International Journal of Management, IT & Engineering
(ISSN: 2249-0558)

CONTENTS

Chief Patron

Patron

Chief Advisors

Editorial Board

Editorial Advisory Board

Associate Editors

October 2011

IJMIE

Volume 1, Issue 5

ISSN: 2249-0558

Title

MEASURING QUANTITATIVE MAINTAINABILITY OF CONCEPTUAL MODEL FOR RE-ENGINEERING PROCESS

Author(s)

Shabana Kausar
Department of Computer Science
University of Agriculture Faisalabad Pakistan

Mr. Ahmed Mateen
Department of Computer Science
University of Agriculture Faisalabad Pakistan

Mr. Ahsan Raza Sattar
Department of Computer Science
University of Agriculture Faisalabad Pakistan

Abstract:

The purpose of this proposed research is to measure quantitative maintainability of the conceptual model. The International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) has defined six external characteristics of the quality assurance; namely, functionality, reliability, usability, maintainability, efficiency, and portability. To measure maintainability of the software, it is necessary to determine first how much the software requires the modification. This research incorporated the case studies of the conceptual model that represents the data of the required software. The conceptual models were converted into improved form by using the schema transformations rules in order to represent the representation at higher level of abstractions. This research determined how much the software requires the maintenance during the re-engineering process. The conceptual models were improved so that reverse engineering and forward engineering can be done easily during the re-engineering process. The proposedresearch was helpful in the planning and analysis phase of the software development for reengineering process to determine the economic, technical, operational, and schedule feasibility of the software.

Introduction:

Conceptual data modeling mean to make a theoretical representation of a company information or person information. In further words we can pronounce that the conceptual data model is used to describe the substantial structure describes the physically present storage structure for the catalog of users. The main idea for using the conceptual model is that it hides the original data or original physical structure and describing these hidden structures as [1]; entities, data types, relations, user operations, user constraints. When we create a data base it consists of the data structure. This data structure represents the data model conceptually because the database requires the data structure. For the creation of database we have to know the data items and the relationships between the data objects and some rules and regulations that will apply on these data objects. A data model actually shows and focuses on the data that for this data what objects are required, what are the relations between the data and what rules and regulations like operations will be performed on the data [1]. The data model that is constructed from the data is not dependent on user constraints. The data that is converted in the data model that used to create

a data base, this data model looks like the original data to user. User imagines that, this is the original data but actually it is the data base that consists of software constraints of data model. The data model works as a connection between the data base and real world applications and processes and it shows the physical representation of data [2].

Entity-relationship model is also called entity relationship diagram. The **(ERD)** model is used for developing higher level metaphors for conceptual data models. Entity relationship diagram presents graphical information for displaying conceptual data models in the structure of entity relationship diagrams. Information system designs used such data models that are elaborated with entity relationship diagrams. An example that can explain the entity relationship diagram is; the model in which some data is going to store in database and some is going to retrieve from the database [3].

Materials and Methods:

The main purpose of the research is to measure the maintainability of the reengineering process as quantities. This measuring process should be done during the software development. The case studies use the schema transformation rules and functional dependencies. When we present software, there is an important role of modifiability and this is used for the improvement of the software. In this research we have used the schemas transformation rules that were described by different researchers for the modification of the model that were presented through case studies [4].

A schema model can be improved by many ways but in this research we have used the restructuring concept. For changing and modifying any model we have to eliminate regular entity, weak entity, relationship of entities [4, 9].

Qualitative method is an effective but time consuming method. In this type of method we have to involve from starting and we have to know each step of method. When the project starts it also starts with the project. When the project is completed we got the solution. During the process we cannot get anything. For using this type of method developers or researchers must have the complete knowledge of the project otherwise they cannot get the correct solution. This method

also has limitations that are researcher should know each step of project that is difficult for every researcher. The second limitation is it is a time consuming method [5].

Quantitative Research is better than the qualitative research and in this type of research we give the numerical values to the solution. These numerical values are 1 or 0. This method or research is used in the last step of the project and statistical methods used in this research. This research requires a deep knowledge of the software because the data that is collected by different researchers is in the form of qualitative [5].

