2014 IEEE Sixth International Conference on Technology for Education

Equivalence of Problems in Problem Based e-Learning of Database

Abu Sayed Md. Latiful Hoque*, Golam Md. Muradul Bashir†, Md. Rasel Uddin*
* Dept. of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
asmlatifulhoque@cse.buet.ac.bd, rrasel@gmail.com
† Dept. of Computer and Communication Engineering, Patuakhali Science and Technology University, Patuakhali, Bangladesh
murad98csekuet@yahoo.com

***Abstract*—Problem-based learning (PBL) is a way to learn what is needed to solve a problem, how can a solution be obtained quickly, precisely and professionally. To achieve the goal of problem-based learning, problem design and assign same level of problems among the students are important in engineering classroom environment. SQL is a major part in Database course. In problem based e-Learning of SQL, it is essential to find out the equivalence of an SQL problems to assign the set of problems to a set of students. This is necessary for equal judgment of the performance of individual students. We have developed a complexity model to find out the equivalence of problems for Problem based e-learning of database. In this model, complexity of problems is found by parsing the given solution of the problem in top down approach. We have applied our model to well-known SQL Learning and Evaluation System (SQL-LES). We have compared our calculated complexity value with the complexity value in the question bank of SQL-LES assigned by the SQL experts and found that in most cases our model generate similar complexity value as SQL-LES. Application of our model will reduce the instructor workload in SQL-LES.**

Keywords*—*Problem Based e-Learning (PBeL), SQL-LES, SQL.

I. INTRODUCTION

Problem-based learning (PBL) is an instructional method in which students learn through facilitated problem solving. In PBL, teacher fetches some problems and assign similar level of problems among the students. It is necessary to assign similar problems to different groups of students where grading of students is necessary and in that case the problems should be equivalent. The equivalence of problems means that the complexity and effort to solve the problem is within a specified boundary. The boundary value depends on the problem domain and teaching policy. So it is necessary to develop a complexity model to find out complexity value that can be used to identify problem equivalence. Existing problem-based learning focus on the PBL methodology, learning content, learning environment, problem design, e-learning for PBL, tools to submit report, discussion forum, problem design, monitor student activity and others. Existing systems do not focus on the complexity of the problem and distribution of the equivalent problems among the students.

SQL-LES is a Problem Based e-Learning and e-Evaluation system developed in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology[1]. The complexity values of SQL problems is assigned manually based on domain knowledge of the instructors. If the class size is large and multiple instructors produce multiple assignments then it is difficult to have an equivalence of assignments. Students performance sometime varies because of the dissimilarities of the assignments given by different instructors. At the same time, as the SQL question bank contains hundreds of questions, it is extremely difficult to obtain a global complexity value of each SQL problem to reuse the problems.

The objectives of this research are to: (1) design a complexity model to find out complexity value of an SQL problem, (2) apply the proposed complexity model on Problem Based e-Learning of Database and (3) evaluate the performance of the model by applying the model in SQL-LES. We have proposed a complexity model that decomposes the problem in top-down fashion to find out the required domain knowledge. Through top-down analysis, the problem has been divided into sub-problems. Then each sub-problem is analyzed to find out critical, meaningful and complex items to solve the problem. All discovered items are then marked with some weighted values. Total complexity value has been calculated by using different mathematical models.

II. LITERATURE REVIEW

Hmelo-Silver [2] described what and how student learn in Problem-based learning (PBL). PBL approach has been applied in a course Advance Software Engineering in engineering education by[3]. Non-traditional teaching method was introduced for inexperienced students to understand in software engineering classroom by Ita et al. [4] using PBL. Lian et al. [5] has developed e-Learning system for problem-based education. Design of problems, analyses of solutions submitted by the student groups and how learning objectives were achieved are given by Mantriet et al. [6]. They have shown that the cooperative groups perform better than independent individuals on a wide range of problems. R. Laughlin et al. [7] described the effectiveness of group size on intellective problems. P. Lai et al. The effect of quality assurance system on the implementation of PBL teaching strategy to courses has been studied by [8]. Fontes et al. [9] introduced multi agents system to support problem-based learning. According to this approach, four types of agents are proposed: a Problem Detector Agent (PDAg), a Student Agent (SAg), an Animated Interface Agent, and Work Group Creation Agents (WCAg). Web-based environment to implement problem-based learning has been developed by [10]. Kazuo et al. [11] developed a new design engineering educational framework using an e-learning system called ShareFast, Semantic Web-based software for document management system with workflow. Web base learn-

