XML CLEANING MODEL FOR DATA QUALITY IMPROVEMENT USING CONDITIONAL INTEGRITY CONSTRAINTS

by

MOHAMMED RAGHEB HAKAWATI

(1540211789)

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

School of Computer and Communication Engineering

UNIVERSITI MALAYSIA PERLIS

2018

UNIVERSITI MALAYSIA PERLIS

DECLARATION OF THESIS

Author's Full Name	MOHAMMED RAGHEB HAKAWATI
Title	XML Cleaning Model for Data Quality Improvement Using Conditional Integrity Constraints
Date of Birth	09 MAY 1982
Academic Session	2017/2018

I hereby declare that this thesis becomes the property of Universiti Malaysia Perlis (UniMAP) and to be placed at the library of UniMAP. This thesis is classified as:

	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997) *
	RESTRICTED	(Contains restricted information as specified by the organization where research was done) *
✓	**OPEN ACCESS**	I agree that my thesis to be published as online open access (Full Text)

I, the author, give permission to reproduce this thesis in whole or in part for the purpose of research or academic exchange only (except during the period of Five years, if so requested above)

Certified by:

SIGNATURE OF CANDIDATE	**SIGNATURE OF SUPERVISOR**
N011497913	DR. YASMIN MOHD YACOB
(PASSPORT NO)	**NAME OF SUPERVISOR**
Date: 09 July 2018	Date: 09 July 2018

In the Name of Allah, The Most Beneficent, The Most Merciful

ACKNOWLEDGMENT

Coming to the end of this long journey, it is my pleasure to express my gratitude to a large number of people who have contributed, in many different ways, to make my success a part of their own.

I would like to express my gratitude to Dr. Yasmin Yacob, for her teaching and countless hours of help and guidance provided throughout the completion of this research work. The knowledge shared with me is not only academic but also life-long lessons for which I am grateful. In addition, I must thank my co-supervisors Dr. Amiza Amir and Dr. Rafikha Raof for their ideas, insights, and contributions brought to improve the quality of this research work. Finally, I would like to thank Prof. Puteh Saad, for her patience, guidance, respect, encouragement, and support provided for me during the beginning of my study.

I also could not have been here without the love and support of my family: my father, mother, brother, and warmhearted sisters; to whom I have always looked when the going went rough, and for keeping me on the right track. I also would like to thank my lovely wife, Seham for her encouragement and understanding during three years of life away from home. Finally, I would like to dedicate this humble work to my life hope and my eyes light, my daughters, Masah and Taj.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFD	Approximate Functional Dependencies
CD	Conditional Dependencies
CFD	Conditional Functional Dependencies
CIND	Conditional Inclusion Dependencies
CRM	Customer Relationship Management
DSS	Decision Support Systems
DTD	Document Type Definition
ERD	Entity Relationship Diagram
ETL	Extract, Transform, Loading
FD	Functional Dependencies
FK	Foreign Key
GTT	Generalized Tree Tuple
IC	Integrity Constraints
JSON	Java Script Object Notation
MD	Matching Dependencies
ORM	Object Role Modelling
RDF	Resource Description Framework
SysML	Systems Modelling Language
TT	Tree Tuple
UML	Unified Modelling Language
XAFD	XML Approximate Functional Dependencies
XCFD	XML Conditional Functional Dependencies
XCIND	XML Conditional Inclusion Dependencies
XCSD	XML Conditional Structural Dependencies
XFD	XML Functional Dependencies
XIND	XML Inclusion Dependencies
XML	Extensible Markup Language
XNF	XML Normal Form
XSD	XML Schema Definition

LIST OF SYMBOLS

φ	XCFD Dependency
ψ	XCIND Dependency
θ	Support threshold of the XIND dependency
δ	Confidence threshold of the XIND dependency
θ_{tp}	Support threshold of the XCIND dependency
δ_{tp}	Confidence threshold of the XCIND dependency
ϵ	Number of Inconsistencies
$\mathrm{R_P}$	Relation to Pivot Path
C_P	Tuple Class
μ	Noise Factor
ε	Error threshold

