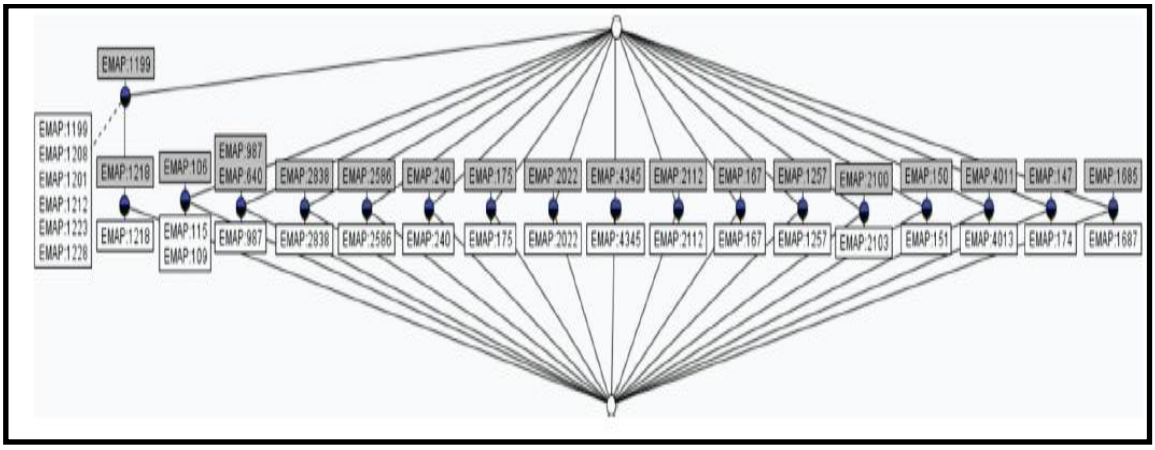

Figure 28: Concept lattice showing Tissues with contradicting textual annotations as depicted in Figure 7 of (Dau 2013a)

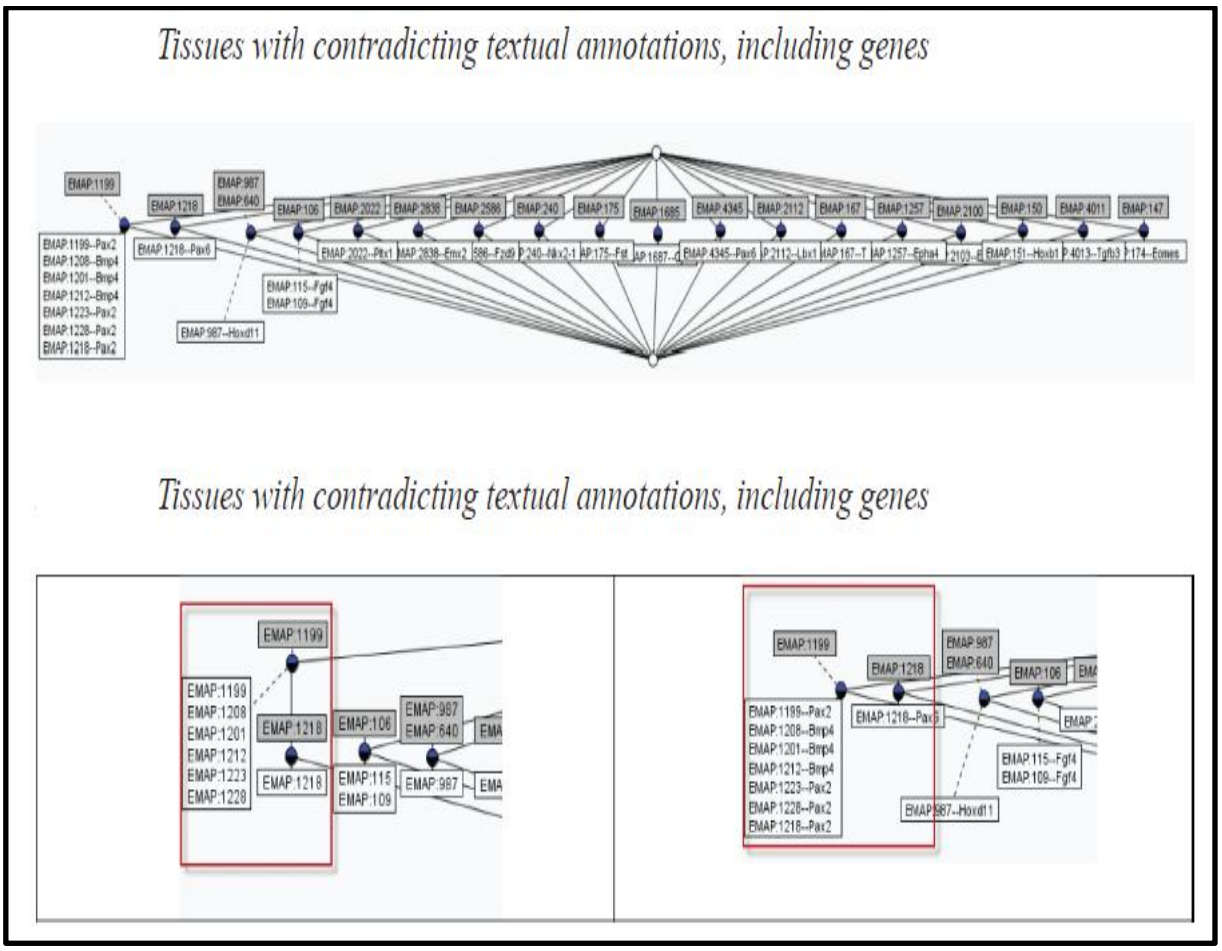

Figure 29: Tissues with contradicting textual annotations including genes as depicted in Figure 8 and 9 in (Dau 2013a)

Figure 30 shows the genes associated to each tissue and their corresponding Theiler Stage. This information enables the data analyst to identify the tissues that have contradictory genes in particular Theiler Stage. For example, it can be observed from Figure 30 that there is a contradiction between EMAP:1218 and EMAP:1119 in TS15 where *Pax2* is the contradicting gene. Table 13 shows the corresponding query.

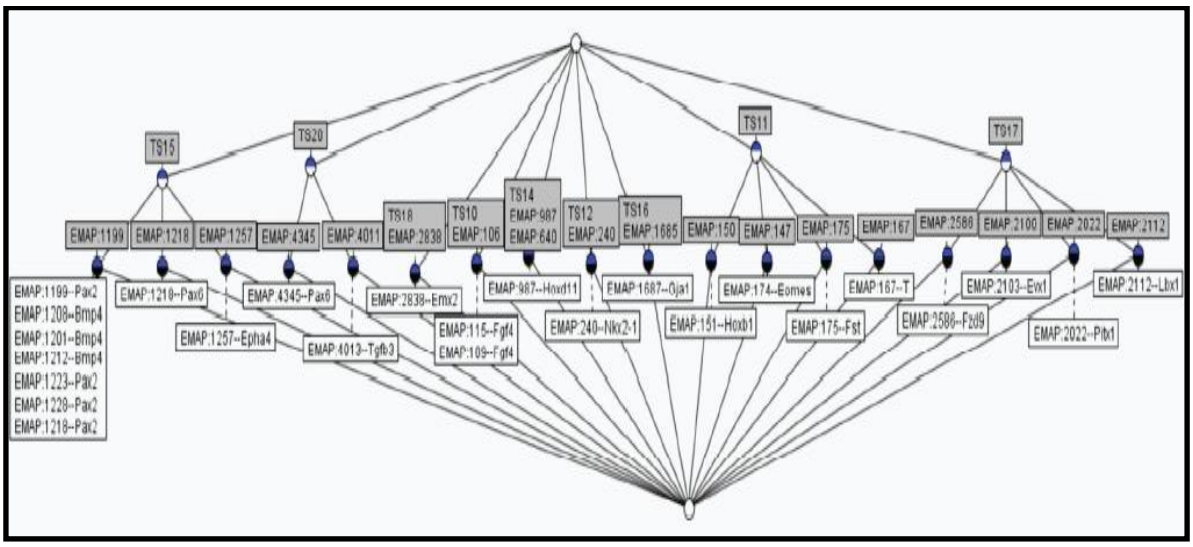

Figure 30: A reproduction of the query in Figure 10 of (Dau 2013a) showing concept lattice containing contradicting pairs of tissues where genes, tissues, and TS are included in the diagram

Table 13: A reproduction of the query in Table 8 of (Dau 2013a) showing how the use of the union keyword

```
Select distinct ?o0 ?o1 ?a0 where
{
?x0 rdf:type :Tissue ; rdfs:label ?o0 .
?x2 rdf:type :Gene ; rdfs:label ?o1 .
{
?x1 rdf:type :Tissue ; rdfs:label ?a0 .
?ta1 :in_tissue ?x0 ; :has_involved_gene ?x2 ; :has_strength
:level_detected_derived .
?ta2 :in_tissue ?x1 ; :has_involved_gene ?x2 ; :has_strength :level_not_detected .
{{?x0 :is_part_of ?x1 . Filter(!sameTerm(?x1, ?x0))} UNION { Filter(sameTerm(?x0,
?x1))}}
}
UNION
{
?x1 rdf:type :Tissue .
?x1 :has_theiler_stage ?ts1 .
?ts1 rdfs:label ?a0 .
?ta1 :in_tissue ?x0 ; :has_involved_gene ?x2 ; :has_strength
:level_detected_derived .
?ta2 :in_tissue ?x1 ; :has_involved_gene ?x2 ; :has_strength :level_not_detected .
{{?x0 :is_part_of ?x1 . Filter(!sameTerm(?x1, ?x0))} UNION { Filter(sameTerm(?x0,
?x1))}}
}
}
```

8.2.4 Summary

Dau's approach provides comprehensive details on how the EMAGE RDF data set can be queried for data containing IID. It also presents how the retrieved record set can be visualised for instances of IID in a concept lattice. However, its concept lattices can be bulky and may be difficult to read.

8.3 CUBIST Approach

8.3.1 Introduction

CUBIST integrates elegant approaches in dealing with RDF data. It uses a set of pre-defined queries to query the ontology of EMAGE RDF data set, which are then converted to a formal context and associated chats. CUBIST deals with IID existing in its explored EMAGE RDF data set through approaches such as utilizing distinct colour, interactive exploration, fault tolerance, and co-occurrence. These approaches have been briefly explained in chapter 7.

8.3.2 Application

The CUBIST is an integrated application. Applications such as the FcaBedrock and the In-Close are embedded in CUBIST.

8.3.3 Queries and Experimental Results

CUBIST uses a set of pre-defined queries to query the ontology of EMAGE data set. These queries are designed to explore the data set according to the user's interactive exploration. CUBIST also has options to display the associated lattice and charts. Figure 31 and Figure 32 below depicts a tabular result and the associated charts of such exploration as presented in (Melo et al. 2013).

	Attribute	Values	Objects
☑	embryo	level_detected , level_not_detected , level_weak ,	Arc, Foxa2, Smad2, Otx2, Smad5, Smad1
☑	endoderm	level_detected , level_not_detected ,	Foxa2, Zic2, Zic3
☑	definitive endoderm	level_detected ,	Hhex, Cer1
☑	parietal endoderm	level_strong ,	Fst
☑	extraembryonic ectoderm	level_detected ,	Bmp4, Zic2
☑	ectoderm	level_detected ,	Wnt3, Smad5, Otx2, Eomes, Zic2, Tdgf1, Zic3, T
☑	mesoderm	level_detected ,	Wnt3, Smad5, Lefty2, Eomes, T, Mesp1, Tdgf1
☑	primitive streak	level_detected , level_strong , level_not_detected ,	Lefty2, Eomes, Gsc, Foxa2, T, Fgf8, Mesp1, Fst, Smad1, Otx2
☑	primitive endoderm	level_detected , level_not_detected ,	Hhex, Wnt3, Fgf8, Smad2, Otx2, Gsc, Foxa2, Cer1, Sox17
☑	visceral endoderm	level_strong , level_detected ,	Smad1, Foxa2, Sox17
☑	extraembryonic component	level_weak , level_strong , level_detected ,	Smad5, Bmp4, Arc, Smad2

Figure 31: Genes, tissues and level of expression in Theiler Stage 9 as depicted in Figure 1 in (Melo et al. 2013)

Figure 32: CUBIST user interface displaying the concept lattice for genes, tissues and level of expression in Theiler Stage 9 as depicted in Figure 2 in (Melo et al. 2013). Its main components: 1) Toolbar; 2) Visualisation canvas; 3) Dashboard; and 4) Selection

Identifying and visualising IID in EMAGE RDF data set

CUBIST provides its users with options such as co-occurrence tab and filter buttons (see Figure 31 and 32 above). These options enable CUBIST user to identify and visualise IID in EMAGE data set. Figure 31 depicts CUBIST filter buttons which is used to retrieve Genes, tissues and level of expression in Theiler Stage 9.