978-1-4799-6489-5/14 $31.00 © 2014 IEEE
DOI 10.1109/T4E.2014.23

IEEE computer society

ing environment to support problem-based learning has been developed by Eleni et al. [12]. In this approach, students and instructors use the web as a virtual place to collaborate and create new knowledge and new educational experiences. Research works are found in PBL design, development for different courses and classroom environment but not for equivalence of problems. In the field of NLP, IR or Artificial Intelligence, the researchers are interested in computational complexity and many works are found in the literature. We are interested in cognitive complexity of a problem in PBeL.

III. Complexity Model: System Architecture and Analysis

Problem is the core element in Problem-based Learning. By solving the practical problems, the learners can explore the concept and principles behind the issues, developing their self-learning ability, and implement the meaningful construction of the knowledge. In PBL, different levels of problems are distributed among the student to solve. It is important to distribute similar level of problem between different students. We have determined the levels of the problems depending on the complexity values of the problems.

A. Complexity Model

Problem complexity depends on how much domain knowledge requires solving a problem. To find out used domain knowledge, complexity model divides problems into sub-problems. Then each sub-problem is analyzed to find out critical, meaningful and complex item to solve the problem. Then we assign complexity value to each item based on item type and position of use in SQL statement. We have used top-down method to analyze problems. Fig. 1 shows the top down decomposition of a problem.

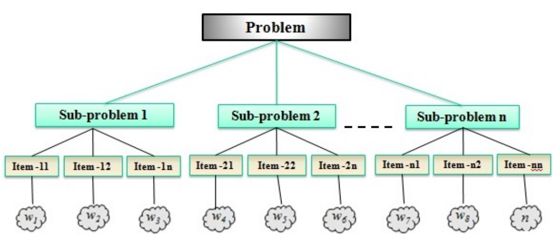

Fig. 1. Top-down Analysis of PBL Problem.

B. Top Down Analysis of SQL Operations

In general, all SQL operations can be broadly classified into the categories like SELECT, CREATE, INSERT, UPDATE, DELETE. SQL is a special purpose programming language to manipulate data in Relational Database Management System. The most common operation in SQL is the query, which is performed with the declarative SELECT statement. SELECT retrieves data from one or more tables, or expressions. SELECT statement has following clauses with many selections options, parameters and keywords as follows:

- FROM CLAUSE -indicate data source from which data to be retrieved
- WHERE CLAUSE -uses to specify which data is to be retrieved
- GROUP BY CLAUSE -groups data to apply aggregate function
- HAVING CLAUSE -uses with Group By clause to filter groups
- ORDER BY CLAUSE -identifies which columns are used to sort the resulting data

General Format of SQL SELECT Statement:

> SELECT [ALL — DISTINCT] column1[,column2] FROM table1[,table2 — Sub Query] [WHERE ”conditions”] [GROUP BY ”column-list”] [HAVING ”conditions] [ORDER BY ”column-list” [ASC — DESC]]

Complexity of a given problem depends on how many database clauses have used with options and parameters. Top-down Analysis of SQL SELECT statement is given in Fig. 2. Each clause can use functions, predicates, columns or expression to make it meaningful. The main purpose of database clauses is to prepare data for user with desire shape.

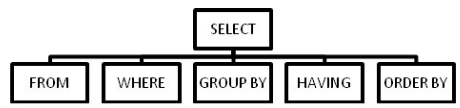

Fig. 2. Top-down Analysis of SELECT Statement

The detailed Tree Structure of SQL-SELECT Statement is given in Fig. 3. In the same way we can represent the tree structure of create, update, insert, and delete statement.