MODEL PEMBERSIHAN XML UNTUK PENAMBAHBAIKAN KUALITI DATA MENGGUNAKAN KEKANGAN KONDISI INTEGRITI

ABSTRAK

Extensible Markup Language (XML) muncul sebagai piawaian utama dalam mewakili dan bertukar data,iaitu dengan lebih daripada 60% daripada jumlah, XML dianggap sebagai jenis dokumen yang paling dominan di laman sesawang. Namun, kualiti XML tidak seperti yang dijangkakan. Maka, semakin penting untuk menyediakan model penuh bagi mengesan, dan membetulkan sifat tidak konsisten yang diakui sebagai pelanggaran terhadap kebergantungan data yang menyebabkan kualiti data XML berkurangan. Kekangan integriti XML memainkan peranan penting bagi memastikan set data XML berfungsi secara konsisten. Walau bagaimanapun, kemampuannya untuk menyelesaikan isu-isu kualiti data masih kurang berkesan. Sebab utama masalah ini adalah berpunca daripada kebergantungan terhadap data model lama yang secara asasnya hanya memastikan keberkesanan skema dan bukannya data itu sendiri. Tujuan kajian ini untuk meningkatkan kualiti dokumen XML dengan memperkenalkan model pembersihan yang dipertingkatkan berdasarkan model kekangan integriti XML baru yang dipanggil *kebergantungan sandaran penyertaan XML* (XCIND) dan *kebergantungan sandaran fungsion XML (XCFD)*. Notasi peraturan baru direka terutamanya bagi meningkatkan contoh data dan meluaskan kebergantungan model lama XML dengan menguatkuasakan jadual corak konstan berkaitan semantik. Seterusnya, satu set kebergantungan bersyarat anggaran minima (XCFD, XCIND) ditemui dan dipelajari dari pokok XML menggunakan satu set algoritma perlombongan. Akhirnya, data tidak konsisten akan dikesan menggunakan pertanyaan penolakan untuk peraturan perlombongan dan dibaiki menggunakan set pernyataan kemas kini yang berbeza sebagai penyelesaian untuk nilai data yang tidak konsisten. Melalui penilaian eksperimen yang meluas pada set data XML yang sebenar, algoritma perlombongan yang dicadangkan menunjukkan keberkesanan dan prestasi tinggi dalam menemui semua kebergantungan bersyarat yang berbeza nilai ambang *sokongan* dan *keyakinan*. Keputusan menunjukkan bahawa model baru boleh meningkatkan kualiti XML dengan mengesan nilai sebenar data yang palsu daripada model sebelumnya yang bergantung kepada kebergantungan tradisional. Tambahan pula, *XML Cleaner* dapat merasakan sifat tidak konsisten antara pokok tupel sama atau antara pokok tupel pelbagai tahap di dalam pokok XML menggunakan kebergantungan bersyarat yang dinyatakan. Selanjutnya, kualiti dokumen dinilai menggunakan dua ukuran (Ketepatan dan *Ingat Semula*) dan, ketepatan dokumen XML bertambah baik untuk ukuran tersebut masing-masing melebihi 94% dan 83%. Akhirnya, kekangan XML integriti bersyarat sama seperti hubungan yang lain, membuktikan keupayaannya untuk menghasilkan piawai baru aplikasi pembersihan untuk model data XML yang lebih baik, terutamanya dalam era data raya.

XML Cleaning Model for Data Quality Improvement Using Conditional Integrity Constraints

