ATTRIBUTES SANITIZATION IN OBJECT-ORIENTED DESIGN TO IMPROVE DATABASE STRUCTURE

MOHD ZAINURI BIN SARINGAT

A thesis submitted in
fulfillment of the requirement for the award of the
Doctor of Philosophy in Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

JUNE 2014

DEDICATION

Special dedicated to my family and all my friends. Without you, I would not be able to go so far as where I am now. Hopefully, Allah will bless us all. Ameen.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisor Prof. Dr. Rosziati Ibrahim who gave me the opportunity to be her student. I am deeply grateful to my supervisor for the continuous support of my Doctor of Philosophy study and research, for her patience, motivation, enthusiasm, and immense knowledge along this study. Her guidance helped me in all the time of research and writing of this thesis. Her extensive discussions around my work and interesting explorations in this new field have been very helpful for this study. I could not have imagined having a better supervisor and mentor for my Doctor of Philosophy study.

My sincere thanks also goes to all members in the Faculty of Computer Science and Information Technology for their encouragement and motivation in making me take my first step into Ph.D. studies.

I warmly thank all members in the postgraduate room for their valuable advice, constructive comments, and their important support throughout this work.

I owe my loving thanks to my wife Maznah Ismail and my kids. Thanks too to my loving parents Saringat Ithnin and Misnah Taib. Without their encouragement and understanding, it would have been impossible for me to finish this work. My special gratitude is due to my brother, and my sisters for their loving support.

Lastly, I also wish to thank all members that I cannot specify here for their support, cooperation and contribution along the way and the financial support from the government through the Ministry of Higher Education and Universiti Tun Hussein Onn Malaysia (UTHM) is gratefully acknowledged.

ABSTRACT

Modelling using the Entity Relationship Model was introduced more than thirty years ago. Until late 1990's, object-oriented introduced class diagram. However, designing a good database is still a serious issue. Some of the issues are very difficult to handle such as consistency checking between system design and database design, redundancy of data, mismatch of the data structure with the user's needs in the database and unused data in the database. In this thesis, a new technique called UInData is introduced as an alternative method for designing database based on attribute sanitization. The proposed technique will extract class behaviour from class diagram to produce schema table which will then be compared with the user interface to normalize the structure. Attributes sanitization is introduced to remove the unused attributes and to provide final schema table. An experiment using three case studies has shown that some improvements of designing optimal database have been achieved in term of data sanitization and data accessibility. Attribute sanitization was applied in LAS, SPKS and MPBP database. Data sanitizations have removed 2.2%, 14.1% and 24.5% from defined attributes which are not used by user interface. Meanwhile the results shown in data accessibility for these three cases have shown that LAS was reduced by 50%, SPKS have not reducing of data accessibility and MPBP was reduced by 20% when UInData is used as compared to using ordinary object-oriented. Therefore, the UInData is a good alternative technique to improve database structure.

ABSTRAK

Permodelan mengunakan Model Hubungan Entiti telah diperkenalkan sejak lebih tiga puluh tahun. Sehingga lewat 1990-an, kaedah berorientasikan objek memperkenalkan *class diagram*. Walaubagaimanapun, proses mereka bentuk pangkalan data yang baik masih menjadi isu yang besar. Antara isu yang dikenalpasti adalah sukar mempastikan keserasian diantara rekabentuk sistem dan pangkalan data, pertindanan data, stuktur pengkalan data tidak memenuhi kehendak pengguna dan atribut yang tidak digunakan dalam pengkalan data. Dalam tesis ini, satu teknik baru yang dinamakan UInData telah diperkenalkan sebagai kaedah alternatif untuk menghasilkan pangkalan data dengan membuang atribut. Teknik ini akan mengambilkira ciri-ciri *kelas* bagi menghasilkan jadual skema dan membandingkannya dengan antara muka penguna bagi melaksanakan struktur pernormalan. Pembuangan atribut diperkenalkan bagi membuang atribut yang tidak digunakan bagi menghasilkan skema jadual yang terkini. Pengujian terhadap tiga kajian kes mendapati penambahbaikan rekabentuk pangkalan data telah dicapai dalam penghapusan data dan capaian data. Penghapusan attribut telah diaplikasikan terhadap pangkalan data LAS, SPKS dan MPBP. Penghapusan data telah memadam 2.2%, 14.1% dan 24.5% dari attribut yang dikenalpasti bilamana ia tidak digunakan dalam antaramuka pengguna. Sementara itu dalam capaian data bagi ketiga-tiga kajian kes menunjukkan capaian data bagi LAS telah berkurangan sebanyak 50%, SPKS tiada perbezaan dalam kadar capian data dan MPBP telah berkurangan sebanyak 20%. Ini menunjukkan UinData adalah salah satu teknik yang berguna dalam menambahbaik rekabentuk pangkalan data.