In Figure 31, it can be observed that some genes have contradictory values in their associated tissues. For example, *Arc*, *Foxa2*, *Smad2*, *Otx2*, *Smad5*, and *Smad1* are genes associated with contradictory expression levels (detected, not detected, and weak) in the tissue *embryo* of the Theiler Stage 09. The entire selected objects of the figure can be visualised as a concept lattice or chart. Figure 32 depicts the corresponding concept lattice and bar chart.

CUBIST utilises distinct colours to emphasis the data conflict in EMAGE according to the inconsistency type (Melo et al. 2013). For example, a red colour can be used to indicate that a gene is both detected and not detected (binary inconsistency) in the same tissue at the same Theiler Stage.

CUBIST also uses its co-occurrence tab (see Figure 32) to select two sets of attributes to identify objects that are shared by the two attributes. The result of such exploration is displayed in a concept lattices and also in an associated chart as described in a CUBIST YouTube[25] presentation. In so doing, contradictory attributes can be selected and objects sharing such attributes can be visualised.

Reasoning with IID in EMAGE RDF data set

CUBIST provides different matrices for reasoning with the incomplete or inconsistent EMAGE RDF data. Some of CUBIST reasoning methods include automatically fixing of flip-fops and fault tolerance technique. Flip-fops occur in EMAGE when a gene is not expressed in Theiler Stage, say *K,* but expressed in a preceding Theiler Stage say *K – 1* and also in the subsequent Theiler Stage, say *K +1*. For example, a gene is not expressed in the *limb* in Theiler Stage 15, but it is expressed in the *limb* in Theiler Stage 14 and 16. Flip-flops are indicators of either missing or incorrect data in the database (Melo et al. 2013). Such missing data are automatically included in CUBIST. CUBIST also applies fault tolerance as a means of inferring missing data so that the missing crosses in the formal context can be assumed to exist. This is described in (Dau 2013b; Andrews and McLeod 2013).

8.3.4 Summary

CUBIST methods were developed by professional researchers over many years. Its approaches are principled and provide the basis for other researchers to develop further techniques. The CUBIST application provides IID approaches such as the use of the co-occurrence tab, use of the filter buttons, the use of distinct colour to depict particular type of inconsistence, and the automatic fixing of incomplete data through the flip-fops technique.

8.4 Semi-automated FcaBedrock Approach

8.4.1 Introduction

A subset of a non-propagated EMAGE RDF data set was stored in Owlim-SE triple store. The data set has 1,216,277 triples. Section 6.5 provides a description of the

[25] https://www.youtube.com/watch?v=Kuu756nr1_I

EMAGE RDF data set. SPARQL queries were applied to the stored data set to retrieve tissues, associated genes and their corresponding gene expressions of a particular Theiler Stage. The applied queries were designed with the aim of retrieving record sets whose objects (tissues) have attribute values (genes and associated expressions) which are mutually exclusive.

As discussed in chapters 2 and 7, a binary form of IID exists when the same object is associated with attribute values that have opposite meanings. An example of attribute values that have opposite meanings in EMAGE is 'gene-detected' and 'gene-not detected' where gene is the attribute while 'detected' and 'not detected' are the values. An analogue type of IID exists when the same object is associated with attribute values that are slightly contradictory. Examples of attribute values that are slightly contradictory in EMAGE are 'gene-weak_expression' and 'gene-medium_expression' where gene is the attribute; weak expression and medium expression are the values. McLeod and Burger (2011) explain that "*there are two definitions of inconsistency: binary (expressed versus not expressed) and analogue (e.g. strong expression is distinct from weak expression despite both levels suggesting a gene is expressed)*." Each of these distinct groups of gene expression (binary or analogue gene expression) is mutually exclusive. An inconsistency will exist in a tissue of a Theiler Stage where its gene is associated with more than one expression level from any of the distinct groups of gene expression.

Each retrieved record set from the EMAGE database is stored as a comma-separated value (CSV) file and subsequently read by the FcaBedrock application. FcaBedrock is used to restrict the single-valued attributes of the CSV file. Consequently, the many-valued and the no-valued attributes are converted to a formal context file. The ConExp is then used to build, visualise and edit the concept lattice. This semi-automated approach is also described in section 7.4.1. The summary of how the semi-automated FcaBedrock approach is applied on the EMAGE RDF data set is as follows:

- Each retrieved record set as shown in Figure 33, is stored as a three columns CSV file. Figure 33 shows an example of a query result retrieved from the Owlim-SE.

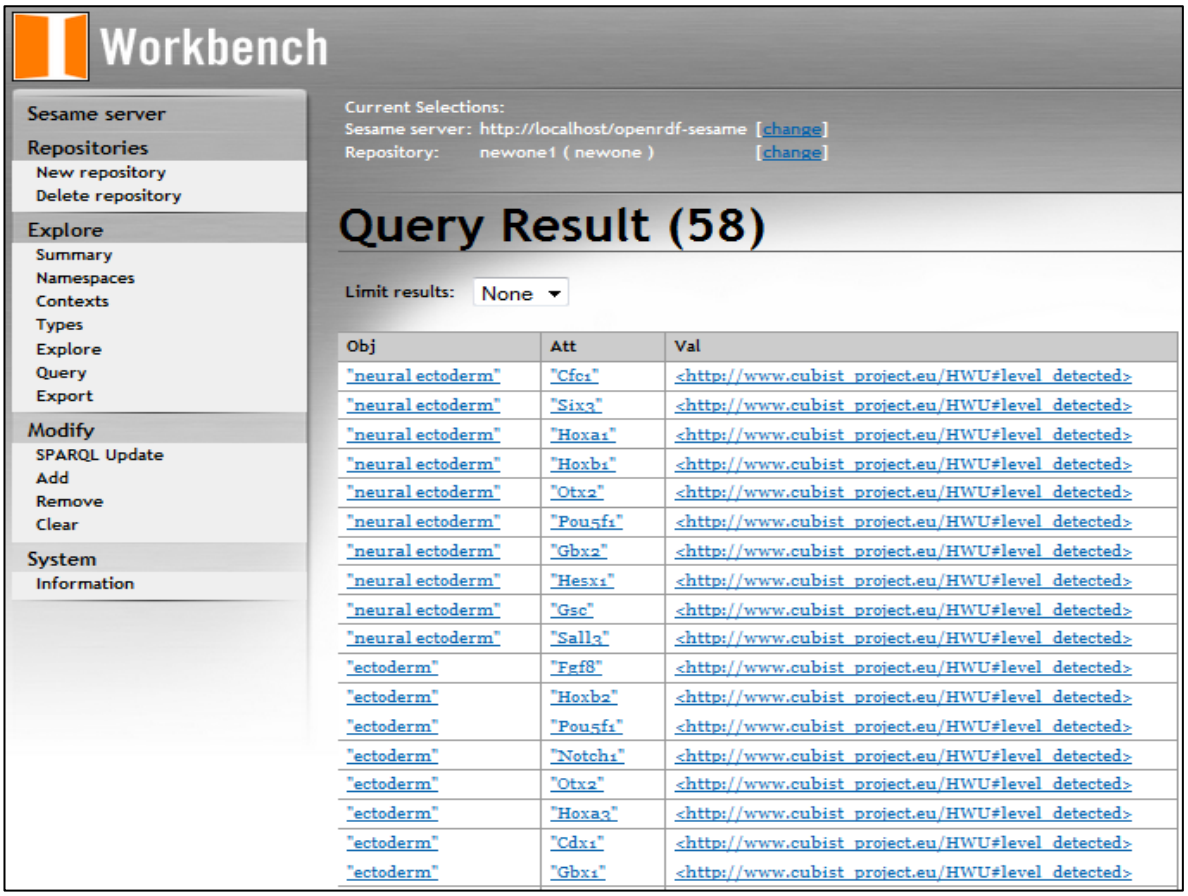

Figure 33: Example of a retrieved query result from Owlim-SE

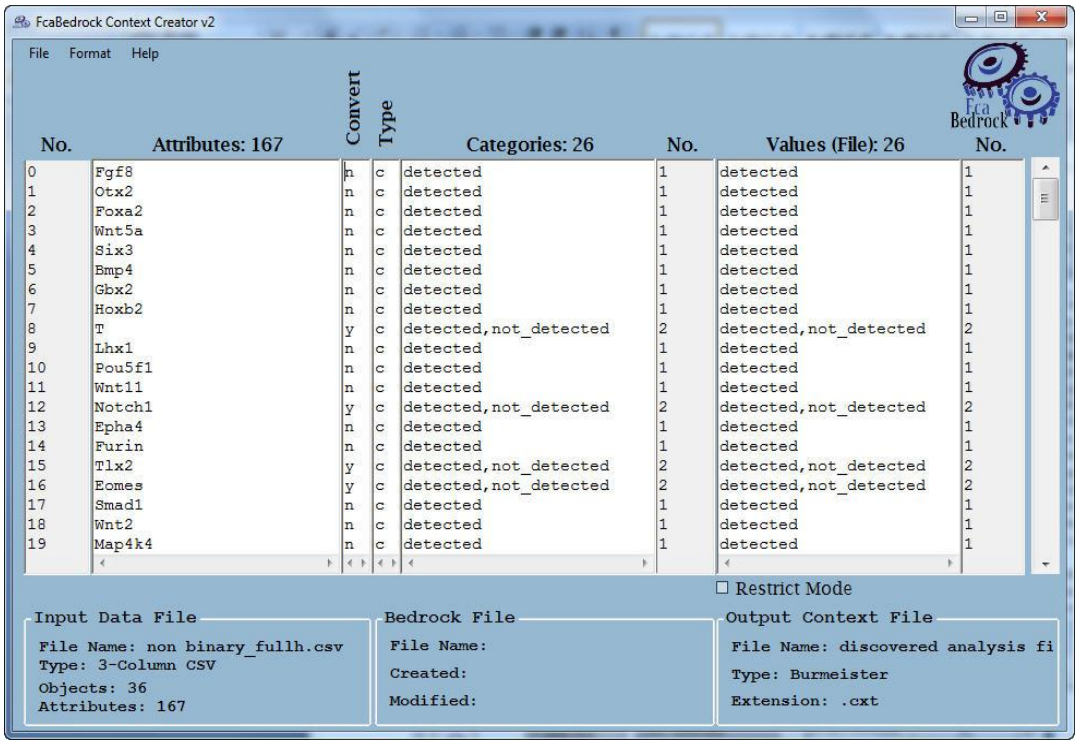

Figure 34: A semi-automated processing of EMAGE data in FcaBedrock

- Each of the stored CSV file is read by FcaBedrock and its single-valued attributes are manually restricted as exemplified in Figure 34 above

- The output file from the FcaBedrock is read and visualised through the use of the ConExp as exemplified in Figure 35 below

The concept lattice such as Figure 35 is consequently edited to separate out and exclusively visualise the IID. This is achieved by deselecting attributes which are not contradictory. Consequently, "*mesoderm*" is the only object with contradictory attribute (T-detected and T-not_detected). Also, the ConExp enables the visualisation of the incomplete data as evident in Figure 35. These incomplete data are depicted as a list of objects with no associated attribute at the topmost node of the concept lattice. This work will therefore focus on the visualisation of inconsistent data in subsequent concept lattices built by the semi-automated FcaBedrock approach.

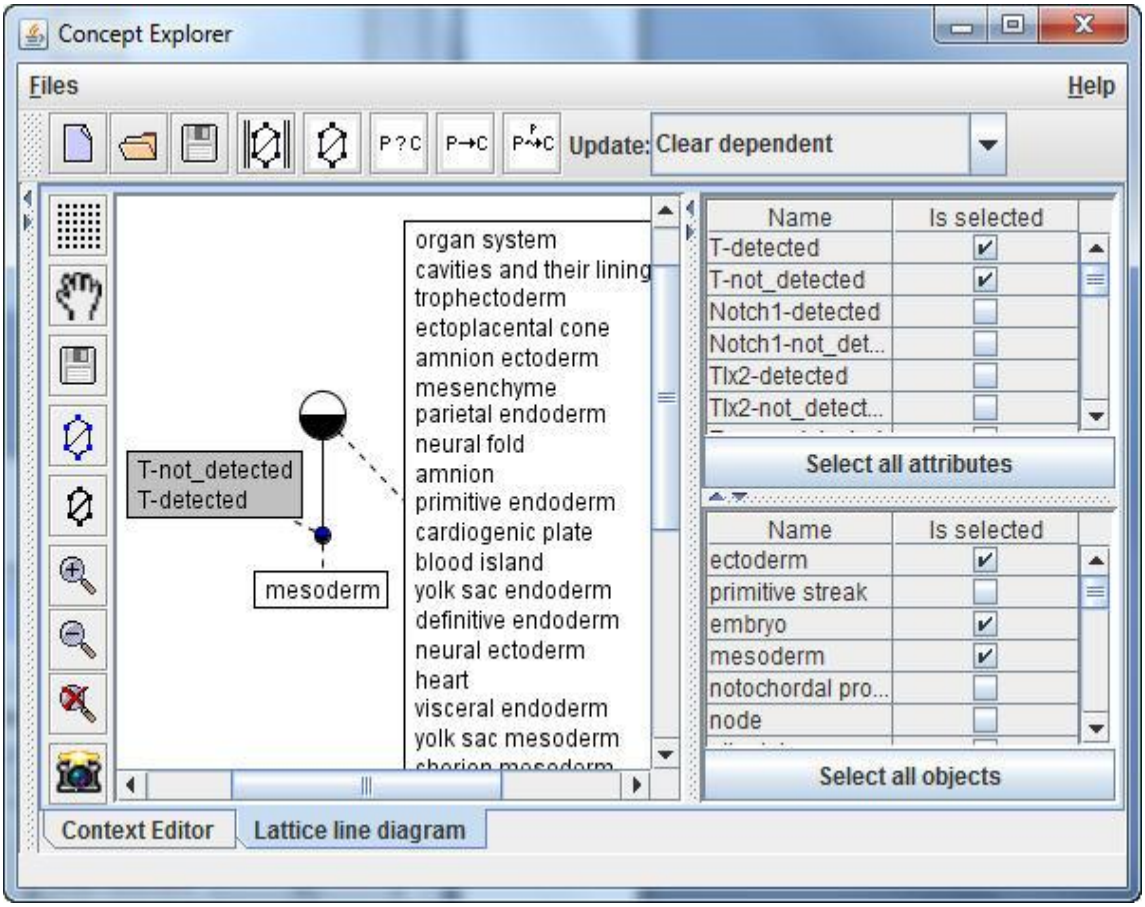

Figure 35: Visualising the output file of the FcaBedrock application in ConExp

8.4.2 Application

The applications used in this approach are the Owlim-SE, the FcaBedrock and the ConExp.

