

A Proposal for Constructing Relational Database from Class Diagram

Mohd Zainuri Saringat

Faculty of Information Technology and Multimedia

Universiti Tun Hussein Onn Malaysia

Parit Raja, Batu Pahat, 86400, Johor, Malaysia

Tel: 60-7-453-8093 E-mail: zainuri@uthm.edu.my

Tutut Herawan

Department of Mathematics Education

Universitas Ahmad Dahlan

Yogyakarta 55166, Indonesia

Tel: 62-2-91-331-7212 E-mail: tutut81@uad.ac.id

Rosziati Ibrahim

Faculty of Information Technology and Multimedia

Universiti Tun Hussein Onn Malaysia

Parit Raja, Batu Pahat, 86400, Johor, Malaysia

Tel: 60-7-453-8001 E-mail: rosziati@uthm.edu.my

This research was supported under the Science Fund Grant Vote S014.

Abstract

Database system is important to ensure the data can be stored, updated and retrieved for future use. Data modelling using the Entity Relationship Model has been introduced more than thirty years. However, designing a good database system is still an attractive issue particularly in system analysis and design because of very hard to do consistency checking between system design and database design. In this paper, a proposal for designing a relational database system based on Object Oriented Analysis and Design is presented. The database system is created by the schema table that extract from class diagram. The rules applying in this paper is following the object oriented concept. It is based on the relationships among the classes, multiplicity, attributes name, class name, data type and the behaviours of the classes. Beside that the user is required to insert record to accomplish a good designing the schema tables to avoid redundancy data. Finally, an automatic editor called CDGeST is proposed in order to automate the process.

Keywords: UML, Class diagram, Object oriented, Schema table

1. Introduction

Nowadays, people interact daily with the system in their life when they are doing banking transaction, bill payment, making telephone calls and others. Using those applicable systems, there are a series of important data that must be stored for future use such as for monthly reports, billing or receipts. The place to store all these data is called database. In the old days before computer system is introduced, people used files to keep the records.

In producing a system or software application, there are four fundamental process activities. Sommerville (Sommerville, 2007) have written that the process activities are software specification, software design and implementation, software validation and software evolution. Software specification is a process writing the user requirements. We can use existing standards technique to write documents such as using natural technique that proposed by Heninger (Heninger, 1980), the US Department of Defense and the IEEE. Davis (Davis, 1993) have discusses some of these and the IEEE/ANSI 830-1998 is the widely used. Normally, the System Specification Document (SRS) will be produce as the deliverable for this phase to verify the user requirements with the system

developer. However, in the establish company, they used own form as the standard way to show the user requirements.

Modelling or designing a software application in software developing phase is an important part in order to make sure that the developing is congruent with the specification. A model can specify, visualize and document models of software system, including their structure and design. It will be use to counter back with the users for the second time before starting the coding. It assures that business functionality is complete and correct, end-user needs are met and program design supports requirements for scalability, robustness, security and extendibility.

Currently, process modelling using object oriented approach is already well known. This technique using Unified Modelling Language (UML) which is consists of various notations such as use case, sequence diagram, class diagram and others. Normally, during system analysis and design, the developer will produce use case diagram, state diagram and sequence diagram to confirm the understanding of the developer with the users regarding the system (Dobing and Parsons, 2008). When the users agreed with the design, the class diagram will be produced to show the interaction among the entities. In class diagram, it will show the behaviours of the class such as generalization, association and multiplicity. It also contains detail about the class such as the attributes, name and data type (Object Management Group, 2008). Practitioners using class diagram for the development team member as their internal design (Jason, 2003). This class diagram will be used as the initial part of designing the database system.

Many tools are available to represent UML notations to design system such as Rational Rose, Together, Poseidon, Power Designer, MagicDraw and so on (Hadj Ali, Boufares, and Abdellatif, 2006). However, neither tool provides facility to generate optimum schema table from the class diagram nor tool have facilities to check consistency from class diagram to schema table. Such as in Rational Rose, the user must indicate which class will be the permanent or transient. In this case, the permanent class will be transfer to table in schema table. Another class will transfer to the template program code such as C++. In this research, we propose an idea to create schema table from class diagram and built an editor to apply the technique. This will helps a novice database designer or who not familiar with database concept. Database is a heart for any system application and designing database is a critical part, even it is not really involves with the end users.