Software reengineering is a concept in which reverse engineering and forward engineering are main points. Reverse engineering and the forward engineering in reverse engineering it is used to recover the design. In this engineering we analyze a program and to represent the program at a high-level abstraction. In this no source code is included it just about the design and its recovery. From this process the documentation of the overall system is created. Before more detailed analysis the overall functionality of the entire system must be understood [6]. Forward engineering is the process of introducing the new rules and regulations in the business applications. Forward engineering requires some software engineering tools; principles, methods and concept. When there is a process of redesigning related to reengineering some challenges are there. Some main challenges that include are, that is to understand the problems in existing system and the limitations that we want to measure. These challenges are in making structural redesign. When we look towards the review of literature some researchers have suggested that the graphical system design techniques should be used to improve the redesign process, these graphical techniques includes Data Flow diagrams. But still maintainability issue is there [6].

This case study is from [8]. The functional dependencies are given in section 3.1 Where the primary key is VistNo, ProviderNo.

Functional Dependencies:

FD1: PatientNo → PatientAge

FD2: PatientNo → PatientCity

FD3: PatientNo → PatientZip

FD4: PatientZip → PatientCity

FD5: VisitNo → PatientNo

FD6: VisitNo → VistDate

FD7: VisitNo → PatientAge

FD8: VisitNo → PatientCity

FD9: VisitNo → PatientZip

FD10: VistNo, ProviderNo → Diagnosis

Consider the following table proposed weights by researchers [4, 7].

Category	Weight
Adding a Regular Entity Type	1.0
Adding a Weak Entity Type	1.5
Adding a Relationship Type	0.5
Changing an Attribute Type	0.5
Changing Structure Constraints (Cardinality or Participation)	0.5
Changing Degree of a Relationship Type	2.0

Table 1: Proposed Weights for a Change

Conceptual Model:

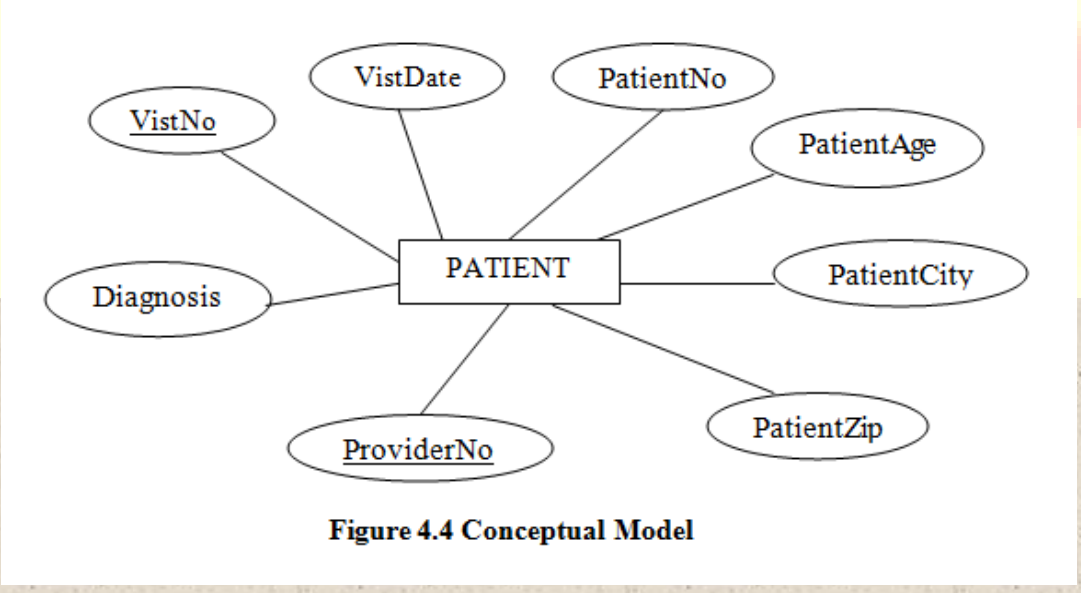

Figure 4.4 Conceptual Model

PATIENT relation [8] is

PATIENT (VistNo, PatientAge, PatientZip, ProviderNo, PatientCity, PatientNo, VistDate, Diagnosis)

4.2.3 Improved Conceptual Model after First Transformation

Transformation of figure 4.4 according to (Hussain *et al*,. 2005).