8.4.3 Queries and experimental results

Identifying and visualising the binary IID in the non-propagated EMAGE RDF data set

The SPARQL query in Table 14 is used in retrieving objects with binary gene expression from the non-propagated EMAGE RDF data store. The query retrieved tissues, associated genes and their corresponding binary expressions existing in Theiler Stage 11. Figure 36 depicts the corresponding concept lattice. Figure 36 depicts that the tissue *mesoderm* has binary inconsistency in the TS11 when the data set is not propagated.

Table 14: SPARQL query for retrieving Objects with binary gene expression from EMAGE dataset

SPARQL Query 1	Explanation
select distinct ?Obj ?Att ?Val{ ?x1 rdf:type :Textual_Annotation ; :in_tissue ?z; :has_involved_gene ?g ; :has_strength ?Val . ?z :has_theiler_stage :theiler_stage_11 . ?g rdfs:label ?Att . ?z rdfs:label ?Obj . Filter(?Val = :level_detected || ?Val = :level_not_detected) }	**variables to be returned by query** **Subgraph containing genes with expressions in tissues from Theiler Stage 11** **Limit the graph to "detected" or "not_detected" expressions**

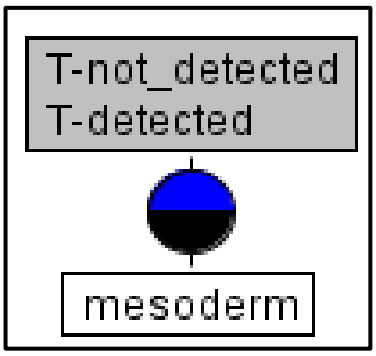

Figure 36: Binary inconsistency in non-propagated data set in TS 11

Identifying and visualising the analogue IID in the non-propagated EMAGE RDF data set

The SPARQL query in Table 15 (see below), is used in retrieving objects with analogue gene expression from the stored EMAGE data in the triple store. It retrieved non-propagated tissues, associated genes and their corresponding analogue expressions existing in Theiler Stage 11 of the stored EMAGE RDF data set. Figure 37 (see below), depicts the corresponding concept lattice. Figure 37 show that the tissue *neural ectoderm* has analogue inconsistency in Theiler Stage 11 when the data set is not propagated.

Table 15: SPARQL query for retrieving Objects with analogue gene expression from EMAGE dataset

SPARQL Query 1	Explanation
select distinct ?Obj ?Att ?Val {	**variables to be returned by query**
?x1 rdf:type :Textual_Annotation ; :in_tissue ?z; :has_involved_gene ?g ; :has_strength ?Val . ?z :has_theiler_stage :theiler_stage_11 . ?g rdfs:label ?Att . ?z rdfs:label ?Obj .	**Subgraph containing genes with expressions in tissues from Theiler Stage 11**
Filter(?Val = :level_strong || ?Val = :level_weak || ?Val = :level_moderate)	**Limit the graph to "detected" or "not_detected" expressions**

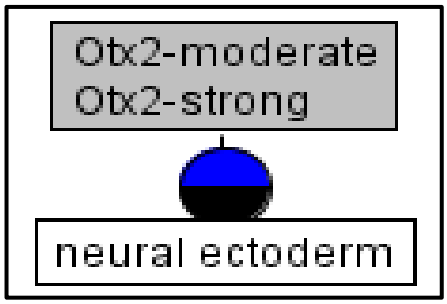

Figure 37: Analogue inconsistency in non-propagated data set in TS 11

Identifying and visualising the binary IID in EMAGE RDF data set where the gene expressions are negatively propagated

The EMAGE RDF data was propagated on the fly while retrieving tissues, associated genes and their corresponding gene expressions. Table 16 depicts the negative propagation of EMAGE RDF data and the request for the tissues, associated genes and their corresponding binary expressions in Theiler stage 11. Figure 38 (see below), depicts the corresponding concept lattice.

When gene expressions are negatively propagated, a gene which is not detected in a higher granularity tissue is assigned to related lower granularity tissues. In Figure 36 (see above), *mesoderm* is the only inconsistent tissue in the explored data set. When the same data set is negatively propagated, more inconsistent tissues are observed. Figure 38 shows all the binary inconsistent tissues where the same data set was negatively propagated. There are two additional inconsistent tissues in the data set. They include *neural ectoderm* (*Hoxb1*-detected and not detected) and *primitive streak* (*Eomes*- detected and not detected). Evidently, these tissues (*neural ectoderm* and *primitive streak*) are related lower granularity tissues to mesoderm (see Figure 9 above).

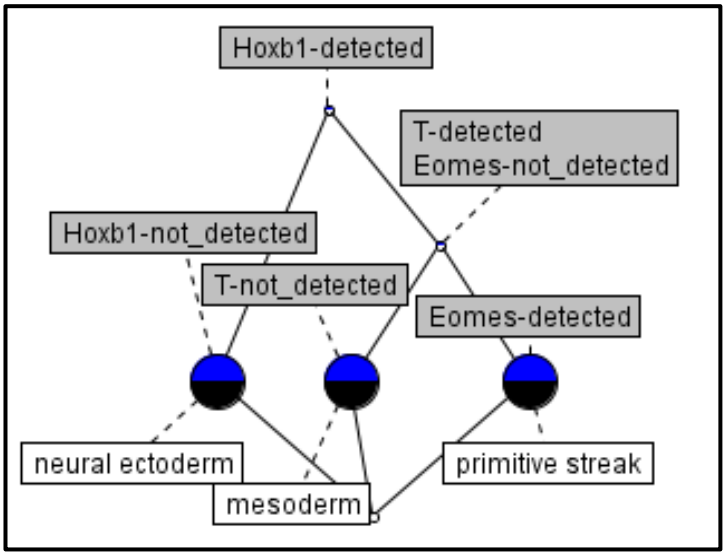

Figure 38: Binary inconsistency in negatively propagated data set in TS 11

Table 16: SPARQL query for negatively propagating and retrieving gene expressions from EMAGE dataset

SPARQL Query 2	Explanation
select distinct ?obj ?Att ?Val where { {	**variables to be returned by query**
?x rdf:type :Textual_Annotation ; :in_tissue ?z; :has_involved_gene ?g ; :has_strength :level_not_detected . ?z :has_theiler_stage :theiler_stage_11 . ?g rdfs:label ?Att. ?k :is_part_of ?z . {bind(?z as ?w).bind(?k as ?w)} ?w rdfs:label ?obj . ?x :has_strength ?Val .	**Subgraph for negatively propagating "level_not_detected" between related tissues ...pattern 1**
Filter(?Val = :level_detected || ?Val = :level_not_detected) }	**limit the subgraphs (pattern 1)to "detected" or "not_detected" expressions**
Union	**Keyword for Combining graph patterns**
{ ?x1 rdf:type :Textual_Annotation ; :in_tissue ?k; :has_involved_gene ?g ; :has_strength ?Val . ?k :has_theiler_stage :theiler_stage_11 . ?g rdfs:label ?Att. bind(?k as ?w) ?w rdfs:label ?obj. }	**Subgraphs showing gene expressions in non-propagated tissues ...pattern 2**
Filter(?Val = :level_detected || ?Val = :level_not_detected) }	**Limit the two subgraphs to "detected" or "not_detected" expressions**

Identifying and visualising the IID in EMAGE RDF data set where the gene expressions are positively propagated

Table 17 (see below), depicts the positive propagation of EMAGE RDF data and the request for the tissues, associated genes and their corresponding gene expressions existing in Theiler stage 11. Figure 39 (see below), depicts the corresponding concept lattice.

Table 17: SPARQL query for positively propagating and retrieving analogue expressions

SPARQL Query 3	Explanation
select distinct ?Obj ?Att ?Val { {	**variables to be returned by query**
?x rdf:type :Textual_Annotation ; :in_tissue ?k ; :has_involved_gene ?g ; :has_strength ?Val . ?z :has_theiler_stage :theiler_stage_11 . ?g rdfs:label ?Att. ?k :is_part_of ?z ; rdfs:label ?w . ?z rdfs:label ?e . {bind(?w as ?Obj). bind(?e as ?Obj)} }	**Subgraph for positively propagating gene expressions between related tissues ...pattern 1**
union	**Keyword for Combining graph patterns**
{ ?x1 rdf:type :Textual_Annotation ; :in_tissue ?z; :has_involved_gene ?g ; :has_strength ?Val . ?z :has_theiler_stage :theiler_stage_11 ; rdfs:label ?n . ?g rdfs:label ?Att . bind(?n as ?Obj) }	**Pattern of gene expressions in non-propagated tissues ...pattern 2**
Filter(?Val = :level_strong || ?Val = :level_weak || ?Val = :level_moderate) }	**Limit pattern 1 and 2 to "strong", "moderate" or "weak" expressions**

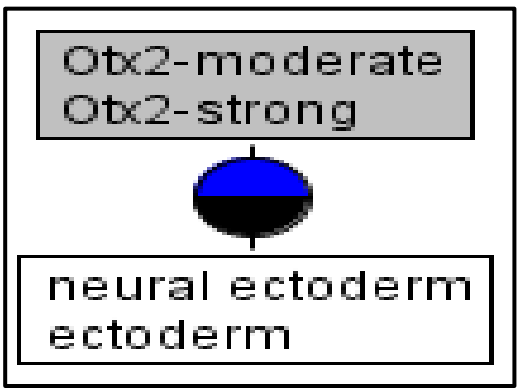

Figure 39: Analogue inconsistency in positively propagated data set in TS 11

In positive propagation, *detected* gene expression levels (strong, moderate, weak or possible) associated with a lower granularity tissue (child tissue) are assigned to related higher granularity tissues. Evidently, *neural ectoderm* is of a lower granularity to *ectoderm* (see Figure 9). In Figure 37, it is identified that *neural ectoderm* is

inconsistent (contains analogue gene expression). This inconsistency is assigned to *ectoderm* as a consequence of the positive propagation of gene expressions in the data set.

Measuring IID in the propagated EMAGE RDF data set

When dealing with IID, there are some situations when it is necessary to measure the level of inconsistency existing in an inconsistent data set (Grant and Hunter 2011). Such measures can help one to understand how sound or unsound the investigated data set is. A measure of inconsistency in the EMAGE can be achieved by counting the number of particular tissues concatenated with an associated gene whose expressions are inconsistent. The query in Table 18 (see below) implements such counts and Figure 40 (see below) presents the corresponding concept lattice. The concept lattice in Figure 40 is an example of a quantitative analysis in this work.

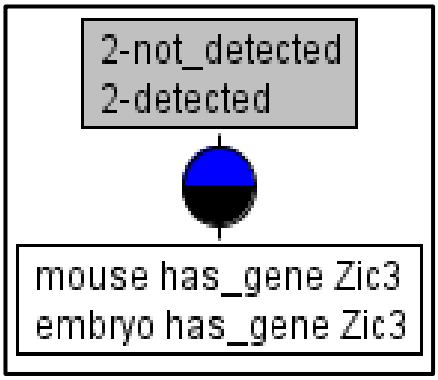

Figure 40: Concept lattice diagrams showing Amount of binary inconsistency of the negatively propagated data set in TS 08

Figure 40 shows that the gene expressions associated with *Zic3* was twice identified as binary inconsistent in the mouse and embryo. These expressions are identified in Theiler Stage 08 (see Table 18 below).