CONTENTS

LIST OF PUBLICATIONS

Journals:

1. **Mohd Zainuri Saringat**, Rosziati Ibrahim, Noraini Ibrahim and Tutut Herawan. On Data Sanitization in Designing Database From Class Diagram. *Journal of Computing*, Vol. 3, Issue 4, April 2011.

2. **Mohd Zainuri Saringat**, Rosziati Ibrahim & Tutut Herawan. A Proposal for Constructing Relational Database from Class Diagram. *Canadian Journal: Computer and Information Science,* Vol. 3, No. 2, pp. 38-46, 2010.

Proceedings:

1. **Mohd Zainuri Saringat**, Rosziati Ibrahim, Noraini Ibrahim and Adekunle Adeshina. Constructing Schema Table from Class Diagram. 2nd World Conference on Information Technology (WCIT 2011). *AWERProcedia Information Technology and Computer Science*, Vol. 1, May 2012, pp 699-708.

2. **Mohd Zainuri Saringat**, Rosziati Ibrahim, Noraini Ibrahim and Tutut Herawan. On Database Normalization using User Interface Normal Form. *2010 International Conference On Intelligent Computing, Changsha, China. Lecture Notes in Computer Science, 2010, Springer Verlag Berlin Heidelberg*, pp 571-578.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

1NF	-	First Normal Form
2NF	-	Second Normal Form
3NF	-	Third Normal Form
4NF	-	Fourth Normal Form
ANSI/IEEE	-	American National Standards Institute/ Institute of Electrical and Electronics Engineers
AS	-	Attributes Sanitization
CD	-	Class Diagram
DBA	-	Database Administrator
DBMS	-	Database Management System
DFD	-	Data Flow Diagram
ERD	-	Entity Relationship Diagram
GUI	-	Graphical User Interface
K	-	Primary Key
KDD	-	Knowledge Discovery in Database
LAS	-	Lecturer Accessing System
MPBP	-	Pengurusan Kompaun Majlis Perbandaran Batu Pahat
OCL	-	Object Constraint Language
OOD	-	Object-oriented Design
OOT	-	Object-oriented Technique
S	-	Sanitization
SMP	-	Student Information System
SPKS	-	Sistem Pengurusan Klinik Sejahtera
SRS	-	System Specification Document
SSADM	-	Structure System Analysis and Design
UInData	-	Attributes Sanitization in Object-oriented Design to Improve Database Structure

UML	-	Unified Modelling Language
UINF	-	User Interface Normal Form
UTHM	-	Universiti Tun Hussein Onn Malaysia

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Living in this modern era is interesting but complex. Since the early 1980’s, when computer gains its popularity, the human social interaction dramatically changes. Furthermore, with the increase in the number of human population in the world, life has become more challenging. People are migrating from their country to other countries for many reasons such as study, work, travel and business. This phenomenon encourages the establishment of good platform for communications through which people can interact to share information, do banking transactions, pay bills and purchase goods. All these involve huge data. The data must always be available and free from errors. Incorrect data will lead to wrong information and inaccurate decision making [1, 2]. The characteristics of good quality information are that the information must be accurate, available and accessible right on time and in right form [3]. To ensure this, good technique in designing database must be well defined. Another reason for a good database design is to increase the speed of accessing data. Bad database design can slow down the data transfer. In a worst condition, the data will not be accessible and no information can be provided. In other words, the database must be precise and concise. Using those applicable systems, there are a series of important data that must be stored for future use. The data can also be used to generate information such as monthly reports, bills and receipts. The place to store all these data is known as database. However, current technique focuses on designing the database structure based on user requirement, but there is no consent whether the data store is used or not.

Before the computer system was introduced, people used files to keep all records. The same thing happens with the filing systems; the data must be well-organised to avoid misplace, error, wasting of time and redundancy. Designing the database is closely related to user needs. It involves a long process and starting to identify when the user requests for the system.