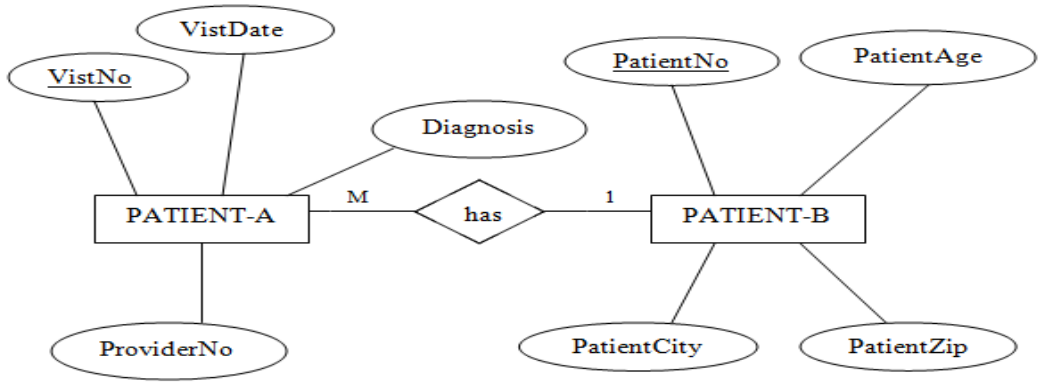

Figure 4.5 Improved Conceptual Model after First Transformation

PATIENT-A relation [8], is

PATIENT-A (VistNo, ProviderNo, PatientNo, VistDate, Diagnosis)

PATIENT-B relation (Michael, 2009) is

PATIENT-B (PatientNo, PatientCity, PatientAge, PatientZip)

4.2.4 Improved Conceptual Model after Second Transformation

Transformation of figure 4.5: according to ((Hussain *et al*., 2007).

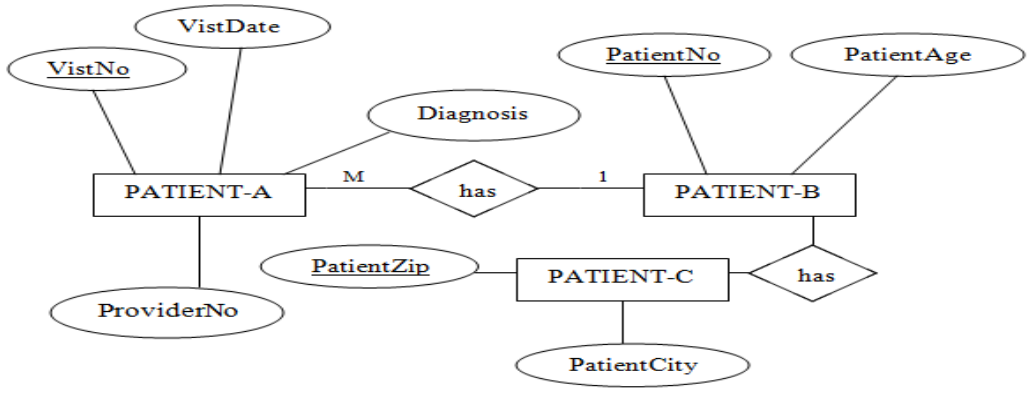

Figure 4.6 Improved Conceptual Model after Second Transformation

PATIENT-A relation [8] is

PATIENT-A (VistNo, PatientNo, ProviderNo, VistDate, Diagnosis)

PATIENT-B relation (Michael, 2009) is

PATIENT-B (PatientNo, PatientAge)

PATIENT-C relation [8] is

PATIENT-C (PatientZip, PatientCity)

C_i	n_i	w_i	$n_i w_i$
Adding Regular Entity Types	**2**	**1.0**	**2**
Adding Weak Entity Types	**0**	**1.5**	**0**
Adding Relationship Types	**2**	**0.5**	**1**
Attributes Moved to Regular ET	**4**	**0.5**	**2**
Attributes Moved to Weak ET	**0**	**0.5**	**0**
ETC			**5**