Table 18: SPARQL query for evaluating Objects with binary gene expression from EMAGE dataset

SPARQL Query 1	Explanation

select distinct (concat (?Obj, " has_gene ", ?Att) as ?Obj2) (count(?Val) as ?no_of_experiment) ((?Val) as ?strength) { {
 	
?x rdf:type :Textual_Annotation ; :in_tissue ?k ; :has_involved_gene ?g ; :has_strength ?Val .
?z :has_theiler_stage :theiler_stage_08 .
?g rdfs:label ?Att.
?k :is_part_of ?z ; rdfs:label ?w . ?z rdfs:label ?e .
{bind(?w as ?Obj). bind(?e as ?Obj)} }
 	
union
{
?x1 rdf:type :Textual_Annotation ; :in_tissue ?z; :has_involved_gene ?g ; :has_strength ?Val .
?z :has_theiler_stage :theiler_stage_08 ; rdfs:label ?n . ?g rdfs:label ?Att .
bind(?n as ?Obj) }
 	
Filter(?Val = :level_detected || ?Val = :level_not_detected)
 	
}
group by ?Obj ?Att ?Val
	variables to be returned by query
	
	Subgraph for negatively propagating "level_not_detected" between related tissues ...pattern 1
	
	Keyword for Combining graph patterns
	
	Subgraphs showing gene expressions in non-propagated tissues ...pattern 2
	
	Limit the two subgraphs to "detected" or "not_detected" expressions
	
	enable the counting of variables

8.4.4 Summary

The use of the semi-automated FcaBedrock approach provides a means to exclusively visualise and identify IID existing in a record set. Nevertheless, the approach does not always permit an exclusive visualisation of IID in its explored set of data, the association of *mesoderm* with *Eomes-not_detected* is not inconsistent as evident in Figure 38 (see above). A comprehensive evaluation of this approach and its comparison to other FCA approaches is presented in chapter 9.

8.5 Automated FcaBedrock approach

8.5.1 Introduction

The automated FcaBedrock approach is explained in section 7.4.2. Its approach follows the same steps as outlined in section 8.4 with the exception that it is

automatically processed. A summary of how the automated FcaBedrock approach is applied on each record set retrieved from the EMAGE RDF data set are as follows:

- Each stored CSV file is read by the extended FcaBedrock while selecting the inconsistency mode dialogue box as exemplified in Figure 41 below

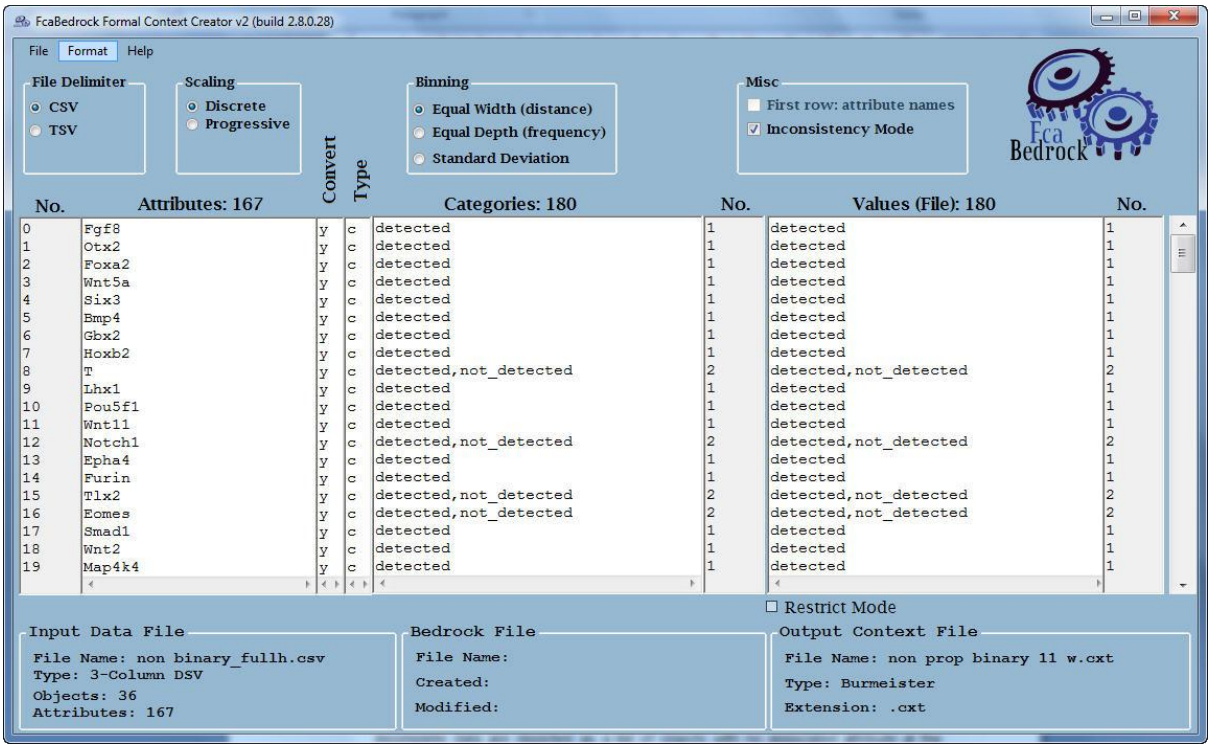

Figure 41: Automated processing of EMAGE data in FcaBedrock

- The output file from the extended FcaBedrock is read and transformed to a context file which is subsequently visualised through the ConExp. This is exemplified in Figure 42 below

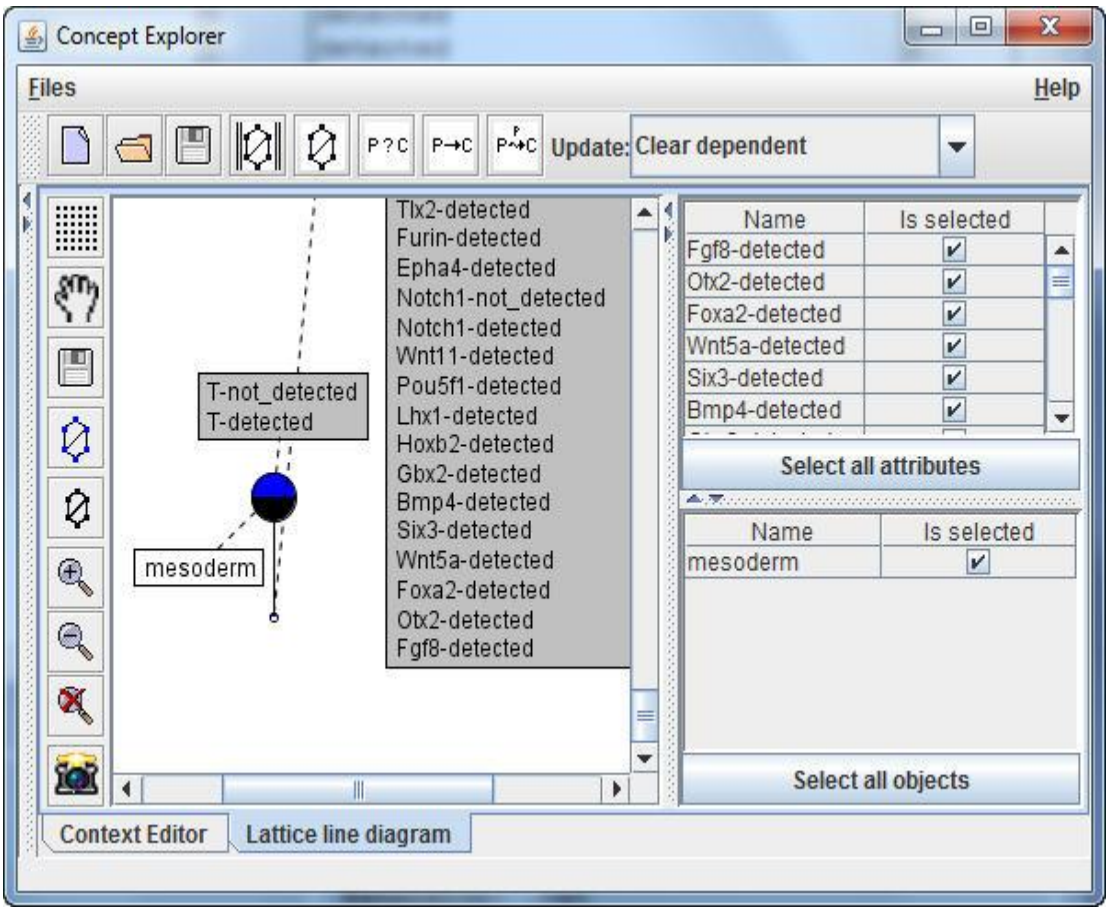

Figure 42: Visualising the output file of the extended FcaBedrock application in ConExp

8.5.2 Application

The applications used in this approach are the Owlim-SE, an extended FcaBedrock and the ConExp

8.5.3 Queries and experimental results

This approach was used to investigate the same data set as the semi-automated FcaBedrock approach. It also used the same queries and the results from the two approaches are almost the same with the exceptions of how their incomplete data are displayed and the display of the binary inconsistency in a negatively propagated data set.

Unlike the semi-automated FcaBedrock approach, the incomplete data of the automated FcaBedrock approach are depicted at the bottom node of the concept lattice as a list of attributes with no associated object (see Figure 42 above). The concept lattice showing binary inconsistency in negatively propagated data set (see Figure 43 below) is slightly different from Figure 38. Figure 43 shows only the binary inconsistent

tissues which include *neural ectoderm, primitive streak*, and *mesoderm*. This is different from Figure 38, which did not exclusively display the inconsistent data in the concept lattice.

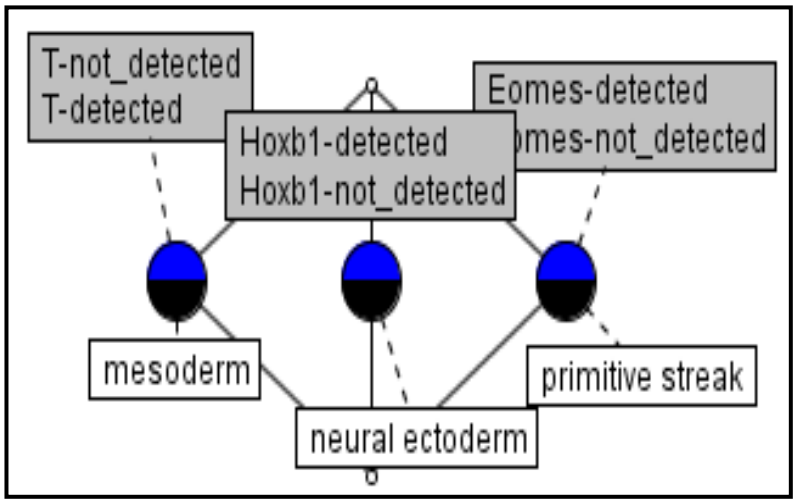

Figure 43: Binary inconsistency in negatively propagated data set

8.5.4 Summary

The automation process in the automated FcaBedrock approach is a distinctive feature of the approach. It has been shown that the automated FcaBedrock approach automatically separates out and exclusively visualises the IID in a noisy record set.

Chapter 9: Evaluation of IID Approaches

9.1 Introduction

This chapter evaluates the different FCA approaches for dealing with IID in EMAGE RDF data set. These approaches include Dau, CUBIST, Semi-automated FcaBedrock and Automated FcaBedrock approaches as explained in chapter 8. The DESMET methodology was used in evaluating the approaches with the aim of determining the most effective FCA approach to identify and visualise the IID in an RDF data set.

It is noted that there are many methodologies and tools that can be used in evaluating software tools and methods. Works such as (Grau et al. 2004, Kitchenham 1996; Kichenham et al. 1997, Hlupic and Mann 1995; Mohamed et al. 2004), describe such methods. Jadhav and Sonar (2009) provide a systematic review of papers published in this domain of study. The review identify among other things, that there is a lack of a common list of generic software evaluation criteria and associated meaning. This work resorts to evaluating its IID approaches through the guidelines of the DESMET methodology.

The DESMET methodology can be used by an evaluator or an academic researcher developing or investigating a new method (Kichenham et al. 1997). It is aimed at evaluating specific methods/tools in specific circumstances (Kichenham et al. 1997).The DESMET methodology (Kitchenham 1996; Kichenham et al. 1997) is used in this work to evaluate the various FCA approaches presented in chapter 8. Nine evaluation methods which include formal experiments, quantitative case studies, and qualitative screening are identified in (Kitchenham 1996; Kichenham et al. 1997). In this work, feature analysis based on notable features identified in IID processing applications such as in (Fan et al. 2008; Fazzinga et al., 2006; Raman and Hellerstein 2001) are used. Feature analysis is also known as qualitative screening (Kitchenham 1996; Kichenham et al. 1997).

The evaluation assessed the extent to which the examined FCA approaches provide features that can enable the identification and visualisation of the IID in an RDF data set. Section 9.2 identifies the essential features for approaches that enable the visual identification of IID. Section 9.3 explores the evaluation features in each of the applied IID approach. A grade is assigned to each of the FCA approaches in section 9.3. The key messages and finding of this chapter are presented in section 9.4.