In current practice for software development using object oriented methodology, there are two famous ways on how people establish the database. The first technique is they can transfer the class diagram to the tables with give consideration on the behaviours of the class such as generalization, association, multiplicity, and attributes (Sommerville, 2007; Hadj Ali, Boufares, and Abdellatif, 2006). Mostly people built a database according to the structure of class diagram. The class is referring to the user requirements and classify base on person, thing, event or place. Then the class diagram is used to represent model data with ignoring the method. The UML does not have specific notation for database modelling. However, the class diagram can be used to represent semantic data (Sommerville, 2007).

The second approaches as a famous practice that the software engineer will use difference diagram to present database design. They use object oriented notation to represent the system design such as use case, state diagram, sequence diagram and others. Then, they use the relational data model to build the database design. This concept was introduced by Codd in 1970 (Codd, 1970). It was a popular way and easy to implement. Entity Relationship (ER) diagram was the famous model used to present the relation of the data. Furthermore, Codd has introduced normalization technique to avoid redundancy (Codd, 1970; Codd, 1972).

Both techniques are commonly used by the software developer. However there is no integrity checking from system design to database design. Without consistency checking, it will lead to losing important data during transferring system design to database design. Concerning to this, an automatic tool applies the technique to transfer class diagram into schema table based on criteria in class diagram is proposed here. Using this technique, we can make sure the continuity from system design until creating database.

The rest of the paper is organized as follows; Section 2 presents the related work. Section 3 discusses the framework of the research. We give an example in Section 4. Finally, our research conclusion and future work are in Section 5.

2. Related Works

In system development, gathering user requirements to complete analysis and design phase is the most important before the system engineer dive to coding phase. Design system has divide to two parts which are design of system logical and database logical. Database is contains collection of logically related data. Data is very important in future because it can process to generate information. The information can help people to make

decision and future plan. Major problem in make decision is data not accurate (Zhu, Shanker, and Cai, 2007).

A research done by Hadj Ali *et al.* (Hadj Ali, Boufares, and Abdellatif, 2006) on their paper had reveal types of constraints for well-formed conceptual model to produce database from UML class diagram. They wrote a list of criteria that must be considered during the process to create database. In their paper they stated that Object Constraint Language (OCL) was a part of the UML standards that should be considered to describe constraints on object-oriented model such as class diagram. Even though they had not proposed their idea to producing database, they had contributed to give a comprehensive classification on integrity constraints using UML class diagram.

Alsaadi (Alsaadi, 2006) in his paper wrote that class diagram can be used to produce a schema for system database. He discussed how to check the data structure or data integrities relating to database system using sequence diagram. He suggested a technique using binary class diagram and Object Constraint language (OCL) to ensuring data correctness from class diagram. To show the idea, he gave an example on a sequence diagram derived from the class diagram to ensure data correctness. But he did not show the transferring technique to schema database. He *et al.* (He, Ying, and Zhang, 2002) have done a research for designing and building a multimedia database. They proposed an idea to identify the classes from the user requirements. Then, they converted the class to schema class declarations using Object Definition Language. They used the Object Oriented Design (OOD) as the guide line to build the database. It was done manually. The key was choosing from the attributes in class and applies the principle base on relationship. They followed the name for the table and column as what they were stated in class.

Another research conducted by Ali *et al.* (Ali, Shukur, and Idris, 2007) in their work proposes a tool to check the consistencies of class diagrams. The idea was just to check the name and the numbers of classes, the number of relationship and checking the specification of class object. In their work, they show the technique how to formulate the rules to check the diagram.