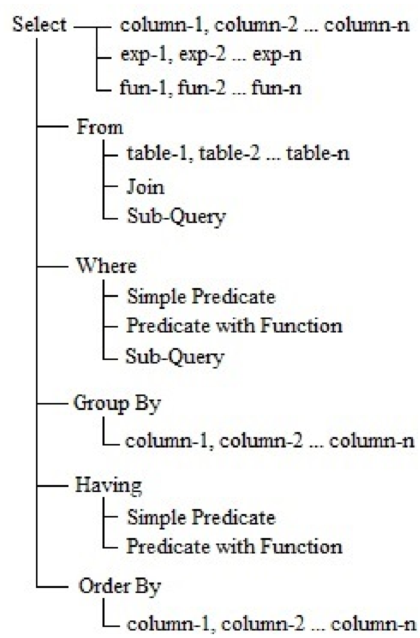

Fig. 3. Tree Structure of SQL SELECT Statement

C. Level of Used Database Clauses

SELECT statement has five clauses. Each clause can use function, column name, predicate or sub-query as parameter. Same type of parameter can be used with different clauses and there are different meanings. So the complexity of the used parameter depends on the type and position of use in SQL statement.

D. Complexity Value of Database Clauses with SELECT Statement

Complexity Values of Database Clauses with SELECT Statement are shown in Table I.

TABLE I. COMPLEXITY VALUE OF SQL ITEMS WITH DIFFERENT CLAUSES

Type of the SQL Clause	Used in Level	Complexity Value
Functions	SELECT	Fv
	Where	
	Having	
Columns	SELECT	Cv
	Group By	
	Order By	
Tables	From	Tv
Predicate	SELECT	Pv
	Where	
	Having	
Expression	SELECT	Ev
	From	

E. Complexity Value of SQL SELECT Operation (C_P)

Complexity value of a given SQL problem has been calculated by analyzing SQL statement. SQL statement is a combination of different types of clauses. Database clauses can use lots of keywords, functions, expressions and predicates to make the SQL query more purposeful and efficient. So SQL complexity depends on used clauses and parameters. Database operations are the key items in database programming. The complexity value Cp of any SQL operation can be defined as

$$\text{Complexity Value}=\sum_{i=1}^{n} \sum_{j=1}^{m_i} W_{ij}\times \log_2(1+K_{ij})$$

Let n = number of sub-problems in a problem, k = number of similar item in sub-problem, j = item type, wi = complexity value of i type item. Using the above equation and Table I, the complexity value of SELECT operation,

$$C_{SELECT}=CL_V+T_V+\sum_{i=1}^{|S|} A_i |\ A_i \epsilon\ S \Lambda\ S \subseteq \{F_V, C_V, P_V, E_V\}$$

Where, CL_V is complexity value of usages of SELECT clauses in the expression, F_v is functional value, C_v is columns value, T_v is tables value, P_v is predicates value, E_v is expression value.

Complexity Value of Column (C_v): Table has a specified number of columns. Most of the time user does not need all column value. In this situation, user has to mention desired column in SQL query. Some aggregate function also depends on column which has to define by user. So column has many roles in SQL query. Column value (C_v), depends on the number of used column and the level where it is used. We have calculated column value in the following way:

$$C_v= C_{ws}\times\log_2(1+m)+C_{wo}\times\log_2(1+n)$$

Where, C_{ws} is the column weight with SELECT clause, cwO is the column weight with other clauses like GROUP BY and ORDER BY clause; m is the number of used column with SELECT clause, n is the number of used column with GROUP BY and/or ORDER BY clause. Table II shows the complexity values of column (C_v) for different number of columns for C_{ws}=1 and C_{wo}=2.

TABLE II. COMPLEXITY VALUE OF COLUMN WITH DIFFERENT CLAUSES

Number of Columns	Used with	Column Weight cwS / cwO	Complexity Value (CV)
2	SELECT	1	1.585
4			2.322
6			2.807
2	GROUP BY/ ORDER BY	2	4.755
4			6.966
6			8.421

In such way we can measure the complexity value of function, table, predicate and expression.