In producing a system or software application, there are four fundamental process activities: software specification, software design and implementation, software validation and software evolution [1-3]. Software specification is a process of writing the user requirements to get system specification [4, 5]. People can use existing standards technique to write documents such as using the natural technique as proposed by Heninger [6], the US Department of Defence and the IEEE. Another technique in writing system documentation that has been discussed by Pohl and Rupp [5] includes the famous and widely used technique, IEEE/ANSI 830-1998. Normally, the System Specification Document (SRS) will be produced as the deliverable for this phase to verify the user requirements with the system developer. However, in an established company, they usually use their own form as the standard way to show the user requirements. These user requirements will be used in the next phase to the software design and implementation. The system design is part of the software fundamental process as illustrated in Figure 1.1.

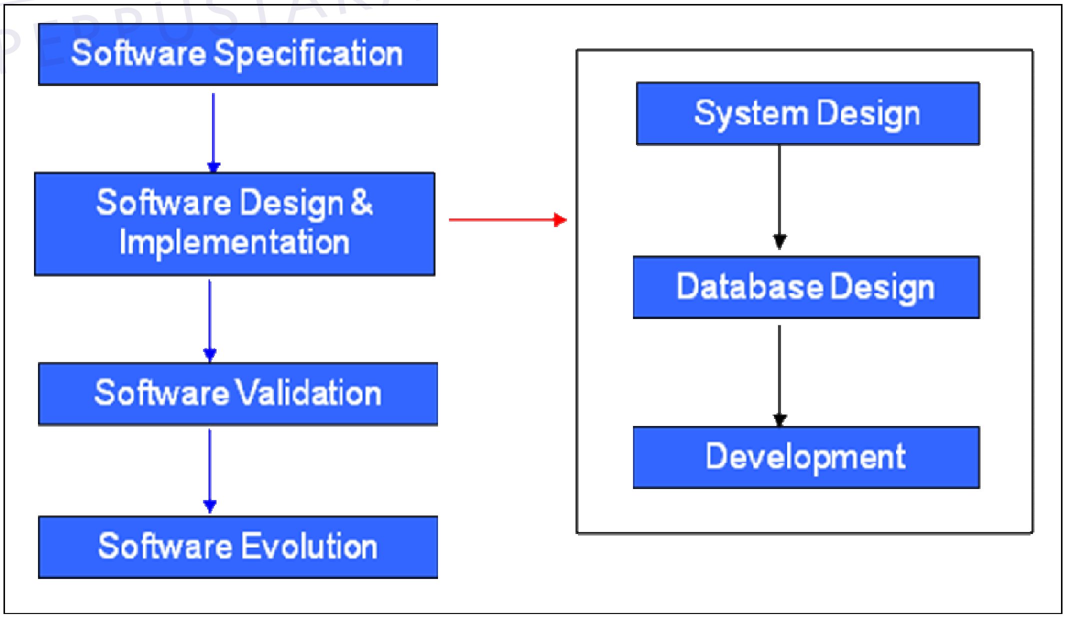

Figure 1.1: System Design: Specify How System Works

After that, the process of collecting the requirements is completed. Modelling or designing a software application in software design and developing phase is an important part in order to make sure that the development is congruent with the specification. A model can specify, visualise and document models of software system including their structure and design. During system design, the software engineer is needed to complete three main tasks, which are designing the system logic, user interface and database. These system logic, user interface and database will be used to counter back with the users before starting the coding. The counter back process assures that business functionality is complete and correct, end-user needs are met and program design supports the requirements for scalability, robustness, security and extendibility. The components involved in system design are illustrated as in Figure 1.2.