ABSTRACT

Extensible Markup Language (XML) is emerging as the primary standard for representing and exchanging data, with more than 60% of the total, XML considered the most dominant document type over the web; nevertheless, their quality is not as expected. Consequently, it has become increasingly important to provide a full model which is able to detect, and correct inconsistencies recognized as violations of data dependencies causing the decrease of XML data quality. XML integrity constraint plays an important role in keeping XML dataset as consistent as possible, but their ability to solve data quality issues is still intangible. The main reason is that old-fashioned data dependencies were basically introduced to maintain the consistency of schema rather than that of data. The purpose of this study is to improve the quality of XML documents by introducing an enhanced cleaning model based on a new type of XML integrity constraints called XML Conditional Inclusion Dependencies (XCIND) and XML Conditional Functional dependencies (XCFD). The notations of the new rules are designed mainly for improving data instance and extended traditional XML dependencies by enforcing pattern tableaus of semantically related constants. Subsequent to this, a set of minimal approximate conditional dependencies (XCFD, XCIND) is discovered and learned from the XML tree using a set of mining algorithms. Finally, data inconsistencies are detected using denial queries for mined rules and repaired using a different set of update statements as solutions for inconsistent data values. Through the extensive experimental evaluation of real XML datasets, proposed mining algorithms demonstrated their efficacy and high performance in discovering all conditional dependencies with different *support* and *confidence* thresholds. The results showed that the new model could increase XML quality by detecting more real spurious data values than previous models based on traditional dependencies. Furthermore, the XML Cleaner can sense inconsistencies between same tree tuples or even between multilevel tree tuples insides the XML tree using the mentioned conditional dependencies. Moreover, the quality of the documents was assessed using two measures (Precision and Recall), and the accuracy of XML documents was improved over 94%, 83% respectively for these measures. To this end, XML conditional integrity constraints, just as their relational counterpart, prove their ability to pave the way toward new standards of cleaning applications for XML data model, especially in the big data era.

Keywords: XML, Integrity Constraints, Conditional Dependencies, Data Quality, Data Cleaning.

CHAPTER 1

INTRODUCTION

1.1 Overview

Today, data become the lifeblood of businesses, as different database applications, such as Decision Support Systems, Customer Relationship Management, Data Warehouses, Web Services, and eLearning Systems are being used; beneficial information and knowledge can be gained from considerable amounts of data. However, investigations demonstrate that heaps of such applications fail to run successfully and efficiently due to many issues, such as poor system design or weak query performance, yet nothing is sure to cause applications failure than the carelessness of data quality issues (Juddoo, 2016; Li, 2012).

According to studies and reports presented by *V12-Data* in 2015, the expenses of bad data might be considerably higher than 12% lost revenue. About 28% of individuals who had issues related to the delivery of emails said that customer service has endured accordingly, while 21% experienced reputation damage. The vast majority of the organizations (86%) admitted that their data might be inaccurate somehow. About 44% of businesses and organizations reported that missing or imperfect data are the most frequent problems alongside obsolete information (Bedgood, 2015)

Therefore, organizations that seek to extract valuable or high-quality information from raw or low-quality data must engage in the process of data cleaning as an essential process as shown in Figure 1.1. Many raw datasets typically contain erroneous information such as misspellings and missing values. Although cleaning of data has been a long-established issue, it becomes critical again due to the increased interest in web data and big data (Saha & Srivastava, 2014).

Figure 1.1: The Relationship between Data Quality and Data Cleaning.

The close relationship between big data and data cleaning has gained much attention in the last decade (Caldarola & Rinaldi, 2015; Chen et al., 2013; Fan, 2015; Jagadish et al., 2014; Saha & Srivastava, 2014). That is because nothing meaningful can be obtained from a significant amount of corrupted information.

Nowadays, the need to effectively manage business information, which is filled with inconsistencies and incompleteness, is more important than ever before to help business making right decisions, deriving accurate reports, and improving the overall trustworthiness of available data sources. Numerous investigations conducted by the Computing Research Association (2012), have highlighted the value of effective and efficient techniques for handling " erroneous data" at scale. Despite the fact that this issue has gotten critical consideration over time in the relational database literature (Fan & Geerts, 2012), XML cleaning approaches fall short in providing a practical solution for big data and web data (Chen et al., 2013).

Extensible Markup Language (XML) stands out rapidly amongst essential data file formats; It has been used for scientific data such as DNA sequences (Roberts, Vincze, Posfai, & Macelis, 2015), to annotate extensive documents such as DrugBank database (Knox et al., 2011), or for exchanging data over the Web for e-commerce benefits (Chan, Lee, & Heng, 2014). Furthermore, giant software vendors, including Oracle, Microsoft, IBM, as well as new startup companies such Altova, Oxygen are developing tools to manage XML data and applications like XML Spy and XML editor (Altova, 2017; Oxygen, 2017).