Ha and Kang (Ha and Kang, 2008) have discussed the idea to create cross checking to improve consistency between UML static diagram and dynamic diagram. They had proposed base on meta-models for both diagrams. Secondly they describes how to derived consistency checking from meta-models and finally, the derive rules will be evaluated through a case study. Ha and Kang (Ha and Kang, 2008) discussed the rules that are specified formally with Object Constraint Language (OCL).

In process software development using UML specification, requirements analysis and design are usually done using diagrams. Use case diagram is mostly used to specify services of the system (Chonoles and Schardt, 2003). These use cases represent the functionalities of the system. Most often, the system engineer will use another diagrams such as sequence diagram, class diagram and statechart diagram to represent user requirements (Dobing and Parsons, 2008). All these diagrams are developed based on the user perspective since they are the person who is going to use the system.

Dobing and Parsons (Dobing and Parsons, 2008) in their interviews based research with almost 2,700 UML practitioners and clients through the web. From their observed that use case diagrams and class diagrams have highest usage levels followed by sequence, state and activity diagrams. The current work is based on these finding and we decided to use class diagram as our based diagram to extract the data connectivity for producing schema table for a database system. Furthermore the class diagram is used to show entities, element of analysis and design such as associations, aggregations, generalization, inheritance, relationships among entities and others (Chonoles and Schardt, 2003). All the features will be considered when constructing a schema database for a system.

Boufares and Bennaceur (Boufares and Bennaceur, 2003) have written on their paper about consistency problems in cardinality constraints. They have discuss on the problems occurred and wrote a suggestion method to solve it. They mention that they have developed a case tool to realize the idea using C++. However, they not described with details on how the tools is running.

In the previous works, we found that many researchers work on data integrity or quality in UML diagram especially in designing system (Hadj Ali, Boufares, and Abdellatif, 2006; Alsaadi, 2006; Ha and Kang, 2008; Zhu, Shanker, and Cai, 2007; Boufares and Bennaceur, 2003). A few of them discussed the idea of extracting manually schema table from class diagram using OCL (Hadj Ali, Boufares, and Abdellatif, 2006; He, Ying, and Zhang, 2002). However in our research, we propose an alternative way to transfer UML diagram to schema database using automatic tool based on constraints in class diagram and domain data. Furthermore, this technique has continuity from system design to database design for checking consistency of data flow. Our automatic tool is called CDGeST (Class Diagram Generator for Schema Table).

3. The Framework of CDGeST

Our research is to provide an alternative way on how to design optimum database system. We divide the research into two phases. The aim in the first phase is to produce the schema tables. Then the schema tables will be used as the input for normalize process in second phase. However, in this paper we just discuss the first phase only.

During the process gathering user requirements, the system engineer will produce UML diagrams such as class diagram to get user requirements. The diagrams will be used to verify the user needs. In this research, we will focus on the class diagram as the prominent UML diagram. It will be used as the input with considering the constraints during creating schema table. At the end of this research, we aim to have an editor to draw class diagram and auto generate tool to transfer class diagram to schema table. This tool is called CDGeST.

The framework is divided into three components which is the input, the algorithm and the output as the final result. The users for this tool are required to draw a system class diagram. After the drawing is completed, the tool will check the basic rule for class diagram. If the rules are fulfilled, the tool will check the class diagram constraints and domain data for producing the schema tables as the output. The architecture of this framework has been illustrated in Figure 1, where it shows the process to generate schema table.

<<INSERT FIGURE 1>>

3.1 Input

The input for this research framework is a class diagram. The user will draw class diagram using this editor. In the diagram there are eight types of rules to be checked to produce schema table. These rules have been details up as below.

Step 1. The class must have a unique name. It is not allow having more than one class with same name. The name must be in one word and space is not allowed between the words. It can be combination of a few meaning words. However, the name must not be more than thirty characters.

Step 2. Relationships in class diagram are defined using associations and links. It is used to indicate the relationship between classes in class diagram.

Step 3. The class must have more than one attribute. In class, each attribute must have a unique name with not more than thirty characters. The name must be in one word. A space is not allowed along the name.

Step 4. The class must have an identifier to distinguish among the object. It can be from one attribute or combination of a few attributes. The user must state the identifier.