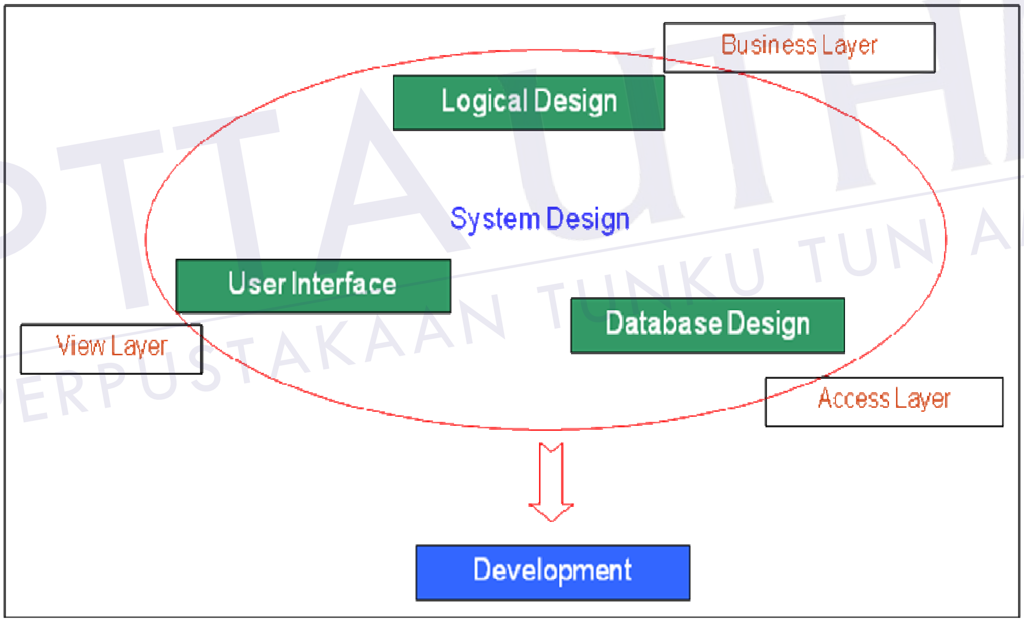

Figure 1.2: The System Design Components

Designing database is very important in system development [7]. It involves storing data and generating information for a long term process. Currently, there are two well-known techniques in designing database, which are the structured technique and the object-oriented technique. Even though the techniques have been introduced for more than 30 years, there are still some issues in designing database. One of the issues is designing optimal tables.

This thesis is a study that defines an alternative approach in designing optimal database system based on object-oriented concept. Optimal database means all defining attributes table in database are used. The discussion is made to present the technique on how attributes sanitization is applied in proposed technique which is not implemented in the current defining database technique, whether using the structure or object-oriented technique. The discussion includes process flow of the proposed technique, rules, example of implementation and the comparisons with current techniques to show that this technique is better than previous techniques.

Normally, in object-oriented approaches, Unified Modelling Language (UML) is used to represent the system requirements that consist of 13 notations such as use case, sequence diagram and class diagram [8-10]. This study has been done to consider all the features and the constraints in the class diagram to be used in producing the first schema table. Attributes sanitization is a technique to remove duplicated class or attributes in constructing the first schema table. Then, this schema table will be compared with the user interface to apply the second step which is to normalize the structure based on user interfaces. Upon completion of the process of normalization in the structure of schema table, the attributes sanitization technique is again applied to remove unused data in producing the final schema table. Then, the schema table is available to create the structure to implement the optimal database. This chapter presents the background of this study including the related issues, aim, research objectives, scope and the limitation of study and lastly the organization of the thesis.

The third process in system design as illustrated in Figure 1.1 is software validation. Software validation was involved with verification and validation processes are used to ensure the development product are confirm to the requirements [11, 12]. The last process in system design is software evaluation. The evaluation is involved with the activities improvement the system after deliver to the used. Both last processes was not involved in this research and not used in verified the attributes sanitization in designing object-oriented to improve database design.

1.2 Problem Statement

System application consists of software design and database design. Software design will show the business process of the system. Software design will do the system logical and sketch the user interface. Another activity is database design to describe the flow of data involving in the system [13]. We cannot deny that software design and database design are tightly coupled. However, until now, both these designs are not congruent. Furthermore, the practitioners have distinguished these two activities in many ways.

In universities, students learn different subjects. System analysis and design subject represents software design. Another subject is database system or database design. This phenomenon also happens in organizations. Most of the companies break up into two positions, which are software engineers or analyst programmers for developing the system and database administrators (DBA) for managing the database. This situation is supported with strong reasons such as for security or to focus on particular jobs that can increase the quality of tasks. However, to break the tasks into two groups, there must be mutual understanding in order to get high quality system.