Table 2: Effort Required for Fig. 4. 4

Therefore Modifiability is 1/5 = 0.200

Maintainability = Modifiability * Number of Schema transformation rules applied

= 0.200 * 2 = 0.4

C_i	n_i	w_i	$n_i w_i$
Adding Regular Entity Types	**1**	**1.0**	**1.0**
Adding Weak Entity Types	**0**	**1.5**	**0**
Adding Relationship Types	**1**	**0.5**	**0.5**
Attributes Moved to Regular ET	**3**	**0.5**	**1.5**
Attributes Moved to Weak ET	**0**	**0.5**	**0**
ETC			**3**

Table 3: Effort Required for Fig. 4.5

Therefore Modifiability is 1/3 = 0.333

Maintainability = Modifiability * Number of Schema transformation rules applied

= 0.333 * 1 = 0.333

Result:

The need of defining quantitative and objective measures to assess the quality of ERD model cannot be over-emphasized. This research proposes a set of easy-to-apply quantitative metrics which can measure the structural complexity of a given ERD model in terms of its two important characteristics are modifiability and maintainability. Modifiability of a ERD model is identified as an important characteristic to discuss understandability.

A metric for modifiability is also defined. The application of these metrics is demonstrated with the help of a scenario and the resulting metrics values are compared to discuss the structural complexity of given conceptual models. From discussing these conceptual models we have made the model as reliable, recoverability, fault tolerance and made its performance level up at stated

conditions. Usability of the performance level is high and from the usability the learning ability, operability and understandability of a user is become high. Efficiency and performance of software become high due to these models and maintainability issues in which analyzing a software, changing a software, testing a software become very easy. Some portability issues also resolved like installation, replacement and adaptability.

An information system of a company or software depend on the quality of its ERD model. We have discussed and elaborated the conceptual model in terms of quantitative measure and show its results with the help of case studies. But when we look towards the literature review a very little work has been done in this way. We have discussed it with functional dependencies and measure into schema model and used them with fuzzy logic. In which it has the membership values between 0 and 1 and then implement through different case studies.

References:

- Jeffrey, A Hoffer, Mary, B. Prescott, and Fred, R Mcfadden, (2002), "Modern Database Management", 6th Ed, Prentice Hall, USA.
- Hussain, T., Shamil, S., and Awais M. "On Measuring Structural Complexity of a Conceptual Model". In Proceedings of the International Conference on Software Engineering, Novosibirsk, Russia, pp 71-75, June 20-24, 2005.
- Assenova, P., Johannesson, P. (1996) "Improving Quality in Conceptual Modelling by the Use of Schema Transformations". ACM SIGMOD 15th Int'l Conf. On Conceptual Modeling 1996, pp. 277-291.
- Hussain, T., Shamail, S., and Awais, M.M., "Applying Fuzzy Logic to Measuring Completeness of a Conceptual Model". In Applied Mathematic and Computation, 185(2), pp. 1078-1086, Publisher Elsevier Science, February 2007.
- http:// wilderdom. Com /research/ Qualitative Versus Quantitative Research. Html
- Sourrouille, J.L. and Caplat G., "A pragmatic view on consistency checking of uml models", in Kuzniarz L., Huzar Z., Reggio G., Sourrouille J. L., Staron M.: Workshop on Consistency Problems in UML-based software development II, Workshop Materials,Research Report, 2003.

- Hussain, T., Shamil, S., and Awais M. "Schema Transformations- A Quality Perspective". In Proceedings of the International Multi-topic Conference, National University of Computer and Emerging Sciences, pp 645-649. Lahore, Pakistan December 24-26, 2004.
- Michael V. Mannino., (2009) "Database Design, Application Development, and Administration", 3rd ED, McGraw-Hill, Inc.
- Hussain, T., Shamil, S., and Awais M. "Measuring Quality of a Conceptual Model through Fuzzy Completeness Index". In Proceedings of the International Conference on Intelligent Computing, (ICIC' 05), Hefei, Chins, August 23-26, 2005.