Review Report on Estimating the Back-End Cost of Business Software using ER diagram Artifact

SWAYAMPRAVA PARIDA
School of Computer Engineering
KIIT University
Bhubaneswar, India
swayamprava89@gmail.com

SAMARESH MISHRA
School of Computer Engineering
KIIT University
Bhubaneswar, India
samaresh2@gmail.com

Abstract- Accurate and timely cost estimation is an important software project management activity for a successful software development project. At the present time of business software development, the back-end development plays a very vital role along with in front-end development. The back-end cost of business software is generally depends on the database size and the structural size of individual data items. As software cost estimation is done in the early phase of software development, so the artifacts available at the conceptual data modelling using ER diagram along with the concepts of generalization, specialization and aggregation has considered for size estimation of database system of software. In this paper, we have studied some existing models and metrics commonly used for estimating back-end size, its complexity and development effort.

Keywords- Entity size, Relationship size, Complexity, Artifacts, cost estimation, ER model

I. INTRODUCTION

The accurate and timely estimation of software development cost is a vital and important activity of software development projects [1]. The development cost of software is largely depended on a number of key factors associated with software and identified during the early phase of its development. Nowadays, most business software systems uses huge amount of data resulted from organizations day to day business operations. The challenges for practitioners focus more on data management in terms of proper generation, storage and retrieval of data rather than on application program development. Again, it has been observed from industry that the relational database systems are becoming more popular for managing data in database part of business data-intensive software systems. For estimating cost of software, it has been felt to estimate the cost of data part and application part of software separately. Many established estimation techniques are targeted to application part of software system and are scarcely applicable to data part of software [1][2][3]. These techniques of estimation uses the inputs from the requirements analysis and specification phase of software development processes for predicting software size, its complexity and effort of development. Subsequently, the artifacts of design phase are considered for getting more accurate estimation. In back-end cost estimation models, many published research works have considered the artifacts of ER (Entity Relationship) model and extended ER model, which are used popularly as a part of conceptual design tool for back-end development of software [1][5][6]. This back-end cost has been categorized as the cost of size, complexity, efforts and development time. The development effort estimation of back-end part of software is generally depends on back-end size and its complexity. The back-end complexity [1][5][6] is generally dominated by its structural complexity and database complexity, where the structural complexity represents the number of entities, number of relationships, and the number of attributes of a ER model and the database complexity represents the relationships of mapping cardinality, the depth of referential integrity constraints, the depth of attribute inheritance. For the software cost estimation, various models and techniques are available in the literature and are useful for different kind of projects [1][2][3][4]. These techniques have been reported to achieve with varying degrees of success in accurate estimation of cost. This paper has reviewed a number models and metrics available and practiced by practitioners for estimating back-end size, its complexity and development effort.

This paper has been organized as follows: In section-II, we have presented the review of literature, and in section-III, we have highlighted the comparison among various models and metrics, in Section-IV, we have summarizes the discussion. Finally, in section-V, we have given our conclusion and future work.

II. REVIEW OF LITERATURE

A number of back-end estimation models, techniques and metrics have been proposed by different researchers to estimate database size, its complexity, effort of development, and quality of database structure. Since relational database systems are popularly used in database development of data-centric software system, therefore the artifacts of ER diagram and its extension are primarily considered as input for predicting back-end cost. This section discusses a number of models, metrics and their outcomes available in the literature.

Geoffery J. Kennedy,1996[3] has proposed three elementary data structures in order to describe a systematic treatment of ER diagram. The elementary structures includes (1) the *hierarchy structure,* considers the relation between child and parent nodes of ER diagram, (2) the *many-to-many* structure, which represents the many-to-many relationships among an associative entity with real world entity and (3) lastly the *parallel linkage structure,* which represents the multiple occurrences, reflecting the association among principal entity with dependent entity. The work estimates the effort by considering the factors such as entity, attribute, and relationship captured in an ER diagram.

This work of Geoffery J. Kennedy has proposed an algorithm for calculating the total number of occurrences of all the elementary data structure of an ER diagram and was empirically validated by calculating total effort, C1 as per equation (1):

$$C1=904 - 3.05\,C2 - 51.0\,C3 + 28.7\,C4 - 33.8\,C5 + 72.4\,C6 \quad (1)$$

Where the C2 represents number of attributes, C3 represents number of entities and C4, C5, C6 represents the numbers of each of the elementary structures present respectively.

Yaun Zhao et al., 2003[1] has proposed model for estimating database development cost by considered the ER diagram artifacts such as entities, attributes, relationships and path complexity. The work was validated empirically using following multiple linear regression equation (2):

$$Y = 23.01+ 3.80NOE +1.03NOR + 0.01NOA + 0.49NOP \quad (2)$$

Where "NOE" represent number of entities, "NOR" represents total number of relationships, "NOA" represents total number of attributes and "NOP" represents total number of paths.

Here they estimated software cost using adjacency matrix generator, path complexity generator, metric generating tool, statistic module. The complexity metric, named as path complexity was proposed to find the number of paths and their length for accessing other entities. For getting path complexity, the work has first generated the corresponding graph from the given ER diagram and then using the proposed algorithm, *Search Path,* the number paths are measured.