The situation is more chaotic when the development approaches in software development are broken into two techniques, which are the structured and the object-oriented approaches. The structured approach is also known as functional, procedural and imperative. This approach consists of two techniques, which are Data Flow Diagram (DFD) for software design [13-15] and Entity Relationship Diagram (ERD), which are used for data modelling to assist during the development of the database [16, 17]. The approach was popular in the 1970's. Some programming languages that apply structures or procedures concept are Pascal, C and COBOL. The relational data model concept was introduced by E. F. Codd in 1970 [18]. It is a popular model and easy to implement. Entity Relationship (ER) diagram is a famous model to present the relation of the data [19]. Furthermore, Codd introduced normalization technique to avoid data redundancy [20]. He introduces three techniques of normal form to ensure that the database design is free from anomalies. However, the result of this technique has decomposed into too many tables.

REFERENCES

1. Sommerville, I. *Software Engineering 9 (Ninth edition)*. Eighth. Harlow: Pearson Addison Wesley. 2010.
2. Roth, R. M., Dennis, A. and Wixom, B. H. *System Analysis And Design*. Singapore: John Wiley. 2013.
3. Stair, R. and Reynolds, G. *Principles of Information System*. Tenth Edition. Thomson. 2012.
4. Babu, P. A., Murali, N., Swaminathan, P. and Kumar, C. S. Making Formal Software Specification Easy. *Proceeding of the 2nd International Conference on Reliability, Safety & Hazard (ICRESH-2010)*. IEEE. 2010. pp 511-516.
5. Pohl, K. and Rupp, C. *Requirements Engineering Fundamentals*. First Edition. California: Library of Congress. 2011.
6. Heninger, K. L. Specifying Software Requirement for Complex Systems: New Techniques and Their Applications. *Proceeding of the IEEE Transaction on Software Engineering*. 1980. pp 2-13.
7. Connolly, T. and Begg, C. *Database Systems*. Fifth Edition. Addison Wesley. 2010.
8. OMG. (2009). *OMG Unified Modeling LanguageTM (OMG UML)*. Retrieved on April 2, 2014, from http://www.uml.org/#UMLProfiles.
9. Yan, W. and Du, Y. Research on Reverse Engineering from Formal Models to UML Models. *Proceeding of the 3rd International Symposium on Parallel Architectures, Algorithms and Programming*. IEEE. 2010. pp 406-411.
10. Deenis, A., Wixom, B. H. and Tegarden, D. *System Analysis Design UML Version 2.0*. Fourth Edition. John Willey & Sons. 2012.
11. Cadar, C. and Dadeau, F. Constraints in Software Testing, Verification and Analysis CSTVA 2013. *Proceeding of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*. 2013. pp 208-209.
12. Software & Systems Engineering Standards Commitee (2012). *IEEE Standard for System and Software Verification and Validation.* New York: IEEE Computer Society.

13. Elmasri, R. and Navathe, S. *Fundamentals of Database Systems*. Sixth Edition. Singapore: Pearson. 2011.

14. Ganc, C. and Sarson, T. *System Structured Analysis and Design Tools and Techniques*. New Jersey: Prentice Hall. 1979.

15. Ibrahim, R. and Yen, Y. S. Formalization of The Data Flow Diagram Rules for Consistency Check. *International Journal of Software Engineering & Application (IJSEA)*. 2010. *1(1)*. pp. 99-106.

16. Kendall, K. E. and Kendall, J. E. *System Analysis and Design*. Ninth Edition. New Jersey: Pearson. 2013.

17. Jiang, X. and Song, G. The Database Design for The Control Equipment Management System. *Proceeding of the 2012 3rd International Conference On System Science, Engineering Design and Manufacturing Information (ICSEM)*. IEEE. 2012. pp 261-264.

18. Codd, E. F. A Relational Model for Large Shared Data Banks. *Communication of ACM*. 1970. *13(6)*. pp. 377-387.

19. Chen, P. P. The Entity Relationship Model - Toward a Unified View of Data. *ACM Transaction on Database System*. 1976. *1 (1)*. pp. 9-36.

20. Codd, E. F. (1969). *Derivability, Redundancy and Consistency of Relations in Large Data Banks*. IBM Research Report. Retrieved on 18 January, 2010, from http://technology.amis.nl/ blogacc/blog/wp-content/images/RJ599.pdf.

21. Chonoles, M. J. and Schardt, J. A. *UML 2 For Dummies*. Wiley Publishing, Inc. 2003.

22. Ha, I. and Kang, B. Cross Checking Rules to Improve Consistency between UML Static Diagram and Dynamic Diagram. In. *Intelligent Data Engineering and Automated Learning – IDEAL 2008* Springer Berlin/ Heidelberg. pp 436-443; 2008.