9.2 Essential Features for Approaches that Enable the Visual Identification of IID

The research question, "*How can FCA tools and techniques be used to identify and visualise IID in RDF data?"* was considered in choosing the essential features that should exist in the approaches that deal with IID in RDF data set. An application that is designed to enable the visual identification of IID should have the capability to identify IID, automate the processing of IID, enable the exclusive visualisation of the identified IID and should also be applicable to different data sets. These attributes are further expatiated upon below.

This work also considers that an FCA application for dealing with IID should have some level of automation. This will enable such approach to be applied in a large RDF data set. Such FCA application will have an automatically processing system. This study considers the level of automation, as one of the factors in evaluating FCA applications that is designed to identify and visualise IID.

An exclusive visualisation of IID is another essential attribute of an FCA application that can effectively process IID. Given that there is an exponential growth of formal concepts and crossing edges in a concept lattice, relative to the context file (Dau 2013b), there is therefore the necessity to exclusively visualise IID if it is to be completely identified from a large and noisy data set. For example, it will be incomprehensible to build a concept lattice that visualises all the 100,000 formal concepts of a data set when dealing with 5 records whose data are inconsistent. IID in such lattice will be difficult to identify. FCA applications that process IID should be able to exclusively visualise the IID to identify the IID.

In addition, FCA applications used in the processing of IID should be capable of dealing with different data sets from different contexts. Consequently, the approach/tool should have some generalizability. This is necessary to eliminate the threats of external credibility. External credibility pertains to the conformability and transferability of findings and conclusions (Onwuegbuzie and Leech 2007).

9.3 An Assessment of FCA Approaches used in Dealing with IID in RDF Data

9.3.2 An Assessment of Dau's Approach

Dau's approach is documented in (Dau 2013a) and also explained in section 8.2. It is an automated approach and can be used to identify and visualise IID. Dau's approach can be applied on any other record set retrieved from a triple store when visually identifying IID in an RDF data set. This implies that it is generalizable. It also provides

an automated system of processing a record set to a formal context file. For instance, through the SPARQL2Context creator, queries can be issued to retrieve a record set which is saved as a context file through a few clicks of some buttons.

However, the concept lattice built from the context file as produced by Dau's tool (SPARQL2Context creator) is bulky when dealing with IID in a large data set. This is because the SPARQL2Context creator unlike the FcaBedrock, does not have a means to restrict attribute-values in the investigated data set. As a result, the consistent data and the IID are all converted to a context file. Such context file from a large data set usually results to a bulky and unreadable concept lattice. Dau's approach do not exclusively visualise IID in its investigated data set. Figure 29 and 30 depicts a bulky concept lattice produced from Dau's approach. Table 19 (see below), summarizes the capability of Dau's approach in dealing with IID.

9.3.1 An assessment of CUBIST approaches

CUBIST can identify inconsistent data as explained in section 8.3. It can identify incomplete data through representing such objects at the topmost node of the lattice or through the representation of attributes without an associated object at the bottom of the lattice. It also uses distinct colour to represent particular type of inconsistency thereby enabling the identification of the type of IID.

However, some techniques used in CUBIST cannot be applied to other data set without some corresponding scaling. Examples of such techniques as discussed in section 8.3 include the automatic fixing of incomplete data through the flip-fops technique and the use of distinct colour to represent particular type of inconsistency. These techniques can be used on another data set but with some corresponding scaling of the data. This may also require customising the CUBIST application to suit the investigated data set. Melo et al. (2013) assert that "*Although most of the functionalities in CUBIST can be used with other data than EMAGE (with the corresponding scaling of data), as future work we will extend our experiments to other genes expression data sets like cancer and brain development*."

CUBIST can be used to exclusively visualise all the IID in an investigated data set but this will involve the CUBIST user to manually select each dialogue box associated with contradictory attributes (see Figure 32 above), before visualising the associated concept lattice. Such an approach will be tedious when there are many rows retrieved from EMAGE data set. The co-occurrence tab in CUBIST can be used to identify and visualise the IID in EMAGE data set by exploring for the presence of objects in selected contradictory attributes pair. But the use of the co-occurrence tab

will only explore all the objects associated with the pair of contradictory attributes without visualising other associated inconsistencies or incompleteness. Table 19 (see below), provides a summary of CUBIST capability to deal with IID.

9.3.3 An Assessment of Semi-automated FcaBedrock Approach

The semi-automated FcaBedrock approach is explained in section 8.4. It is used to identify and visualise IID in a data set whose attribute-values are mutually exclusive. The semi-automated FcaBedrock approach can enable its users to exclusively visualise all the IID in objects associated with mutually exclusive attribute-values, where the contradictory attribute-values are distinctively associated to their object(s) and where the data set is stored in CSV format.

However, for data sets which have more complex associations such as where there are contradictory attribute-values which are not distinctively associated to particular objects, the semi-automated FcaBedrock approach will not exclusively depict the IID existing in such data sets. In Figure 38 for example, the attribute-value *Eomes-not_detected* is not distinctively associated with *primitive streak*. It is associated with both *mesoderm* and *primitive streak*. Its association with primitive streak is contradictory but its association with *mesoderm* is not contradictory because *mesoderm* is not associated with *Eomes-detected*.

The semi-automated FcaBedrock approach is not an automated FCA approach. It is also a difficult and painstaking approach when it is used on a large data set. In addition, there are IIDs which do not exist in the realm of mutually exclusive attribute-values e.g. typographical errors. This approach will not be able to identify such inconsistencies. Table 19 (see below), summarizes the capability of the semi-automated FcaBedrock approach in dealing with IID

9.3.2 An Assessment of Automated FcaBedrock Approach

The automated FcaBedrock approach is explained in section 8.5 of this work. It can be used to identify and visualise IID in a data set whose attribute-values are mutually exclusive and where the data is stored in CSV format. It can enable its users to exclusively visualise all the IID in a record set whose attribute-values are mutually exclusive. It is an automated process and it presents concept lattices which are very readable when compared to other FCA approaches understudied in Chapter 8.

Similar to the semi-automated FcaBedrock approach, the automated FcaBedrock approach will not be able to identify IIDs that are not associated with mutually exclusive

attribute-values. Table 19 (see below), summarizes the capability of the automated FcaBedrock approach in dealing with IID

Table 19: A summary of attributes in IID processing tool/approaches

s/no	Automated FcaBedrock approach 1st	CUBIST approaches 2nd	Semi-automated FcaBedrock approach 3rd	Dau's approach 4th
Identification of IID	Yes	Yes	Yes	Yes
Level of automation for identifying of IID	automated	the co-occurrence tab is automated while the use of filter options are Semi-automated	semi-automated	automated
Exclusive view	Yes	To a certain extent	To a certain extent	No
Generalizability of approach	Generalizable	approaches are customized for CUBIST case study	Generalizable	Generalizable

9.4 Key Messages and Findings

Dau's approach does not enable its users to exclusively visualise IID existing in their investigated data set. Although it is generalizable, it provides almost the basic processes in the classical FCA approach (see chapter 4). Consequently, it produces bulky and unreadable concept lattices when used on a large data set.

Dau and other FCA approaches such as the fault tolerance approach were refined and integrated in CUBIST application. This provided CUBIST with robust techniques to deal with IID in EMAGE data set. The CUBIST application provides good visualisation of its explored data and visualisation options such as charts and graphs but most of its approaches are not FCA based nor are they designed to exclusively identify and visualise IID in a data set. Also some of its features or approaches which can be used to identify or visualise IID in EMAGE such as the use of distinct colour to represent particular type of inconsistency are not generalizable without some corresponding scaling.

This work assessed Dau's approach as the least appropriate of the four IID approaches understudied. It assessed CUBIST approaches and the semi-automated

approach as the 2rd and 3nd most suitable IID approaches, respectively. The automated FcaBedrock approach is assessed as the best IID processing approach. The factors identified in this work (see Table 19) are the basis by which these FCA approaches are evaluated. These understudied FCA tools/approaches may be assessed by other researchers differently but given the examined factors, IID in RDF data set is best processed by the automated FcaBedrock approach. The automated FcaBedrock approach provides all the essential features needed to identify and visualise IID. It exclusively identifies and visualises the IID existing in objects associated with many-value attribute in an RDF data set. Its approach can be used in data sets from other contexts. It is also an automated FCA approach. More examples of results from the automated FcaBedrock approach are depicted in appendix E.

Chapter 10: Conclusion and Future work

This study sets out to explore how IID in an RDF data set can be dealt with through the use of FCA and Owlim SE. IID presents some challenges in data analysis as explained in this work. Such challenges include loss of information, inaccurate analysis and cost of correcting the analysis. These challenges are evident in both the traditional and semantic databases. This study identified the CWA and OWA as the basic principles underpinning the existence of IID in traditional and semantic databases respectively. In semantic databases, inconsistent data are allowed and the information in such databases is never assumed to be complete. This is unlike in the traditional databases where inconsistent data can be removed or replaced and where there is an assumption of completeness in the information stored in the database. The factors that contribute to the presence of IID in both the traditional and semantic databases are explained in chapters 2 and 3.

Actually, removing an inconsistent data from a database will only increase the incompleteness of the database thereby introducing inaccuracy in the analysis of the data from the database. For example, if 100 people voted in an election and 10 votes were inconsistent such that the voters ticked more than one candidate as preferred candidate or filled in wrong personal details, deleting the bad votes may lead to incomplete number of votes and inaccurate analysis. Consequently, analysis such as 60% of the voters voted candidate ‘A’ and 40% of the voters voted for candidate ‘B’ will be wrong when the bad votes are excluded. Such an assertion is misleading and incomplete. It is demonstrated in this study that IID should be identified, evaluated, analysed, and even reasoned with, as to provide a sound analysis to the data set user.

IID may constitute a small percentage of an entire data set and identifying such a small proportion in a large data set may be difficult. It is shown in this work how FCA tools and techniques were used to identify and visualise the IID existing in EMAGE RDF data set. This is to answer the research question “How can FCA tools and techniques be used to identify and visualise IID in RDF data?” This work explored the capabilities of association rule, fault tolerance, attribute exploration, CUBIST, Dau, semi-automated FcaBedrock and automated FcaBedrock approaches in dealing with IID in RDF data set. It noted that not all its identified FCA approaches can enable the identification and visualisation of the IID in RDF data. Consequently, the study evaluated Dau, CUBIST, semi-automated FcaBedrock and automated FcaBedrock approaches.

This chapter provides a review of all the chapters in this thesis in section 10.1. The main results of this work with relation to the research objectives are presented in section 10.2. Section 10.3 briefly explains the contributions of this work to knowledge. The challenges encountered in this work are outlined in section 10.4. This chapter is concluded by outlining the anticipated future work in section 10.5.

10.1 A Review of the Various Chapters in this Thesis

Chapter 1 presented the rationale and objectives of the work.

Chapter 2 explained the meaning, the types, the causes, the sources and the prevention of IID in the traditional databases where the CWA is adopted. It noted that IID is an important concept in data processing and that there is great need to properly analysis it in order to avoid inaccurate data analysis.

An overview of ST, semantic databases and RDF are presented in Chapter 3. The chapter also identified and classified existing approaches to dealing with IID in a semantic database notably the rule based, the query based and the combination of FCA techniques with query based approaches. It noted that these approaches do not allow its users to exclusively visualise IID in an RDF database. It explained that the essence of exclusively visualising IID is to ensure a holistic identification of IID when dealing with a large and noisy data set.

Chapter 4 described the classical FCA approach. It explained how formal concepts are derived from formal context and how they are displayed in a lattice structure. Also, it is explained that the classical FCA approach does not enable an exclusive identification and visualisation of the IID when dealing with a large and noisy data set.

Chapter 5 explained the research method used to developing the automated and semi-automated FcaBedrock approaches. It also explained the case study with emphasis on the single case study and the research method used in validating the results obtained in this work. The research ethics and the challenges encountered in the course of this work were also outlined.

Chapter 6 explained the EMAGE which is the case studied in this work. It explained how the e-Mouse Atlas Project (EMAP) is used in EMAGE. In addition, it described the sources of IID in the EMAGE and the EMAGE RDF data set.

Chapter 7 described how the RDF query language (SPARQL) can be used to retrieve IID from a semantic database. It provided comprehensive details about existing and new FCA approaches used in dealing with the IID in a data set.

Chapter 8 explained how Dau, the CUBIST, the semi-automated FcaBedrock and the automated FcaBedrock approaches were used to identify and visualise IID existing in the EMAGE RDF data set.

Chapter 9 compared and evaluated the effectiveness of the FCA approaches which were used in investigating the EMAGE RDF data set with respect to their capability to identify and visualise IID in an RDF data set.

This chapter (chapter 10) provides a brief review of all the chapters in this work. It explains the main contributions of this work to knowledge, the challenges encountered and the future work.