Step 5. Superclass and subclass are the concept on how to present the real nature. There are three concepts of superclass and subclass. First, both classes totally have same attributes. Second, a class having same attributes but not inherits all attributes from superclass. Third, the class inherits a few attributes from the superclass and adds some other attributes.

Step 6. Multiplicity is continuity of relationship. There are three forms of multiplicity which are one to one, one to many and many to many. However in one relationship there must be having one form of multiplicity.

Step 7. Date type is used to define the attributes. Each attribute must has one data type. It is used to indicate the type of data that will be hold by the attribute. It can be string, number, float, alphanumeric or date.

Step 8. The user is requested to insert a few records in each class and not more than ten records.

General rules will be checked to verify the semantic rules of class diagram. However in one class diagram, not all rules will apply concurrently depending on the user requirements and the system needs.

3.2 The Proposed Algorithm

The algorithm for transferring the class diagram to schema tables is based on the eight types of rules that have been categorized in the input for easy to understand and traceability. It is described in Figure 2.

<<INSERT FIGURE 2>>

3.3 Output

After verifying all constraints, the tool will produce schema table in form of Structure Query Language (SQL) in a text format. The file contents with command to create table, attributes, key, data type and relevant information to create table for a system. Then the text file can be imported by other editor such as MySQL or Oracle editor for generated tables.

4. Result and Discussion

In this section, we present an example of an application for a simple student registration subject. The system requirements are shown in Figure 3.

<<INSERT FIGURE 3>>

Step 1. We do the analysis to the requirements. From the requirements we have classify the information into three types of classes to hold the attributes which are student, subject and lecturer.

Step 2. Draw the class diagram to represent the user requirements as in Figure 4 using the editor.

<<INSERT FIGURE 4>>

Step 3. After drawing class diagram, the user is required to insert a few records based on attributes from class diagram. The example of the records inserted is shown as in Figure 5.

<<INSERT FIGURE 5>>

Step 4. The editor will create a link list to store the sample data information from Figure 5.

Step 5. The editor extract the information base on class information, attributes, relationship and multiplicity such as in Figure 6. All the information will be stored in another link list.

<<INSERT FIGURE 6>>

Step 6. The editor will check the naming of classes, attributes, multiplicity and generalization. After that, the records are evaluated to reduce redundancy. Finally, the editor will generate the schema table as shown in Figure 7.

<<INSERT FIGURE 7>>

In our example we show the flow of data gathered from user requirements. Then the user requirements are presented in class diagram. After that, the steps showed how to extract schema table from class diagram.

CDGeST is a tool that is able to generate schema table automatically according to the class diagram. The classes are transformed into tables. The transformation is based on the rule above. However, the number of tables may not be equal with the number of the classes in the class diagram. In certain condition, the number of tables may be the same as well as the number of classes, but it can be less or more. This situation really depends on the structure involved in the class diagram.

5. Conclusions and Future Works

User requirements are very important in developing system. Object oriented is the well known methodology and using UML to show the user requirements. There are thirteen diagrams are available in UML diagram. However, none of the diagram is available to present the database design. With this motivation, in this paper, we have proposed CDGeST as the automatic tool of transferring class diagram to schema table. This tool will check the consistency of data from system design to database design based on the constraints in class diagram and behaviours of data. It can avoid missing data in database design and reduce data duplicated.

In object oriented concept, class is referring to generalize a thing such as animal, person, or a system. It may contain many attributes in one class. Transferring class diagram to schema table is not base on the number of attributes. The tool will not be rearranging the number of attributes. This limitation will be used for future work in second phase in the research for finding the optimum table based on user needs.

References

- Ali, N.H., Shukur, Z., and Idris, S. (2007). *A Design of an Assessment System for UML Class Diagram*. Proceeding of 5th International Conference Computational Science and its Applications. IEEE Press, 539–546.
- Alsaadi, A. (2006). *Checking Data Integrity via the UML Class Diagram*. Proceeding of International Conference on Software Engineering Advances (ICSEA 06). IEEE Press, 37–46.
- Boufares, F., and Bennaceur, H. (2003). *Consistency Problem in ER-schemas for Database Systems*. Information Sciences, 263–274.
- Chonoles, M.J., and Schardt, J.A. (2003). *UML 2 for Dummies*. Indiana: Wiley.
- Codd, E. F. (1970). *A Relational Model for Large Shared Data Banks*. Communication of ACM, 13 (6), 377–387.