23. Alsaadi, A. Checking Data Integrity via the UML Class Diagram. *Proceeding of the Proceeding of International Conference on Software Engineering Advances (ICSEA 06)*. IEEE Press. 2006. pp 37-46.

24. Fitsilis, P., Gerogiannis, V., Kameas, A. and Pavlides, G. Producing Relation Database Schemata from an Object Oriented Design. *Proceeding of the 20th EUROMICRO Conference, EUROMICRO 94*. 1994. pp 251-257.

25. Grimes, S., Alkadi, G. and Beaubouef, T. Design and Implementation of The Rough Relational Database System. *Journal of Computing in Colleges*. 2009. *24(4)*. pp. 88-95.

26. Waseem, A. A., Hussain, S. J. and Shaikh, Z. A. An Extended Synthesis Algorithm for Relational Database Schema Design. *Proceeding of the International Conference on Information Systems and Design of Communication*. Lisboa, Portugal: ACM. 2013. pp 94-100.

27. Kolahi, S. and Libkin, L. An Information-Theoretic Analysis of Worst-Case Redundancy in Database Design. *ACM Tansactions on Database System*. 2010. *35(1)*. pp. 1-32.

28. Heimel, M. Designing a Database System for Modern Processing Architectures. *Proceeding of the SIGMOD'13 PhD Symposium*. New York, USA: ACM. 2013. pp 13-17.

29. Jason, T. R. *UML: A Beginner Guide*. Califonia: McGraw-Hill. 2003.

30. Boufares, F. and Bennaceur, H. Consistency Problem in ER-Schemas for Database Systems. *Information Sciences*. 2004. *163(4)*. pp. 263-274.

31. Ali, A. B. H., Boufares, F. and Abdellatif, A. Checking Constraints Consistency in UML class diagrams. *Proceeding of the International Conference ICTTA '06*. 2006. pp 3599-3604.

32. Lubcke, A., Koppen, V. and Saake, G. A Decision Model to Select the Optimal Storage Architecture for Relational Databases. *Proceeding of the 2011 Fifth International Conference on Research Challenges in Information Science (RCIS)*. IEEE Conference Publication. 2011. pp 1-11.

33. Sheng, C., Zhang, N., Tao, Y. and Jin, X. Optimal Algorithms for Crawing a Hidden Database in The Web. *Proceedings of The VLDB Endowment*. 2012. *5(11)*. pp. 1112-1123.

34. Al-Zubaidy, H., Lambadaris, L. and Talim, J. Optimal Scheduling in High-Speed Downlink Packet Access Networks. *Transactions on Modeling and Computer Simulation (TOMACS)*. 2010. *21(1)*. pp. 1-27.

35. Rosen, K. H. *Discrete Mathmatics and Its Applications*. Fouth Edition. New York: McGraw Hill. 1999.

36. Grimaldi, R. P. *Discrete and Combinatorial Mathematics*. Fifth Edition. New York: Pearson. 2004.

37. Khaswneh, A. A., Saraureh, K. T. A. and Meridji, K. Early Identification, Specification and Measurement for Data Definition and Database Requirements. *Proceeding of the International Conference on Computer Information and Telecomunication Systems (CITS)*. IEEE. 2012. pp 1-5.

38. Rainer, R. K. and Cegielski, C. G. *Introduction to Information Systems*. Third Edition. New York: John Wiley. 2010.

39. Chen, Z., Gua, T., Qien, Z. and Xiao, C. Research on The Method of Data Extraction Based on Category. *Proceeding of the 12th International Conference on Computer and Information Science (ICIS)*. 2013. pp 365-369.

40. Ismail, S., Shaari, M. S., Saip, M. A. and Shaari, N. *Module Information Technology for Management*. Selangor: Pearson. 2010.

41. Coronel, C., Morris, S. and Rob, P. *Database Principles*. Tenth Edition. China: Course Technology Cengage Learning. 2013.

42. Xu, Y., Yu, Z., Mao, C., Wang, Y. and Guo, J. Entity Answer Extraction of Web Table. *Proceeding of the 2010 Seventh International on Fuzzy Systems and Knowledge Discovery (FSKD 2010)*. 2010. pp 2465-2468.