Grijzenhout & Marx (2013), provide in-depth analysis to answer the question "Is the quality of XML documents found on the web sufficient to apply XML technologies like XQuery, XPath, and XSLT?" The results show that on the web, 58% of the existing documents are of the XML file format, nevertheless, one-third of these documents accompanying with valid XML Schema Definition (XSD) or Document Type Definition (DTD). Moreover, about 14% of the documents lack well-formedness. A simple error of mismatching or missing tags will render the entire XML technologies useless over these documents.

The growing interest of XML as the dominant way of exchanging data over the Web, encourages researchers to address XML data cleaning as an open research problem (Fan, Geerts, & Jia, 2008a), and to start searching for data cleaning approaches for XML (Tang, Shao, Ba, Senellart, & Bressan, 2015; Weis, Monod, & Cedex, 2007) especially approaches based on integrity constraints (Hamrouni, Brahmia, & Bouaziz, 2015; Švirec & Mlýnková, 2012).

Data cleaning approaches for XML dataset are as old as XML itself, from 1997 until now, most of them have focused on schema matching to identify and repair data inconsistencies (Algergawy, Nayak, & Saake, 2010; Rusu, Rahayu, & Taniar, 2005; Weis, Naumann, & Brosy, 2006). However, there has been little discussion about data cleaning perspectives used in terms of integrity constraints (Fan, 2005; Flesca, Furfaro, Greco, & Zumpano, 2003; Lima, Rezende, & Oliveira, 2013; Shahriar & Anam, 2008; Tan & Zhang, 2011a; Yu & Jagadish, 2008), which open doors for researchers to address this problem (Almeida, Maio, Oliveira, & Barroso, 2016).

The study of Integrity Constraints (IC) stands out amongst the most critical yet challenging research topics in database theory for schema optimization. For relational databases, constraints are essential to schema design, query optimization, efficient storage, and access methods, for all reasons, relational integrity constraints are important (Elmasri & Navathe, 2016). XML data model, much the same as a relational model, can identify by *type* constraints (int, string, date) and *integrity* constraints (Function, Inclusion). Integrity constraints are essential for the semantics of XML data specifications, moreover, they are beneficial for query optimization, update anomaly prevention, and for information preservation during the process of data integration (Fan & Simeon, 2003).

Integrity constraint cleaning approaches focused on two directions (Bertossi, 2011): *repairing* to find a new dataset that is valid with a minimum difference from the original database (Flesca, Furfaro, & Parisi, 2010; Molinaro, Chomicki, & Marcinkowski, 2009), and *consistent query answering* to provide a result for a given query in every repair of the original database without editing the data (Lian, Chen, & Song, 2010; Staworko & Chomicki, 2006).

Recently, an improved type of Data Dependencies (Integrity Constraints) have been developed to detect data inconsistencies in relational databases called Conditional Dependencies. Conditional Functional Dependencies (CFD) (Bohannon, Fan, Geerts, Jia, & Kementsietsidis, 2007) and Conditional Inclusion Dependencies (CIND) (Fan, Bravo, & Ma, 2007) are an extension of traditional Functional Dependencies (FD) and Inclusion Dependencies (IND) respectively (Elmasri & Navathe, 2016; Fan & Geerts, 2012), with more accurate, expressive and increased capability in terms of quality issues. Furthermore, these types of dependencies provided relational databases with semantics, and meaningful rules based on a subset of tuples, that matches a specific condition rather than entire relation like FDs or even INDs (Caruccio, Deufemia, & Polese, 2016).

Over time, conditional dependencies have proven their strength in error detection, data cleansing, and data auditing. At the same time, the demonstrations show that CFD cleaning approaches provide the user with a better understanding of the quality of the data, thereby assisting the user to improve data quality in an interactive way (Fan, 2012).

The important of CFD in the field of data cleaning encouraged researchers to develop many algorithms to discover and mine these dependencies from relational databases (Aqel, Shilbay, & Hakawati, 2012; Chiang & Miller, 2008; Golab, Karloff, Korn, Srivastava, & Yu, 2008), and proposing data cleaning approaches based on them (Beskales, Ilyas, Golab, & Galiullin, 2013; Fan, Geerts, Jia, & Kementsietsidis, 2008). Nevertheless, a single work addresses the discovery of these dependencies (XCFD) from XML dataset (Vo, Cao, & Rahayu, 2011).