F. Equivalence of Problems

The equivalence of problems means that the complexity to solve the problem is within a specified range. Those problems fall into a specified range are all equivalent problems and any of the problem can be assigned to any student and seem to have equal judgement. Let us consider a problem domain D and P_1, P_2 ... Pn are problems in domain D with complexities C_1, C_2 ... C_n. The problem P_1 will be equivalent to P_2 iff

$$C_1\sim C_2\leq \epsilon \bigwedge\{P_1P_2\}\epsilon\ D$$

Where, ϵ is the allowable range in the specific level of problem. The value of can vary based on the nature of problems, how many levels we want to divide the problem for a particular domain and what is the difference between the lowest and the highest complexity value. Table III shows an example of equivalence of problem.

TABLE III. EQUIVALENCE OF PROBLEMS USING BOUNDARY VALUE

Problem No	Complexity Value (CP)	Problem Level	Boundary Value (CP $\pm$ ϵ)
1	23	Level-1	25$\pm$2
2	23		
3	26		
4	29	Level-1	30$\pm$2
5	30		
6	28		

IV. COMPARISON RESULT WITH EXISTING SQL-LES SYSTEMS

We have applied the complexity model in the problem bank of existing SQL-LES. The complexity value in SQL-LES was given by the instructors during the conduction of classes using SQL-LES.To calculate complexity value using our model, we have collected complexity value of individual database clause, function, predicate and others from three Database specialists. Table IV shows details about the parameter values collected from different database specialists. We have used the average value to find the complexity value.

We have considered four cases and have changed the parameter values using collected parameter value from first, second and third database specialist. The result is shown in Table V. We have calculated complexity value using parameter

TABLE IV. COMPLEXITY VALUE (CV) FOR INDIVIDUAL DATABASE ITEM

SQL Operators	Level in Use	CV By DB Expert 1	CV By DB Expert 2	CV By DB Expert 3	Average CV
Create		3	4	3	3.3
Insert		3	3	3	3
Update		3	3	3	3
Delete		3	3	1	2.3
Select		3	3	2	2.6
Where		3	3	4	3.3
Group By	Select	3	4	3	3.3
Order By		3	2	5	3.3
Having		3	4	3	3.3
Table	From	3	3	2	3.6

TABLE V. : COMPARING COMPLEXITY VALUE WITH EXISTING SQL-LES SYSTEMS BY CHANGING PARAMETER VALUES

Test Case	Total Problem	No. of Similar Problems	No. of Dissimilar Problems	% of Similarity	% of Dissimilarity
Case1	60	46	14	76.6%	23.3%
Case2	60	46	14	76.6%	23.3%
Case3	60	45	15	75%	25%
Case4	60	50	10	83.3%	18.6%

value collected from three database experts and shown in case 1, 2 and 3. Case-4 is the average of three cases. Predicate is the most sensitive parameter in SQL statements. We have changed formula for the most sensitive parameter using different parameter values from database specialists. Test cases one, two, three and four have shown complexity value by changing formula for sensitive parameter. This model defines the equivalence of problems maximum 91.6 % of similarity compared to manually defined equivalence of problems. The result is shown in Table VI.

TABLE VI. : COMPARING COMPLEXITY VALUE WITH EXISTING SQL-LES SYSTEMS BY CHANGING PARAMETER VALUES AND FORMULA

Test Case	Total Problem	No. of Similar Problems	No. of Dissimilar Problems	% of Similarity	% of Dissimilarity
Case1	60	44	16	73.3%	26.6%
Case2	60	46	14	76.6%	23.4%
Case3	60	49	11	81.6%	18.3%
Case4	60	55	5	91.6%	8.3%