43. Nakamura, Y., Sakamoto, K., Inoue, K., Washizaki, H. and Fukazawa, Y. Evaluation of Understanability of UML Class Diagrams by Using Word Similarity. *Proceeding of the International Workshop on Software Measurement and The 6th International Conference on Software Process and Product Measurement*. 2011. pp 178-187.

44. Ltd, A. N. (2005). *Data Sanitization Techniques*. White Paper. Retrieved on July 7, 2012, from http://www.qaguild.com/upload/Datasanitization_Whitepaper.pdf.

45. Bishop, M., Bhumiratana, B., Crawford, R. and Levitt, K. How to sanitize data? *Proceeding of the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. 13th IEEE International Workshops on*. 14-16 June 2004. 2004. pp 217-222.

46. Bishop, M., Crawford, R., Bhumiratana, B., Clark, L. and Levitt, K. Some Problems in Sanitizing Network Data. *Proceeding of the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*. 26-28 June 2006. 2006. pp 307-312.

47. Devit, S. Intelligent Data Cleaning Based on Knowledge Based. *Proceeding of the Information and Multimedi Technology (ICIMT 09)*. Jeju Island: IEEE Press. 2009. pp 393-398.

48. Amiri, A. Dare to Share: Protecting Sensentivi Knowledge with Data Sanitization. *Science Direct: Decision Support Systems*. 2006. *43*. pp. 181-191.

49. Molodtsov, D. Soft Set Theory First Results. *Computer and Mathmatics with Applications*. 1999. pp. 19-13.

50. Herawan, T., Rose, A. N. M. and Deris, M. M. Soft Set Theoretic for Dimensionality Reduction. *Communication in Computer and Information Science*. 2009. pp. 171-178.

51. Nguyen, L. G. Metric Based Attribute Reduction in Decision Tables. *Proceeding of the Proceedings of The Federated Conference on Computer Science and Information Systems*. 2012. pp 311-316.

52. Qisheng, X., Zeyin, X., Yang, Z. and Houchang, X. Study on Attribute Reduction Based on Mapping Matrix. *Proceeding of the 2nd International Confrence on Signal Processing Systems (ICSPS)*. 2010. pp 722-726.

53. Ye, M. and Wu, C. Decision Table Decomposition Using Core Attributes Partition for Attribute Reduction. *Proceeding of the The 5th International Conference on Computer Science and Eduction*. Hefei, China: 2010. pp 23-26.

54. Zhou, J., E, X., Li, Y., Liu, Z., Bai, X., Huang, X. and Yang, D. A New Reduction Algorithm Dealing With The Incomplete Information System. *Proceeding of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. 2009. pp 12-19.

55. Ke-yong, Y., Xiao-guang, H. and Fang-qiao, X. An Algorithm for Attribute Reduction Based on Database Technology. *Proceeding of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*. 2009. pp 53-57.

56. Wang, X., Zheng, X. and Xu, Z. Comparative Research of Attribute Reduction Based on The System Entropy and on The Database System. *Proceeding of the International Symposium on Computational Intelligence and Design (ISCID '08)*. 2008. pp 150-153.

57. Yanbo, W., Huiqiang, W., Xiufeng, W. and Ming, Y. Research on Data Attribute Reduction for Network Security Situation. *Proceeding of the 2012 International Conference on Computer Science and Service System (CSSS)*. 2012. pp 589-593.

58. Yourdon, E. and Constantine, L. L. *Structured Design*. New Jersey: Prentice Hall. 1979.

59. Dobing, B. and Parsons, J. Dimensions of UML Diagram Use: A Survey of Practitioners. *Journal of Database Management*. 2008. *19(1)*. pp. 1-18.

60. Maciaszek, L. A. *Requirements Analysis and System Design*. Third Edition. England: Pearson. 2007.

61. Saringat, M. Z., Ibrahim, R. and Ibrahim, N. A Proposal for Constructing Relational Database from Class Diagram. *Computer and Information Science*. 2010. *3(2)*. pp. 38-46.

62. Castro, T. R. d., Souza, S. N. A. d. and Souza, L. S. d. CASE Tool for Object-Relational Database Designs. *Proceeding of the Conference on Information Systems and Technologies (CISTI)*. 2012. pp 1-6.

63. Felsinger, R. and Pleasant, M. (2001). *Using Rational UML Case Tool*. Retrieved on 16 MAY 2011, from http://www.uml.org.cn/ umltools/psf/ UsingRationalRose.pdf.