Bushra Jamil et al., 2010[2] has proposed a set of relational database metrics for effort estimation of relational database applications by considering five relational database factors such as relationship, queries, tables, reports and forms.

Here the authors have proposed an effort estimation model by considering a new technique called as statistical techniques. This techniques generally based on linear regression and generalized linear model, which are helpful for analyzing linear and non linear effects of independent variable on dependent variable. Here they have proposed a set of metrics based on relational database which includes factors like Number Tables, number of Field, Total number of Key(primary and foreign key), Total number of Relationship(One to One relationships and One-to-Many relationships), Total number of Queries(Nested Select Queries and Simple Select Queries), data entry Forms, Dialog Boxes, sub and customized Forms, Total number of Reports(simple, expression and custom report) for effort estimation.

Those metrics were validated empirically using following equations (3) (4) (5) (6) (7) (8) by [2]:

Effort= 2.94 + 149.3 * (Number of Tables) – 5.53* (Number of Fields) + 72.29 * (Total Primary Keys)+1.03*(Total Foreign Keys) - 218.1 * (Total relationships) - 79 * (One to One Relationships) + 98.9 * (Total Queries) + 4.0 * (Nested Select Queries)-137.6 * (Simple Select Queries) + 4.0 * (Nested Select Queries) (3)

Effort = -7.33 + 2.673 * (No of Tables) -1.6 * (total relationships) + 2.5 *(Total Queries) + 1.14*(Total Relationship)–2.27*(Total Forms)–7.33*(Total Reports) (4)

Effort= 2.94 - 0.032 *(Number of Fields) + 2.90 * (Number of Primary Keys) - 2.62* (Number of Foreign Keys) (5)

Effort = 4.24 + 3.23*(Number of One to One relationships) + 0.007 * (Number One-to-Many relationships) (6)

Effort= -0.61 + 3.62 * (Nested Select Queries) -1.07 * (Simple Select Queries) (7)

Effort = 0.96 + 1.08 *(Data Entry Forms) -5.86 * (Dialog Boxes) + 4.79 *(Sub forms) + 3.0 * (Switchboards) + 1.23 * (Customized forms) (8)

The DSER (Database size estimation based on ER diagram) model [5] is based on estimation of back end cost of business software by using ER diagram. This DSER model identified the essential factors of an ER (and its extension EER) diagram by using data modelling of a system. This model proposed a technique for estimating the size of back-end cost based upon their complexity. That technique depends on various kinds of size metrics for measuring the size of entity, size of relationship and size of semantic integrity constraint, which was helpful for estimating the total development effort by using COCOMO model. In this DSER model structural complexity is taken to be consider which is based on database size. To make the estimation in the earlier phase here the conceptual modelling phase is considered.

This DSER model consist of different DSER metric for measuring the Total database size of the entity, Relationship and Semantic integrity constraint. The work was validated empirically using following equation (9) by [5]:

1) Total size of *Entity* calculated by $\sum_{j=1}^{NE}(size\ of\ entity\ set)j$

2) Total size of *Relationship* calculated by $\sum_{K=1}^{NR}(size\ relationship\ set)k$

3) Total size of *semantic integrity constraints* calculated by $\sum_{k=1}^{j} LOC$

Total size of database = Total size of Entity + Total size of Relationship +
Total size of semantic integrity constraints (9)

Where "NE" represents number of entity, "NR" represents the number of relationship and "LOC" represents the Line of code.

The BEC (Back end complexity) model [6] based on Early Estimation of software by using back-end. The main objective of this model was to estimate the effort by considering the level of complexity. This level of complexity is of different types. Those are entity level, attribute level and the relationship level. Which are helpful for estimating the size and the number of Line of code during the implementation stage of a business software.

Here the authors considered the factors like entity, relationship, business constraint which comes under the BEC (Back end complexity) metric. In which the result of output shows that the developmental effort, the total complexity of entity and its total database complexity are directly proportional to each other. Those metrics were validated empirically using following equations (10):

1) Total complexity of *entity* calculated by $\sum_{1}^{NOE} EC$

2) Total complexity of *Relationship* calculated by $\sum_{i=1}^{k} RC$

3) Total Business constraint calculated by $\sum_{K=1}^{NOC} CK$

Total Complexity of database = TEC+TRC+TSC (10)

Where "TEC" represents Total Entity complexity, "TRC" represents Total Relationship and "TEC" represents Total Business constraint.

Tharwon Arnuphaptrairong, 2013[9] working on his paper reported that Function point analysis is very much necessary for software cost and effort estimation. In which he considered Data flow diagram for solving the timing critical problem. This data flow diagram gives the information which can be helps us for software effort estimation. Function point was introduced by Albrecht which is based on the functionality of the software delivered. This functionality of software delivered is the size of the software represented by Line of code. It means the more is the functions delivered, the more is the Line of codes.