10.2 Main Results of the Research by Research Objectives

This work accomplished the objectives listed in section 1.3 of this thesis. This section explores where and how these objectives were addressed in this work as follows:

1. *To understand IID issues and how they are dealt with in a traditional technology setting (Obj.1)*

This work investigated the traditional and semantic database literature as to understand IID issues and how they are dealt with in a traditional technology setting. Chapter 2 identified the CWA as the underpinning principle of IID in traditional databases. It also identified the null, integrity constraints and the use of optional fields as the major sources of IID in traditional databases. Binary and analogue inconsistencies were identified as types of IID in traditional and semantic databases. It was also outlined in the chapter, several approaches through which IID can be dealt with in traditional databases. Such approaches include resolving/repairing IID, preventing IID, and reasoning with IID.

2. *To understand IID issues in ST setting and also investigate existing approaches used in dealing with IID in a ST setting (Obj. 2 and 3)*

The semantic database literatures were investigated to understand IID issues in ST and the available approaches that are used in dealing with the anomaly. Chapter 3 described how IID in RDF data are processed in a ST setting. It explained how the OWA principles underpin the existence of IID in RDF triple store. RDF data, entailment rules and the different ways by which IID in RDF triple stores are dealt with in ST settings were also explained. These approaches include the rule based approach (entailment), the query based approach, and the combination of query based approach with FCA techniques. The combination of FCA with query based approaches was

further investigated in chapter 7, 8 and 9. The lapses of these approaches were also identified.

3. *To propose FCA as an appropriate and effective technique for dealing with IID in ST setting (Obj. 4)*

The works of Andrews and McLeod (2011), Dau (2013a and 2013b) and Melo et al. (2013), were great sources of information in this study. These works demonstrated how FCA can be used in dealing with IID. The use of FCA to deal with IID provides a comprehensive means of addressing the issues in IID as documented in chapters 4, 7, 8 and 9. Also, the new FCA approaches used in dealing with IID are identified, explained and used in this work as documented in chapters 7, 8 and 9.

4. *To build on existing FCA approaches and develop better novel approaches (Obj. 5)*

The approaches proposed in (Andrews and McLeod 2011; Dau 2013a; Melo et al. 2013) involve the use of the FcaBedrock or a similar tool such as the SPARQL2Context creator (Dau 2013a), to convert an investigated data to a formal context file which is subsequently visualised as a concept lattice. This work built on these existing FCA approaches by creatively using the FcaBedrock tool as evident in the semi-automated FcaBedrock approach. It also modified the FcaBedrock tool to exclusively identify IID in an RDF data set as evident in the automated FcaBedrock approach. These approaches are presented in chapter 7, implemented in chapter 8 and evaluated in chapter 9. Chapter 9 depicted that the best performing FCA approach for processing IID in an RDF data set is the automated FcaBedrock approach.

5. *To apply existing and new FCA approaches to an indicative case study (Obj. 6)*

This study explained in chapters 8 how the existing and new FCA approaches were applied on the EMAGE RDF database. The use of the EMAGE as a use case in a case study research is described in chapters 5. Also, a detailed description of EMAGE is provided in chapter 6.

6. *To compare and evaluate the usefulness and effectiveness of the different approaches (Obj. 7)*

A comparison of the identified FCA approaches that were used in this work to deal with IID in EMAGE RDF data set was presented in Chapter 9. It was also shown in the chapter how the new and existing FCA approaches were assessed and the result of the assessment.

10.3 Contributions of the Research to Knowledge

The key contributions of this work to knowledge are as follows:

1. *This work has increased the available documentations on dealing with IID in semantic databases.*

   One of the rationales for doing this research as explained in chapter 1 is to increase the documentation on semantic database. It is shown in chapter 2 that there are ample documentations on how IID can be dealt with in traditional databases. This is unlike in semantic databases as evident in chapter 3. In this work, the various ways by which IID can be dealt with in a semantic database and the use of FCA approaches in dealing with IID are explained. This thesis and publications derived from this work such as (Nwagwu and Orphanides 2015; Nwagwu 2014; Nwagwu 2013), have increased the amount of documentations on semantic databases, particularly the documentations on dealing with IID existing in an RDF data set through the use of FCA approaches.

2. *This work identified that existing FCA approaches do not enable the exclusive visualisation of the IID existing in its investigated data set.*

   This is the first academic work to identify that the existing FCA approaches do not exclusively visualise the IID in their investigated data set. An unreadable concept lattice will be produced when a large data set is holistically visualised. A better approach to identifying IID in a large data set is to separate out and exclusively visualise the IID as a concept lattice. This will enable an easier identification and better visualisation of the IID in the data set. Unlike the new FCA approaches, it was identified that the existing approaches used in dealing with IID do not exclusively visualise the IID in RDF data set (see chapters 3, 5, and 7).

3. *This study evaluated the different FCA approaches used in dealing with IID in RDF data set.*

   The study presents comprehensive details about the different FCA approaches used in dealing with IID in an RDF data set. Each of the identified FCA approach that can be used in visually identifying IID in an RDF data set was applied on the EMAGE data set as presented in chapter 8 of this work. This made it possible to evaluate their effectiveness in identifying and visualising IID in an RDF data set (see chapter 9). This evaluation will provide a direction to researchers coming into this area of study. It will also enable them to understand what has been done and what is yet to be done.

4. *This work compared the different FCA approaches used in dealing with IID in RDF data set.*

   The FCA approaches which were used to identify IID in EMAGE RDF data set (see chapter 8) were also compared as documented in chapter 9. This was done by comparing their effectiveness and efficiencies in identifying and visualising the IID in the EMAGE RDF data set. Such comparisons will provide a direction to data analysts that will adopt any of the FCA approaches. It will enable the data analyst to understand the limitation of his chosen approach.

10.4 Challenges of this Work

There are undoubtedly challenges to every empirical research and this research was not an exception. This study developed novel approaches which include semi-automated and automated FcaBedrock approaches. These novel approaches only consider the inconsistencies in objects associated to mutually exclusive attribute values. It should be noted that inconsistency in a data set does not only result from contradictory values in mutually exclusive attribute values as considered in these new approaches developed. The new approaches implemented in this work do not have the capability to identify inconsistencies such as typographical errors. There is need to consider other novel approaches that can be used to identify other types of IID that can exist in an RDF data set.

Also, Dau and CUBIST approaches were theoretically evaluated unlike the automated and semi-automated FcaBedrock approaches. This is because the licences for Dau and CUBIST tools could not be obtained for this work. This challenge was overcome by theoretically assessing these tools and approaches based on their associated publications. Although the evaluated publications were based on the analysis of the same data set, the analyses in such publications were not done in this work. Consequently, the validity of the results from such publications will depend on the accuracy of the explanations in the articles.

10.5 Future Work

Aspects of this study have been described in (Nwagwu and Orphanides 2015; Nwagwu 2014; Nwagwu 2013), and further publications are planned. This further work includes applying the identified approaches to other real life cases. One plan is to examine the

DBpedia knowledge base[26]. The DBpedia knowledge base is a good example of an OWA database. IID will be evident in this database because it integrates data from different sources. It will be interesting to explore how the IID in the database is dealt with and how the process can be improved. Also, FCA approaches for identifying, reasoning with and evaluating the IID in a database can be applied to the DBpedia knowledge base.

Another plan is to investigate how FCA can be used to identify IID in a spatial database. That avenue would explore how IID can be identified in data from remote sensing and the spatial databases in Geographical Information Systems (GIS) , which is absent from the EMAGE case study. The association rule approach for identifying IID described in chapter 7, section 7.4.1 can be used in identifying and visualising IID in the spatial data. For example, a picture of a geographical location can be stored as master data set in a spatial database. Subsequent pictures of the same geographical location can be analysed as a sub unit of the master data set to identify distortions, inconsistencies or incompleteness through the use of the association rule approach, as explained in section 7.4.1. Issues of inaccuracy and incompleteness exist in GIS data as identified in Devillers and Jeansoulin (2006 p. 17- 29, 184-207). The use of association rule to deal with IID was not fully explored in this work which provides an opportunity for a future work on this aspect of using FCA to deal with IID.

The automated FcaBedRock approach applies mutually exclusive attribute principle when dealing with IID in a noisy data set. It can convert data in CSV format to formal context before visualising the formal context with a context visualisation tool. It can be improved by integrating the FcaBedRock tool with a context visualisation tool and other methods of dealing with IID such as the association rule. This measure will enrich the ability of the approach to deal with IID existing in an RDF data set and can be explored further in the future.

Conclusively, this work has demonstrated that IID in RDF data set should be identified and visualised as to avoid inaccurate analysis. It has been shown that FCA tools and techniques have the capabilities to effectively and efficiently identify, reason with, evaluate, and visualise the IID existing in an RDF data set. Data scientists involved with the analysis of noisy and large data sets are therefore advised to read, employ, develop and dissimilate the various approaches presented in this work.

[26] http://wiki.dbpedia.org/Datasets2014

References

Abele, L., Legat, C., Grimm, S., and Muller, A. W. (2013, July). Ontology-based validation of plant models. In *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on* (pp. 236-241). IEEE.

Andrews, S. (2011). In-close2, a high performance formal concept miner. In Proceedings of the 19th international conference on Conceptual structures for discovering knowledge, ICCS'11, pages 50{62, Berlin, Heidelberg, Springer-Verlag.

Andrews, S., and McLeod, K. (2011). Gene co-expression in mouse embryo tissues. In F. Dau (Ed.), Proceedings of the 1st CUBIST (Combining and Uniting Business Intelligence with Semantic Technologies) Workshop 2011. (pp. 1-10). Dresden.

Andrews, S., and McLeod, K. (2013). Gene co-expression in mouse embryo tissues. *International Journal of Intelligent Information Technologies (IJIIT)*, *9*(4), 55-68.

Andrews, S., and Orphanides, C. (2010). Analysis of Large Data Sets using Formal Concept Lattices. In *CLA* (pp. 104-115).

Andrews, S., Orphanides, C., and Polovina, S. (2011). Visualising computational intelligence through converting data into formal concepts. In *Next generation data technologies for collective computational intelligence* (pp. 139-165). Springer Berlin Heidelberg.

Annoni, P., and Brüggemann, R. (2008). The dualistic approach of FCA: A further insight into Ontario Lake sediments. *Chemosphere*, *70*(11), 2025-2031.
Allemang, D., & Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier

Afrati, F., Li, C., & Pavlaki, V. (2008, June). Data exchange: Query answering for incomplete data sources. In *Proceedings of the 3rd international conference on Scalable information systems* (p. 6). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Arenas, M., Bertossi, L., & Chomicki, J. (1999, May). Consistent query answers in inconsistent databases. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 68-79). ACM.

Baader, F., Ganter, B., Sertkaya, B., & Sattler, U. (2007, January). Completing Description Logic Knowledge Bases Using Formal Concept Analysis. In *IJCAI* (Vol. 7, pp. 230-235).

Baldock, R. A., Bard, J. B., Burger, A., Burton, N., Christiansen, J., Feng, G., ... & Davidson, D. R. (2003). EMAP and EMAGE. *Neuroinformatics*, *1*(4), 309-325.

Becker, H. S. (1996). The epistemology of qualitative research. *Ethnography and human development: Context and meaning in social inquiry*, 53-71.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.

Bertossi, L., Bravo, L., Franconi, E., & Lopatenko, A. (2008). The complexity and approximation of fixing numerical attributes in databases under integrity constraints. *Information Systems*, *33*(4), 407-434.

Bizer Christian, Tom Heath, and Tim Berners-Lee (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *5*(3), 1-22. last accessed 11 October, 2013 at: http://eprints.soton.ac.uk/271285/1/bizer-heath-berners-lee-ijswis-linked-data.pdf

Bleiholder, J., and Naumann, F. (2008). Data fusion. ACM Computing Surveys (CSUR), 41(1), 1.

Bloomberg Trading Solutions**.** The value of trusted data. Online. Last accessed on 12th January, 2015. Available at http://www.bloomberg.com/trading-solutions/content/uploads/sites/3/2014/06/50245-Bloomberg-Trusted-Data-Data-Points-v6-2014_06_12.pdf

Bravo, L., & Bertossi, L. (2006). Semantically correct query answers in the presence of null values. In *Current Trends in Database Technology–EDBT 2006* (pp. 336-357). Springer Berlin Heidelberg.

Bonifati, A., Chrysanthis, P. K., Ouksel, A. M., & Sattler, K. U. (2008). Distributed databases and peer-to-peer databases: past and present. *ACM SIGMOD Record*, *37*(1), 5-11.

Boström, H., Andler, S. F., Brohede, M., Johansson, R., Karlsson, A., Van Laere, J., ... & Ziemke, T. (2007). On the definition of information fusion as a field of research.

Boyl, P. P., Signore, M., Acampora, D., Martinez-Barbera, J. P., Ilengo, C., Annino, A., ... & Simeone, A. (2001). Forebrain and midbrain development requires epiblast-restricted Otx2 translational control mediated by its 3′ UTR. *Development*, *128*(15), 2989-3000.

Burmeister, P., and Holzer, R. (2005). Treating incomplete knowledge in formal concept
analysis. In *Formal Concept Analysis* (pp. 114-126). Springer Berlin Heidelberg.

Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., & Rosati, R. (2001). Data integration in data warehousing. *International Journal of Cooperative Information Systems*, *10*(03), 237-271.

Calì, A., Calvanese, D., De Giacomo, G., & Lenzerini, M. (2013). Data integration under integrity constraints. In *Seminal Contributions to Information Systems Engineering* (pp. 335-352). Springer Berlin Heidelberg.

Carpineto, C., and Romano, G. (2004). Concept data analysis: Theory and applications. Wiley. com.

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record*, *26*(1), 65-74

Chomicki, J., Marcinkowski, J., & Staworko, S. (2004, November). Computing consistent query answers using conflict hypergraphs. In *Proceedings of the thirteenth*

ACM international conference on Information and knowledge management (pp. 417-426). ACM.

Christiansen, J. H., Yang, Y., Venkataraman, S., Richardson, L., Stevenson, P., Burton, N., ... & Davidson, D. R. (2006). EMAGE: a spatial database of gene expression patterns during Mouse embryo development. *Nucleic acids research*, *34*(suppl 1), D637-D641.

Codd, E. F. (1979). Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems (TODS)*, *4*(4), 397-434.

Codd, E. F. (1986). Missing information (applicable and inapplicable) in relational databases. *ACM Sigmod Record*, *15*(4), 53-53.

Cong, G., Fan, W., Geerts, F., Jia, X., & Ma, S. (2007, September). Improving data quality: Consistency and accuracy. In *Proceedings of the 33rd international conference on Very large data bases* (pp. 315-326). VLDB Endowment.

Cortés-Calabuig, A., Denecker, M., Arieli, O., Van Nuffelen, B., & Bruynooghe, M. (2005). On the local closed-world assumption of data-sources. In *Logic Programming and Nonmonotonic Reasoning* (pp. 145-157). Springer Berlin Heidelberg.

Dau, F. (2013a). Towards Scalingless Generation of Formal Contexts from an Ontology in a Triple Store. *International Journal of Conceptual Structures and Smart Applications* (IJCSSA), 1(1), 18-38.

Dau, F. (2013b). An Implementation for Fault Tolerance and Experimental Results. In *CUBIST Workshop* (pp. 21-30).

Decker, H., & Martinenghi, D. (2011). Inconsistency-tolerant integrity checking. *Knowledge and Data Engineering, IEEE Transactions on*, *23*(2), 218-234.

Denecker, M., Cortés-Calabuig, Á., Bruynooghes, M., & Arieli, O. (2010). Towards a logical reconstruction of a theory for locally closed databases. *ACM Transactions on Database Systems (TODS)*, *35*(3), 22.

Devillers, R., & Jeansoulin, R. (Eds.). (2006). *Fundamentals of spatial data quality*. London: ISTE.

Drumond, L., Rendle, S., and Schmidt-Thieme, L. (2012, March). Predicting RDF triples in incomplete knowledge bases with tensor factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 326-331). ACM.

DuCharme, B. (2011). Learning SPARQL: querying and updating with SPARQL 1.1.

Fan, W., Geerts, F., & Jia, X. (2008). Semandaq: a data quality system based on conditional functional dependencies. *Proceedings of the VLDB Endowment*, *1*(2), 1460-1463.

Fagin, R., Kolaitis, P. G., & Popa, L. (2005a). Data exchange: getting to the core. *ACM Transactions on Database Systems (TODS)*, *30*(1), 174-210.

Fagin, R., Kolaitis, P. G., Miller, R. J., & Popa, L. (2005b). Data exchange: semantics and query answering. *Theoretical Computer Science*, *336*(1), 89-124.

Fazzinga, B., Flesca, S., Furfaro, F., & Parisi, F. (2006). DART: a data acquisition and repairing tool. In *Current Trends in Database Technology–EDBT 2006* (pp. 297-317). Springer Berlin Heidelberg.

Finkelstein, A. (2000, January). A foolish consistency: Technical challenges in consistency management. In *Database and Expert Systems Applications* (pp. 1-5). Springer Berlin Heidelberg.

Flesca, S., Furfaro, F., & Parisi, F. (2010). Querying and repairing inconsistent numerical databases. *ACM Transactions on Database Systems (TODS)*, *35*(2), 14.

Fuxman, A., Fazli, E., & Miller, R. J. (2005, June). Conquer: Efficient management of inconsistent databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 155-166). ACM.

Fürber, C., and Hepp, M. (2010, January). Using SPARQL and SPIN for data quality management on the semantic web. In *Business Information Systems* (pp. 35-46). Springer Berlin Heidelberg.

Ganter, B. (1999). Attribute exploration with background knowledge. *Theoretical Computer Science*, *217*(2), 215-233.

Ganter, B., Stumme, G., and Wille, R. (2002). Formal concept analysis: Methods and applications in computer science. *TU Dresden, http://www. aifb. uni-karlsruhe. de/WBS/gst/FBA03. shtml*.

Ganter, B. (2010). *Two basic algorithms in concept analysis* (pp. 312-340). Springer Berlin Heidelberg.

Grant, J., and Hunter, A. (2011, July). Measuring the good and the bad in inconsistent information. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* (Vol. 22, No. 3, p. 2632).

Grau, G., Carvallo, J. P., Franch, X., & Quer, C. (2004, August). DesCOTS: a software system for selecting COTS components. In *Euromicro Conference, 2004. Proceedings. 30th* (pp. 118-126). IEEE.

Grefen, P. W., & Apers, P. M. (1993). Integrity control in relational database systems—an overview. *Data & Knowledge Engineering*, *10*(2), 187-223.

Gottlob, G., & Zicari, R. (1988, August). Closed World Databases Opened Through Null Values. In *VLDB* (Vol. 88, pp. 50-61).

Halevy, A. Y., Ives, Z. G., Suciu, D., & Tatarinov, I. (2003, March). Schema mediation in peer data management systems. In *Data Engineering, 2003. Proceedings. 19th International Conference on* (pp. 505-516). IEEE.

Harris, S., and Seaborne, A. (2013). SPARQL 1.1 query language. W3C Recommendation 21 March 2013

Hayes, P., and McBride, B. (2004). RDF Semantics. W3C Recommendation 10 February 2004. Last accessed 12 October 2013 at: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Hendler, J. (2009). Web 3.0 Emerging. *Computer*, *42*(1), 111-113.

Hernich, A., Libkin, L., & Schweikardt, N. (2011). Closed world data exchange. *ACM Transactions on Database Systems (TODS)*, *36*(2), 14.

Hitzler, P., Krotzsch, M., and Rudolph, S. (2011). *Foundations of semantic web technologies*. CRC Press.

Hlupic, V., & Mann, A. S. (1995, December). SimSelect: a system for simulation software selection. In *Simulation Conference Proceedings, 1995. Winter* (pp. 720-727). IEEE.

Hunter, A., and Konieczny, S. (2005). Approaches to measuring inconsistent information. In *Inconsistency tolerance* (pp. 191-236). Springer Berlin Heidelberg.

Jadhav, A. S., & Sonar, R. M. (2009). Evaluating and selecting software packages: A review. *Information and software technology*, *51*(3), 555-563.

Jiang, G., Pathak, J., & Chute, C. G. (2009). Formalizing ICD coding rules using formal concept analysis. *Journal of Biomedical Informatics*, *42*(3), 504-517.

Jäschke, R., and Rudolph, S. (2013). Attribute Exploration on the Web. *Peggy Cellier Felix Distel*, 19.

Keim, D., Andrienko, G., Fekete, J. D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). *Visual analytics: Definition, process, and challenges* (pp. 154-175). Springer Berlin Heidelberg.

Keim, D. A. (2001). Visual exploration of large data sets. *Communications of the ACM*, *44*(8), 38-44.

Kelliher, F. (2011). Interpretivism and the pursuit of research legitimisation: an integrated approach to single case design. *Leading Issues in Business Research Methods*, *1*, 45.

Kitchenham, B. A. (1996). Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods. *ACM SIGSOFT Software Engineering Notes*, *21*(1), 11-14.

Kitchenham, B., Linkman, S., & Law, D. (1997). DESMET: a methodology for evaluating software engineering methods and tools. *Computing & Control Engineering Journal*, *8*(3), 120-126.

Klyne, G., Carroll, J. J., and McBride, B. (2004). Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, 10. Available at http://www.w3.org/TR/rdf-concepts/#section-SimpleFacts

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28-44.
Kimball, R., & Caserta, J. (2004). The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. *Canada: Wiley Publishing. Inc*.

Kolaitis, P. G. (2005, June). Schema mappings, data exchange, and metadata management. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 61-75). ACM.

Kumar, M., Garg, D. P., & Zachery, R. A. (2007). A method for judicious fusion of inconsistent multiple sensor data. *Sensors Journal, IEEE*, *7*(5), 723-733.

Lano, K. (2014). Null considered harmful (for transformation verification). In *VOLT 2014, STAF conference, York*.

Lenzerini, M. (2002, June). Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 233-246). ACM.

Libkin, L. (2006, June). Data exchange and incomplete information. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 60-69). ACM.

Ma, Y., Qi, G., Hitzler, P., and Lin, Z. (2007). Measuring inconsistency for description logics based on paraconsistent semantics. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (pp. 30-41). Springer Berlin Heidelberg.

McLeod, K., and Burger, A. (2007, February). Using argumentation to tackle inconsistency and incompleteness in online distributed life science resources. In *Proceedings of IADIS International Conference Applied Computing* (pp. 489-492).

McLeod, K., and Burger, A. (2011). WP7 requirement document of CUBIST Consortium 2010-2013. Available at http://www.cubist-project.eu/fileadmin/CUBIST/user_upload/Deliverable/CUBIST_D7.1.1_HWU_v1.0.pdf

Melo, C., Aufaure, M. A., Orphanides, C., Andrews, S., McLeod, K., & Burger, A. (2013, March). A conceptual approach to gene expression analysis enhanced by visual analytics. *In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 1314-1319). ACM.*

Messai, N., Devignes, M. D., Napoli, A., & Smail-Tabbone, M. (2008, July). Many-Valued Concept Lattices for Conceptual Clustering and Information Retrieval. In *ECAI* (Vol. 178, pp. 127-131).

Mohamed, A., Wanyama, T., Ruhe, G., Eberlein, A., & Far, B. (2004). COTS evaluation supported by knowledge bases. In *Advances in Learning Software Organizations* (pp. 43-54). Springer Berlin Heidelberg.

Motro, A. (1993). Integrity= validity+ completeness. *ACM Transactions on Database Systems (TODS)*, *14*(4), 480-502.

Nwagwu, H. C. (2013) Evaluating and Analyzing Inconsistent RDF Data in a Semantic Dataset: EMAGE Dataset. In The 3rd CUBIST Workshop.

Nwagwu, H. C. (2014). Visualising Inconsistency and Incompleteness in RDF Gene Expression Data using FCA. *International Journal of Conceptual Structures and Smart Applications (IJCSSA)*, *2*(1), 68-82.

Nwagwu, H. C., and Orphanides, C (2015). Visual Analysis of a Large and Noisy Dataset, Submitted to International Journal of Conceptual Structures and Smart Applications (IJCSSA),

Obiedkov, S., Kourie, D. G., and Eloff, J. H. (2009). Building access control models with attribute exploration. *Computers & Security*, *28*(1), 2-7.

Onwuegbuzie, A. J., & Leech, N. L. (2006). Linking research questions to mixed methods data analysis procedures. *The Qualitative Report*, *11*(3), 474-498.

Onwuegbuzie, A. J., & Leech, N. L. (2007). Validity and qualitative research: An oxymoron?. *Quality & Quantity*, *41*(2), 233-249.

Pensa, R. G., & Boulicaut, J. F. (2005a). Towards fault-tolerant formal concept analysis. In *AI* IA 2005: Advances in Artificial Intelligence* (pp. 212-223). Springer Berlin Heidelberg.

Pensa, R. G., and Boulicaut, J. F. (2005b). From local pattern mining to relevant bi-cluster characterization. In *Advances in Intelligent Data Analysis VI* (pp. 293-304). Springer Berlin Heidelberg.

Polovina, S. (2013). A Transaction-Oriented Architecture for Enterprise Systems. *International Journal of Intelligent Information Technologies (IJIIT)*, *9*(4), 69-79.

Powers, S. (2003). *Practical rdf*. " O'Reilly Media, Inc.".

Péron, Y., Raimbault, F., Ménier, G., & Marteau, P. F. (2011). On the detection of inconsistencies in RDF data sets and their correction at ontological level.

Quilitz, B., and Leser, U. (2008). Querying distributed RDF data sources with SPARQL (pp. 524-538). Springer Berlin Heidelberg.

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, *23*(4), 3-13.

Raman, V., & Hellerstein, J. M. (2001, September). Potter's wheel: An interactive data cleaning system. In *VLDB* (Vol. 1, pp. 381-390).

Reiter, R. (1978). *On closed world data bases* (pp. 55-76). Springer US.

Richardson L, Venkataraman S, Stevenson P, Yang Y, Moss J, Graham L, Burton N, Hill B, Rao J, Baldock RA, Armit C**.** (2014) *EMAGE mouse embryo spatial gene expression database: (2014 update)* Nucleic Acids Res. 42(1):D835-44. doi: 10.1093/nar/gkt1155

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, *14*(2), 131-164.