1.2 Issues in XML Integrity Constraints

Semi-structured data model, besides the relational data model, is considered the most data model commonly used for storing, retrieving, and querying valuable data. XML is one of the most common document types over the web which follows semi-structured model (Grijzenhout & Marx, 2013). Because of the growing popularity of XML; the problem of clean XML data accurately and efficiently is revived recently especially with big data era (Abiteboul, Buneman, & Suciu, 2000; Liu, Vincent, & Liu, 2006; Saha & Srivastava, 2014; Tan & Zhang, 2011a, 2011b).

XML integrity constraints are the main criteria used in the classification of data as clean or not in terms of the consistency attribute (Ahmad & Ibrahim, 2008; Arenas &

Libkin, 2004; Arenas, 2006; Deutsch & Tannen, 2005; Fajt, Mlýnková, & Nečaský, 2011; Fan, 2005; Fan & Simeon, 2003; Hakawati et al., 2017; Hartmann, Köhler, Link, Trinh, & Wang, 2008; Karlinger, Vincent, & Schrefl, 2009; Liu, Li, Liu, & Chen, 2012; Shahriar & Liu, 2009; Vincent, Liu, & Liu, 2004; Vo et al., 2011).

Contrariwise relational databases; XML data model has more than a single schema, this fact interprets the multi-data dependencies notations taken into consideration. Some of these notations extend relational *tuples* concept (Arenas, 2006; Arenas & Libkin, 2004; Fan & Simeon, 2003; Yu & Jagadish, 2008, 2006), whereas the others deal with XML as a tree containing a set of *paths* (Ahmad & Ibrahim, 2008; Karlinger et al., 2009; Shahriar & Liu, 2009; Vincent & Liu, 2003; Vincent, Liu, et al., 2004; Vincent, Liu, & Mohania, 2007; Vincent, Schrefl, Liu, Liu, & Dogen, 2004).

Furthermore, XML Functional Dependencies (XFD) are the most notable IC used in the enhancement of data instance (Flesca, Furfaro, Greco, & Zumpano, 2005; Flesca et al., 2003; Hamrouni et al., 2015; Švirec & Mlýnková, 2012; Tan & Zhang, 2011a, 2011b; Yu & Jagadish, 2008). On the other side, matching dependencies, inclusion dependencies, approximate dependencies, conditional dependencies, and association rules have also played important roles in the improvement of relational databases (Adhikari & Rao, 2008; Ebaid et al., 2013; Fan, Geerts, & Jia, 2008a; Gardezi & Bertossi, 2011; Geerts, Mecca, Papotti, & Santoro, 2013; Mayfield, Neville, & Prabhakar, 2010).

Functional Dependencies for XML (XFD), as an extension of relational ones (Vincent et al., 2007), are designed for semantic expressiveness to prevent schema problems (Normalization and Redundancies Detection), in spite of the fact that these dependencies are widely used in improving XML schema (Arenas, 2006; Fan & Simeon, 2003; Vincent et al., 2007; Yu & Jagadish, 2008), they cannot express a proper type of constraints that hold on a subset of XML data (Vo et al., 2011). The main reason is this kind of dependencies covers the whole XML tree and lack of flexibility to accept domain values (Pattern Tableaus) within the rule that matched a subset of the tree conditionally (Bohannon, et al., 2007; Liu et al., 2012). As a result, their ability to detect inconsistencies under a subset of the tree and inside path leaves within the tree tuples that matched pattern tableaus is limited, consequently, any cleaning approach utilize these kinds of dependencies (Bloodgood & Strauss, 2016; Hamrouni et al., 2015; Hartmann et al., 2008; Švirec & Mlýnková, 2012; Tan & Zhang, 2011b; Tan, Zhang, Wang, & Shi, 2013; Yan, Lv, & He, 2014) may not yield maximum benefit and utilization, especially when looking for data inconsistencies.