- Codd, E. F. (1972). *Further Normalization of the Data Base Relational Model*. Data base System, Englewood Cliffs, N.J.:Prentics Hall, 33–64.
- Davis, A.M. (1993). *Software Requirements: Objects, Functions and States*. Englewood Cliffs, NJ: Prentice Hall.
- Dobing, B., and Parsons, J. (2008). Dimensions of UML Diagram Use: A survey of Practitioners. *Journal of Database Management*, 19, 1–18.
- Ha, I., and Kang, B. (2008). *Cross checking Rules to Improve Consistency between UML Static Diagram and Dynamic Diagram*. *Lecture Notes in Computer Science 5326*, Springer-Verlag, 436–443.
- Hadj Ali, A.B., Boufares, F., and Abdellatif, A. (2006). *Checking Constraints Consistency in UML class diagrams*. *Proceeding of International Conference ICTTA '06*, 2, 3599–3604.
- He, Y.L., Ying, Z.L., and Zhang, Y.W. (2002). *The Design and Implementation of a Multimedia Database with Object Oriented Analysis and Design Method*. *Proceeding of 6th International Conference 9 (ICSP '02)*. IEEE Press, 1035-1038.
- Heninger, K.L. (1980). *Specifying Software Requirements for Complex Systems: New Techniques and Their Applications*. *IEEE Transaction on Software Engineering*, 6 (1), 2–13.
- Jason, T.R. (2003). *UML: A Beginner Guide*. McGraw-Hill, California.
- Object Management Group. (2008). *Introduction to OMG's Unified Modeling Language (UML)* http://www.omg.org/gettingstarted/what_is_uml.htm
- Sommerville, I. (2007). *Software Engineering*, Eighth Edition. Pearson, Singapore.
- Zhu, B., Shanker, G. and Cai, Y. (2007). *Integrating Data Quality into Decision-Making Process: An Information Visualization Approach*. *Lecture Notes in Computer Science 4557*, Springer-Verlag, 366–369.

Figures

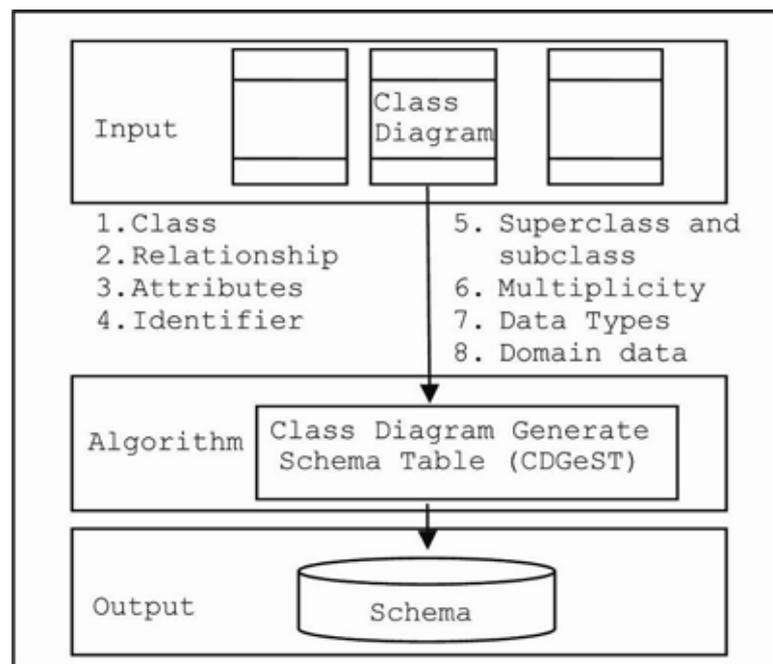


Figure 1. Framework of Auto Generate Tool for UML Class Diagram

The algorithm for transferring class diagram to schema table.