V. CONCLUSION AND FUTURE RESEARCH

Problems are the main elements in problem-based learning. Student will learn through analyzing the assigned problem. To evaluate students performance in PBL session, it is important to assign equivalent problems among the students. Complexity model will be very helpful to define equivalence of problems by analyzing complexity value. In Problem-based Learning and Evaluation of SQL, students are assigned multiple assignments with varying complexity. Existing SQL-LES assign the complexity values of SQL problems manually based on domain knowledge of the instructors. If the class size is large, multiple instructors produce multiple assignments then it is difficult to have an equivalence of assignments. Students performance sometime varies because of the dissimilarities of the assignments given by different instructors. At the same time, as the SQL question bank contains hundreds of questions, it is extremely difficult to obtain a global complexity value for each SQL problem to reuse the problems. We have developed a complexity model to find out the equivalence of problems using the complexity value of SQL problems. To calculate the complexity value of an SQL problem, we have analyzed the problem in top-down fashion to find out the complexity of the problem. We have applied our proposed complexity model on existing problem based SQL-LES question bank and found comparable result. This model defined equivalence of problems a maximum of 91.6% of similarity compared to manually defined equivalence of problems. The minimum similarity is found to be 73%. Our model behaves similar to the existing SQL-LES. In the present system, different instructors can assign different complexity values of the same problem. This will affect the student performance. The use of the complexity model will result uniform complexity values for all students. Manual assignment of complexity values increases the teacher workload. The application of our model will reduce the teacher workload in problem setting. In this work, we have found out the equivalence of SQL problems by analyzing the solution of the problem. The same methodology can be used in problem based learning of other courses of engineering education.

REFERENCES

[1] A. S. M. L. Hoque, M. M. Islam, M. I. Hossain, and M. F. Ahmed, "Problem-based e-learning and eval-uation system for database design and programming in sql," *International Journal of E-Education, E-Business, E-Management and E-Learning-IC4E*, pp. 537–542, 2013.

[2] C. E. Hmelo-Silver, "Problem-based learning: What and how do students learn?" *Educational Psychology Review*, vol. 16, no. 3, pp. 235–266, 2004.

[3] M. Qiu and L. Chen, "A problem-based learning approach to teaching an advanced software engineering course," in *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, vol. 3. IEEE, 2010, pp. 252–255.

[4] I. Richardson and Y. Delaney, "Problem based learning in the software engineering classroom," in *Software Engineering Education and Training, 2009. CSEET'09. 22nd Conference on*. IEEE, 2009, pp. 174–181.

[5] L. Qiu and C. K. Riesbeck, "Designing web-based interactive learning environments for problem-based learning," in *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*. IEEE, 2005, pp. 333–337.

[6] A. Mantri, S. Dutt, J. Gupta, and M. Chitkara, "Designing problems for problem-based learning courses in analogue electronics: Cognitive and pedagogical issues," *Australasian Journal of Engineering Education*, vol. 14, no. 2, pp. 33–42, 2008.

[7] P. R. Laughlin, E. C. Hatch, J. S. Silver, and L. Boh, "Groups perform better than the best individuals on letters-to-numbers problems: effects of group size." *Journal of Personality and social Psychology*, vol. 90, no. 4, pp. 644–651, 2006.

[8] P. LAI, K. C. C. TANG, and ???, "Constraints affecting the implementation of problem-based learning (pbl) strategy in university courses," 1999.

[9] L. M. d. O. Fontes, F. M. M. Neto, A. A. Pontes *et al.*, "A multiagent system to support problem-based learning," *Creative Education*, vol. 2, no. 05, p. 452, 2011.

[10] H.-P. Yueh and W.-J. Lin, "Developing a web-based environment in supporting students team-working and learning in a problem-based learning approach," in *Creating, Connecting and Collaborating through Computing, 2005. C5 2005. Third International Conference on*. IEEE, 2005, pp. 145–149.

[11] K. Hiekata, H. Yamato, P. Rojanakamolsan, and W. Oishi, "A framework for design engineering education with workflow-based e-learning system," *Journal of software*, vol. 2, no. 4, pp. 88–95, 2007.

[12] E. Kaldoudi, P. Bamidis, M. Papaioakeim, and V. Vargemezis, "Problem-based learning via web 2.0 technologies," in *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*. IEEE, 2008, pp. 391–396.