On the other side, up to date, none has attempted to utilize XML Inclusion Dependencies (XIND) in cleaning XML data, because these dependencies required mainly for generating XML foreign keys rather than consistency issues (Fajt, Mlýnková, et al., 2011; Karlinger et al., 2009; S. Shahriar, Liu, 2009; Vincent, Schrefl, et al., 2004). However, many authors advise that using IND as a collaborative constraint with functional dependency will help in detecting more inconsistencies and reducing database faults, thereby totally improving the database quality (Bohannon, Fan, Flaster, & Rastogi, 2005). Furthermore, a modified version of IND with Conditions (CIND) presents an important role for enhancing relational database consistency, in addition to schema optimization (Fan et al., 2007; Ma, Fan, & Bravo, 2014).

1.3 Problem Statement

Increasing the quality of the XML document is crucial for the continued competitiveness of data to help business in making right decisions, deriving accurate reports, and improving the overall trustworthiness of available data sources. More precisely, better data quality leads to less financial costs, less time consumed, and less repairing efforts wasted on poor campaigns (Fan, 2015). However, Data Consistency as one of the five attributes besides Data Accuracy, Data Completeness, Data Currency, and Data Deduplication, used for improving data quality, especially in terms of data validity and integrity using a set of dependency rules known as *integrity constraints* (Fan & Geerts, 2012).

Inspiration from the relational database; conditional dependencies (CFD, CIND) were presented to overcome relational traditional dependencies (FD, IND) limitations, especially data quality issues (Bohannon et al., 2007; Fan et al., 2007). The conditional dependencies own more quality characteristics make them directed toward data cleaning such as covering a subset of the dataset (Fan & Geerts, 2012). Furthermore, these dependencies proved their efficiency in eliminating inconsistencies from relational databases and detecting more inaccurate tuples and fields within tuples over traditional dependencies. Furthermore, cleaning approaches that adopted these dependencies are considered the most used techniques in the last ten years, besides crowdsourcing and knowledge base cleaning approaches (Ganti & Sarma, 2013).

On the other hand, Fassetti and Fazzinga (2007), highlighted the importance of XML *Approximate* dependencies in the area of data cleaning over *Full* dependencies, which is caring more about data integration and schema enhancement. However, this type of dependencies allows the discovery of erroneous or exceptional elements in the data, besides identifying constraints very frequently in the database that are meaningful for data cleaning and analysis issue, even if they are not valid in the whole database.

Furthermore, to develop a constraint-based cleaning model, numerous methods were used to combine data dependencies with databases, for instance, domain experts, crowdsourcing, and rules mining are the main techniques used in creating integrity constraints and business rules (Chu et al., 2015; Chu, Ilyas, Krishnan, & Wang, 2016; Debattista, Lange, & Auer, 2014). In practice, it is necessary to have in place a technique that can automatically discover or learn required dependencies from the excited XML data to be used as data cleaning rules (Fan, Geerts, Jianzhong, & Xiong, 2011).

Previous XML cleaning techniques disregarded the problem of rules discovering (dependencies mining) and used a set of assigned dependencies instead (Flesca et al., 2005; Švirec & Mlýnková, 2012; Tan & Zhang, 2011a). Indeed, it is often unrealistic to solely count on human experts to design data dependencies by an expensive and long manual process. As indicated by Gratner (2007), cleaning rules discovery is critical to commercial data quality tools. Furthermore, assigning a set of dependencies required checking their satisfiability using a long-standing process known as *chasing* (Karlinger et al., 2009; Meier, 2010).

Discovering *approximate conditional* dependencies from an XML dataset (XCFD, XCIND) using previous mining techniques is not an easy step for many reasons; Firstly, traditional dependencies expressing an XML tree do not own patterns tableaus like conditional dependencies, and cover the whole dataset (have support threshold equal to 1) instead of the required XML subset (Yu & Jagadish, 2008). Moreover, these dependencies have no exceptions as an error ratio (have confidence threshold equal to 1) to be flexible for data accuracy (Vo et al., 2011). Secondly, each dependency type uses different mining technique as each has a particular role; for instance, functional dependencies are based on the *association between elements* amongst both sides of the rules, whereas, the role of inclusion dependencies care about the *existence of elements* from left side of the dependency to the right side (Elmasri & Navathe, 2016; Liu et al., 2012).