The size of the functionality is measured by Function point (FP).This FP accepted with a lot of variables from both the practitioner as well as from the academics. The research occur in this area is known as Functional point analysis (FPA).This FPA consists of five function types. Those are external input, external output, external quires, Internal Logical File and External Interface File. These function types are helpful for accessing the complexity. This FP can be calculated by (11):

FP= UFP × TCF (11)

Where the UFP is the Unadjusted Function Points and the TCF is the technical complexity factor.

III. COMPARISION TABLE

SL. NO.	MODEL	METRICS & FACTORS	WHAT TO ESTIMATE	VALIDITY
1.	YuanZhao's Model [1]	Entity Relationship Attribute Complexity Effort	1.effort 2.complexity	Empirical validated
2.	Bushra Jamil's Model [2]	Entity Relationship Attribute Tables Queries Different kind of forms	1.estimate effort	Empirical validated
3.	Kennedy's Model [3]	Entity Relationship Attribute	1.estimate effort	Empirical validated
4.	DSER model [5]	Different attributes like multi valued or derived attribute Entity set Relationship set Generalization Specialization	1. Entity Size 2. Relationship size 3. semantic integrity constraint size 4. database size	Empirical validated
5.	BEC Model [6]	Total entity complexity Total relationship complexity Weight of attribute Relationship point Technical factor Generalization Specialization	1.Entity complexity 2.Relation complexity 3.Estimate effort	Empirical validated
6.	Tharwon Arnuphaptrairong's Model [9]	Functional point External Input and Output External Query Technical factor	1.Estimate effort	Empirical validated

IV. DISCUSSION

After review of literature on estimating the Backend cost we concluded that there are some more additional artifacts available such as lattice, categorization, and complex relationships which may be considered for more accurate cost estimation. In some of the papers authors used some cost estimation techniques based on their Structural complexity which includes the concepts like Generalization, Specialization and Aggregation. But they did not consider any concept to estimate the cost in Conceptual modeling phase. So here our main objective is to introduce a new concept which can easily remove all these inconsistencies. That new concept is called as "Categorization". By the help of this Categorization we can easily represent an intra-entity class relationship by considering more number of super classes. Also in some papers cost estimation methods like LOC, FP, DC, Path complexity are used, but they ignore about an extension of FP-like approach, named "Class Point", which was conceived to estimate the size of object-oriented products. For the early estimation most of the papers generally use the ER model, but they did not consider the deletion of an entity. If we want to delete an instance of an entity type then we use this "deletion constraint". It is helpful for describe business rules that define a specific action. So our motivation is to design a model by considering all these works to archive out broad objective.

V. CONCLUSION

After study of few papers related to Back-end cost estimation by using ER diagram artifacts we got that Back-end cost estimation is a cost driven platform. Various models and techniques are used to estimate the actual cost of software. Recently Back-end cost estimation of business software by using ER diagram artifacts is a biggest challenge for software developer to estimate the cost from the requirement analysis and specification phase of software development process. As this Back-end cost estimation offers a range of economic benefits to their users so now days the effect of this cost estimation increases day by day. This paper summarizes how the Back-end cost estimation occurs in business software by using ER diagram artifacts and how it will be more efficient to calculate the actual Back-end cost of different business software with the help of proposed models and techniques.

REFERENCES

[1] Y. Zhao, H. B. K. Tan, W. Zhang, “Software Cost Estimation through Conceptual Requirement”, Proceedings of the Third International Conference on Quality Software (QSIC’03), IEEE, 2003.

[2] B. Jamil, J. Ferzund, A. Batool, S. Ghafoor, "Empirical Validation of Relational Database Metrics for Effort Estimation", 6th International Conference on Network Computing, IEEE, May 11-13, 2010, pp.1-5.

[3] G. J. Kennedy, “Elementary structures in entity-relationship diagrams: a new metric for effort estimation”, IEEE, 1996, pp. 86-92.

[4] I. R. Khan, A. Alam, H. Anwar, "Efficient Software Cost Estimation using Neuro-Fuzzy Technique ",International Symposium on Computer Engineering & Technology,ISBN:978-81-910304-0-2, 2010.

[5] S. Mishra, R. Mall," Estimation of effort based on back-end size of business software using ER model", World Congress on Information and Communication Technologies, IEEE, Dec 11-14, 2011, pp.1098-1103.

[6] S. Mishra, P. K. Pattnaik, R. Mall, “Early Estimation of Back-End Software Development Effort”, International Journal of Computer Applications, Vol. 33, No.2, November 2011 , pp. 6- 11.

[7] Capers Jones, Estimating Software Costs, “Bringing Realism to estimation”, 2nd Edition, TMH 2007.

[8] Narayan S. Umanath, Richard W. Scamell, “Data Modelling and Database Design” Thomson India Edition 2007.

[9] T. Arnuphaptrairong," Early Stage Software Effort Estimation Using Function Point Analysis: Empirical Evidence", IMECS, March 13 - 15, 2013, Hong Kong.

[10] A. Ali , S. Qadri , S. S. Muhammad , J. A. M. T. Pervaiz ,S. Awan,"Software Cost Estimationthrough Entity Relationship Model",2010.