Sandelowski, M. (2000). Focus on research methods-whatever happened to qualitative description?. *Research in nursing and health*, *23*(4), 334-340.

Scheglmann, S., Gröner, G., Staab, S., & Lämmel, R. (2013, January). Incompleteness-aware programming with RDF data. In *Proceedings of the 2013 workshop on Data driven functional programming* (pp. 11-14). ACM.

Sertkaya, B. (2009). Ontocomp: A protege plugin for completing owl ontologies. In *The Semantic Web: Research and Applications* (pp. 898-902). Springer Berlin Heidelberg.

Schell, C. (1992). The value of the case study as a research strategy. *Manchester, UK: University of Manchester, Manchester Business School*, 1-15.

Schofield, J. W. (2002). Increasing the generalizability of qualitative research. *The qualitative researcher's companion*, 171-203.

Sertkaya, B. (2009). Ontocomp: A protege plugin for completing owl ontologies. In *The Semantic Web: Research and Applications* (pp. 898-902). Springer Berlin Heidelberg.

Sheth, A. P., and Ramakrishnan, C. (2003). Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. *IEEE Data Eng. Bull.*, *26*(4), 40-48.

Sheth, A. (2005). From semantic search & integration to analytics. In *Semantic Interoperability and Integration*.

Sirin, E., & Tao, J. (2009, October). Towards Integrity Constraints in OWL. In *OWLED* (Vol. 529).

Stoilov, D., and Bishop B. (2012), OWLIM-SE Reasoner [online]. Available at: http://owlim.ontotext.com/display/OWLIMv52/OWLIM-SE+Reasoner

Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., & Lakhal, L. (2002). Computing iceberg concept lattices with TITANIC. *Data & knowledge engineering*, *42*(2), 189-222.

Taylor, A., McLeod, K., & Burger, A. (2013). Semantic Visualisation of Gene Expression Information. In *CUBIST Workshop* (pp. 10-20).

Venkataraman S, Stevenson P, Yang Y, Richardson L, Burton N, Perry TP, Smith P, Baldock RA, Davidson DR, Christiansen JH. *EMAGE: Edinburgh Mouse Atlas of Gene Expression: 2008 update.* Nucleic Acids Res. 2008 36:D860-5.

Waraporn, N., & Porkaew, K. (2008). Null semantics for subqueries and atomic predicates. *IAENG International Journal of Computer Science*, *35*(3), 305-313.

Wijsen, J. (2006). A note on database repairing by value modification. *Pre-proceedings of the EDBT*, *6*, 104-107.

Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): Ordered sets. Reidel, Dordrecht-Boston 1982, 445–470.

Wolff, K. E. (1993). A first course in formal concept analysis. *SoftStat*, *93*, 429-438.

Yin, R., K.: Case Study Research: Design and Methods (4th Ed.). Applied Social Research Methods Series, Vol. 5. SAGE (2009)
Zainal, Z. (2007). Case study as a research method. *Jurnal Kemanusiaan bil*.

Zimányi, E., & Pirotte, A. (1997). Imperfect knowledge in databases. In *Proceedings of the Workshop on Uncertainty Management in Information Systems: From Needs to Solutions* (pp. 136-186).

APPENDIX E

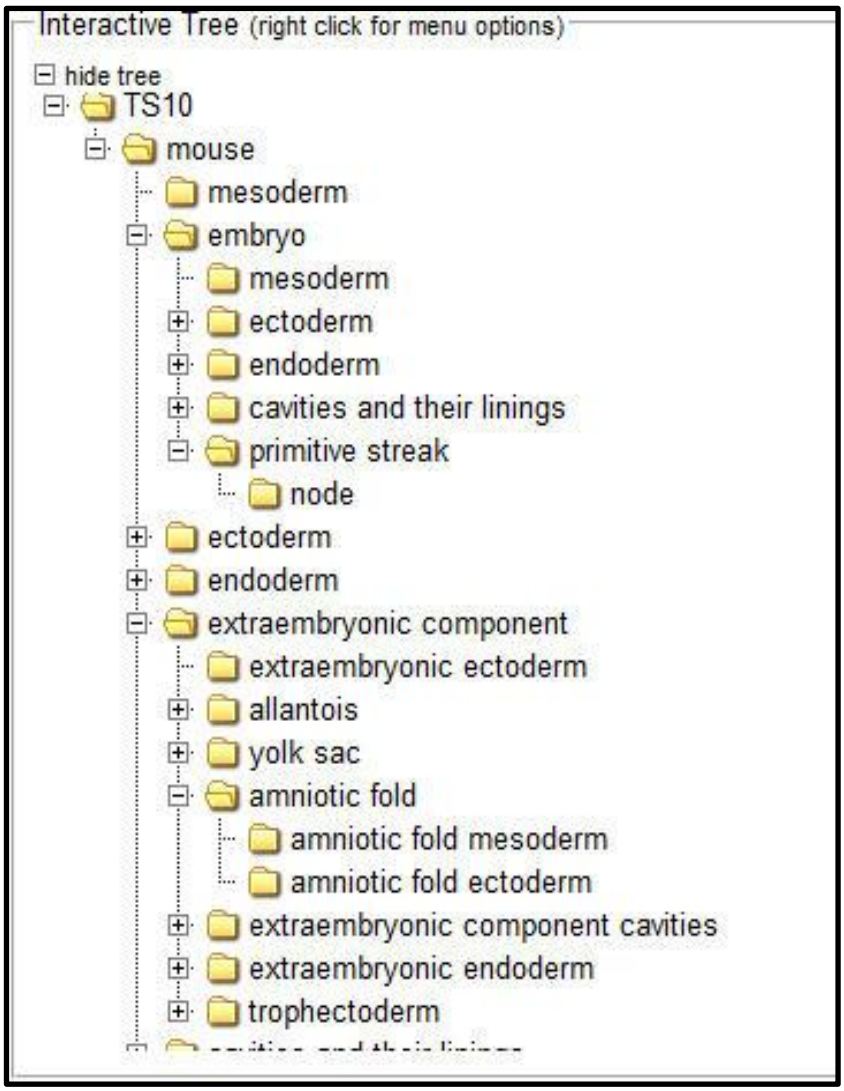

Figure 44: A part of the EMAP Anatomy Ontology of Theiler Stage 10 available in http://www.eMouseatlas.org/emap/ema/DAOAnatomyJSP/anatomy.html?stage=TS10 Last viewed on 12th May 2015

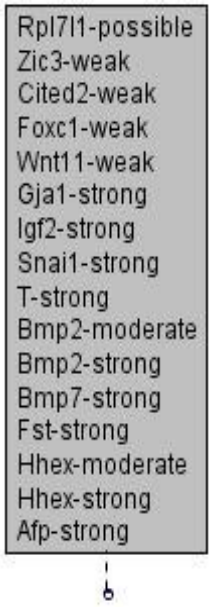

Figure 45: Analogue incompleteness in non-propagated data set in TS 11

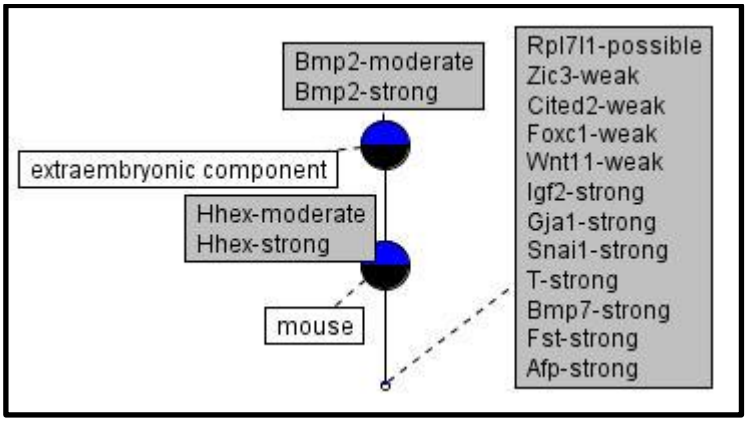

Figure 46: Analogue IID in positively propagated data set in TS 10

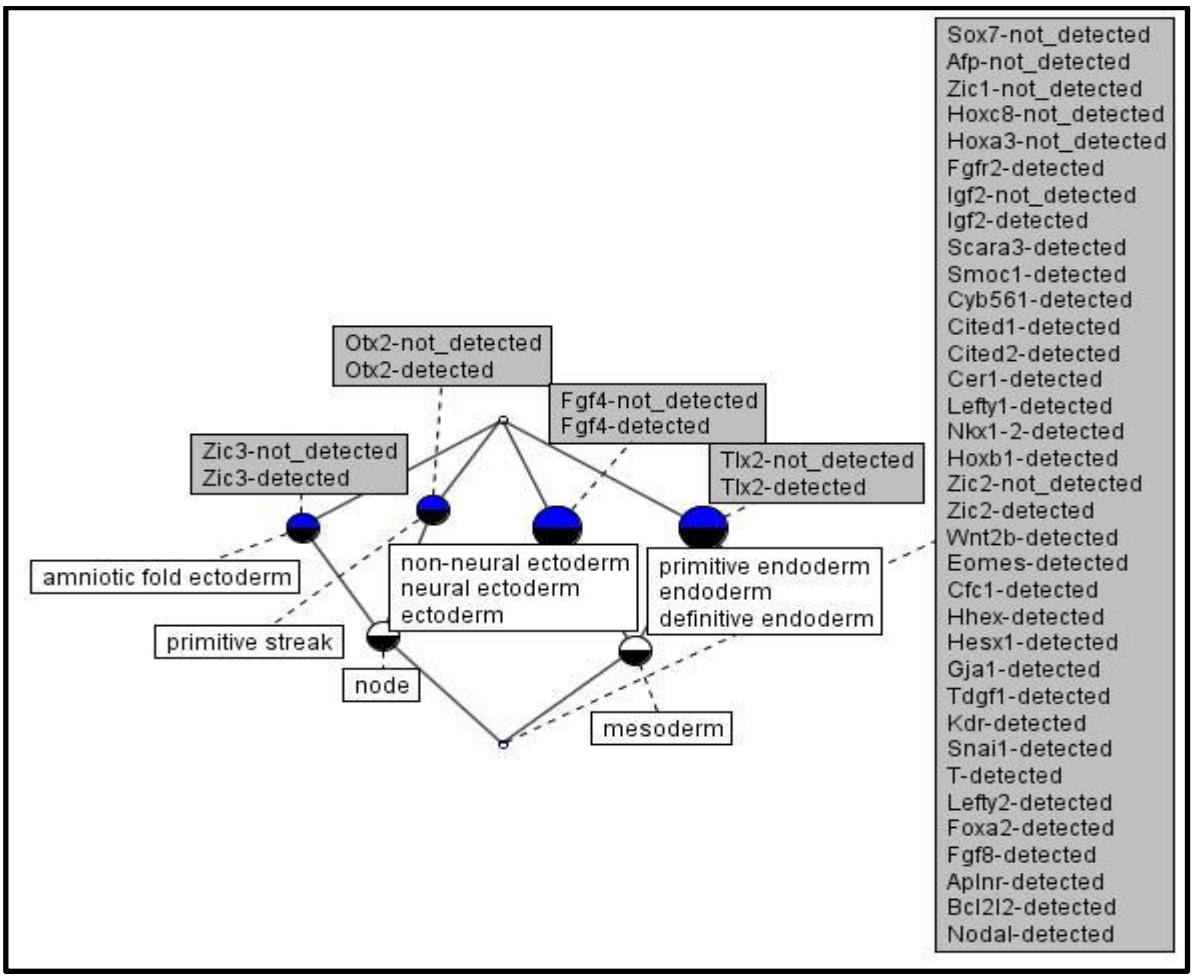

Figure 47: Binary IID in positively propagated data set in TS 10

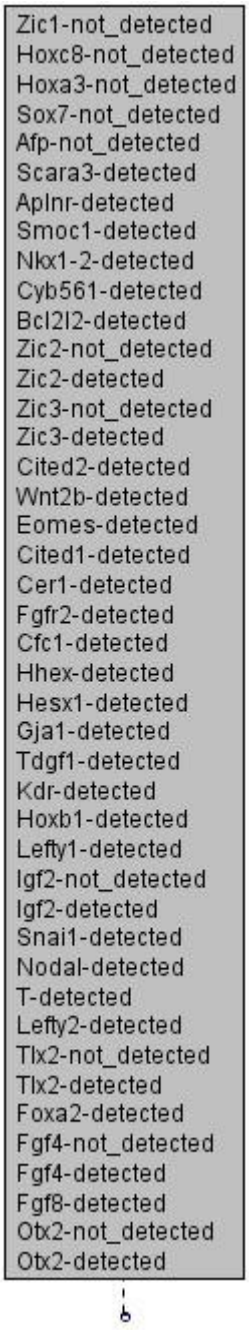

Figure 48: Binary incompleteness in non-propagated data set in TS 10