- 1 Start.
- 2 Read class diagram.
 - 2.1 Extract the class information.
 - 2.2 Extract the attributes of each class.
 - 2.3 Extract the relationship of each class.
 - 2.4 Extract the multiplicity for each links.
 - 2.5 Extract the domain data for each class.
- 3 Generate two dynamic link list to store class constraints and data.
- 4 Go through the class information to avoid duplicate naming classes.
- 5 Go through the link list for detecting attributes.
- 6 Go through all nodes in link list for diagnosing the superclass and subclass.
- 7 Go through the link list for checking multiplicity.
- 8 Go through the link list for checking data type.
- 9 Go through the data link list for checking redundancy data.
- 10 Read from first Node to end Node for generating schema table.
- 11 End

Figure 2. The Proposed Algorithm

The system has requirements as the lists below.

- The faculty offers a number of subjects.
- The students can register any subject from the subjects offered.
- Each subject will be taught by one lecturer.
- The student will get their marks and gred for each subject.
- Less then ten records must be inserted in each class to show the behavior of the instances.

Figure 3. User Requirements

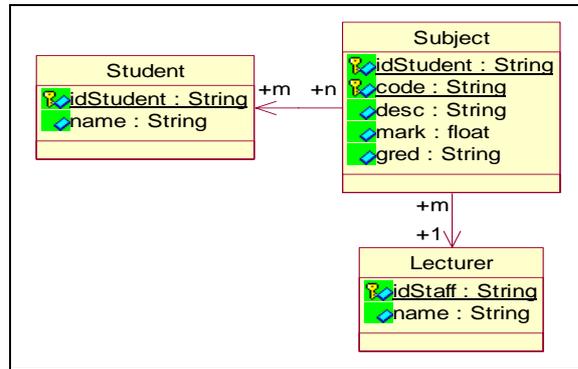


Figure 4. Class Diagram for Student Registration Subject

idStudent	name			
CI08001	Muaz Ismail			
CI08234	Teo Seng Tee			
DC08103	Firhan Bakar			

idStudent	code	desc	mark	gred
CI08001	BIT1083	Database	80	A-
DC08103	BIT1083	Database	60	C+
CI08001	BIT1003	Cobol	70	B
CI08234	BIT1083	Database	50	C
CI08234	BIT1003	Cobol	85	A

idStaff	Name
00218	Afandi Bakar
00567	Zaki Salikon
00786	Maznah Ali

Figure 5. Data Information

```

Student, ATTRIBUTE, START,
idStudent, string, 7, PK, name,
string, 50, END

Student, RELATIONSHIP, START,
Subject, Student, END

Student, MULTIPLICITY, START, M, M,
END

Student, GENERALIZATION, START,END

Subject, ATTRIBUTE, START, code,
string, 7, PK, desc, string, 50,
mark, float, 3, 2 , gred, string,
2, END

Subject, RELATIONSHIP, START,
Student, Subject, END

Subject, MULTIPLICITY, START, M, M,
END

Subject, GENERALIZATION, START,END

Lecture, ATTRIBUTE, START, idSaff,
string, 5, PK, name, string, 50,
END

Lecture, RELATIONSHIP, START,
Subject, Lecture, END

Lecture, MULTIPLICITY, START, M, 1,
END

Lecture, GENERALIZATION, START, END

```

Figure 6. Information Extraction

```

Create table Student
(idStudent varchar(7),
name varchar(50),
Primary Key (idStudent));

Create table Subject
(idStudent varchar(7),
code varchar(7),
mark float(3,2),
gred string(2),
idStaff string(5),
Primary Key (idStudent,code));

Create table Student1
(code varchar(7),
desc varchar(50),
Primary Key (code);
code varchar(7)

Create table Lecturer
(idStaff varchar(5),
name varchar(50),
Primary Key (idStaff));

```

Figure 7. Schema Table