64. Sybase. (2011). *Object-oriented Modeling*. Retrieved on 16 May 2014, from http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc38086.1610/ doc/pdf/ object_oriented_modeling.pdf.

65. Ali, N. H., Shukur, Z. and Idris, S. A Design of an Assessment System for UML Class Diagram. *Proceeding of the Fifth International Conference on Computational Science and Applications* IEEE Press. 2007. pp 539-546.

66. Kieffer, S., Coyette, A. and Vanderdanckt, J. User Interface Design by Sketching: A Complexity Analysis of Widget Representations. *Proceeding of the EICS'10: Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. New York: ACM. 2010. pp 57-66.

67. Zhu, B., Shanker, G. and Cai, Y. Integrating Data Quality into Decision-Making Process: An Information Visualization Approach. In: S.-V. Lecture Notes in Computer Science 4557. 2007. pp 366-369.

68. Dzafic, I., Soto, J., Halilovic, E., Lecek, N. and Music, M. Object-oriented Database and User Interface Design. *Proceeding of the EuroCon 2013*. Croatia: IEEE. 2013. pp 558-563.

69. Ramez, E. *Database System*. Sixth Edition. Boston: Pearson. 2011.

70. Kumar, V. *101 Design Methods*. New Jersey: John Wiley. 2013.

71. Attraran, H. and Far, A. H. The 2011 10th IEEE International Conference. *Proceeding of the Cybernatic Intelligent Systems (CIS)*. London: IEEE. 2011. pp 128-132.

72. Date, C. J. *An Introduction to Database System*. Eighth Edition. Singapore: Pearson. 2004.

73. Dzafic, I., Soto, J., Halilovic, E., Lecek, N. and Music, M. Object-oriented Database and User Interface Design. *Proceeding of the EuroCom 2013*. Croatia: 2013. pp 558-563.

74. Jagannatha, S., Reddy, S. V. P., Kumar, T. V. S. and Kanth, K. R. Simulation and Analysis of Performance Predicition In Distributed Database Using OO Approach. *Proceeding of the 2013 3rd IEEE International Advance Computing Conference (IACC)*. IEEE. 2013. pp 1324-1329.

75. Osman, R., Awan, I. and Woodward, M. E. Performance Evaluation of Database Designs. *Proceeding of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*. 2010. pp 42-49.

76. Boufares, F. and Bennaceur, H. Consistency Problem in ER-schemas for Database Systems. *Information Sciences: an International Journal*. 2003. pp. 263-274.

77. Ibrahim, N. and Ibrahim, R. Checking Inconsistencies in UML Diagrams. *Proceeding of the Software Engineering Postgraduate Workshop (SEPoW 2009)*. Penang, Malaysia: 2009. pp

78. Ibrahim, N., Ibrahim, R. and Mansor, D. Well-Formedness Rules for UML Use Case Diagram. *Proceeding of the Proceeding of International Industrial Informatics Seminar 2009*. Yogyakarta, Indonesia: 2009. pp 169-173.

79. Ahmad, M. A. and Nadem, A. Consistency Checking of UML Models Using Description Logics. *Proceeding of the 2010 6th International Conference on Emerging Technologis*. 2010. pp 310-315.

80. University of Namur. *Towards a Unifying Conceptual Framework for Inconsistency Management Approaches.* 2009.

81. Snell, M. and Powers, L. *Microsoft Visual Studio 2010*. First Edition. Indiana: Pearson. 2011.

82. Saringat, M. Z. *Lecture Assessment System (LAS): Module Manage Marks and Module Manage Attendance*. Master of Computer Science (Real Time Software Engineering). Universiti Teknologi Malaysia (UTM). 2003.

83. Zakaria, Z. *Sistem Pengurusan Klinik Sejahtera.* Bachelor Degree in Information Technology. Universiti Tun Hussein Onn Malaysia. 2013.

84. Jamaluddin, M. A. W. *Sistem Pengurusan Kompaun Majlis Perbandaran Batu Pahat (MPBP).* Bachelor Degree of Information Technology. Universiti Tun Hussein Onn Malaysia. 2013.

85. Ibrahim, N. *Lecture Assessment System: Module Login and Manage User*. Master of Computer Science (Real Time Software Engineering). Universiti Teknologi Malaysia (UTM). 2003.