Informal Aggregation Technique for Software Engineering Experiments

Babatunde K. Olorisade

Mathematical and Computer Sciences Department, Fountain University
Osogbo, Osun State, Nigeria

Abstract

One of the bases for development and standardization in any discipline is continuous empirical verification of knowledge. Thus, empirical replication is required to validate or establish experimental results. When these replications occur, there is also a need to synthesize the different results for a common result. Software Engineering experiments usually fall short of the requirements of the meta-analysis techniques currently in use for this purpose either in number, report or even design. Therefore, there is a need for a less rigorous technique that could serve even as a temporary tool but suitable to software engineering studies and also usable. This study presents an alternative less rigorous aggregation techniques that systematically synthesize the statistic ANOVA results by grouping treatments with seemingly equal level of evidence together. Each group is then ranked on an ordinal scale, grouped and interpreted.

Keywords: *Empirical software engineering, aggregation technique, informal meta-analysis, SE experiments*

1. Introduction

Software engineering, like other engineering fields, needs to formalize, standardize, create uniformity and have certain level of predictable functionality as well as accuracy knowledge of most of its tools, methods and procedures. In order to achieve this, researchers are aiming for extensive and exhaustive empirical research in all areas – testing techniques, review techniques, programming paradigms etc., to underpin software engineering [1, 2], since one of the basis for development in any science or engineering discipline is empirical verification of knowledge [3-5]. Empirical study education (theory and practical) as it applies to software engineering is growing among researchers [6], consequently, the discipline is witnessing increasingly more comprehensive studies conducted on more realistic programs and processes [7]. Software engineering researchers and practitioners are now taking advantage of empirical research, to validate their findings and work. Continuous experimentation and most importantly replication is required to validate or establish experimental results in a discipline like Software Engineering that still needs to underpin most of its practices and techniques.

Nothing widely applicable can be concluded from the results of a single experiment, several replications of such experiments are required for a meaningful deduction [6, 8, 9]. Thus, when researchers perform replications of experiments, there is always a need to combine (aggregate) the results, not only to see similarity or differences but to abstract a common (global) result representative of all the experiments. This type of combination either increases (or reduces) confidence in the individual results or quantifies effect size therefore making the result more exact. It can also reveal essential areas or questions yet to be adequately addressed in past studies [2].

Aggregation (in SE terms) is synthesizing – organizing, summarizing and generalizing [6] the results of multiple experiments to generate pieces of knowledge or evidence that can become facts or used in real world software development. Meta analysis is still the most widely used aggregation technique in SE but it is not always the case that sufficient number of experiments is available to apply a formal aggregation technique (meta-analysis) because of existing variations in the design and execution of the experiments [6, 10]. In fact, it may take decades in Software Engineering which is still emerging (evolving) as an engineering discipline before ample experiments are available to underpin a concept. Yet, software engineering researchers crave to have a global view of existing replications of experiment on a certain subject; sufficient number or not for formal aggregation, in order to be able to say something about their own field. Also, previous attempts to use informal aggregation approach have yielded limited results [10].

There is currently no formal aggregation technique that considers SE's special circumstances and engineering maturity level. This makes SE experiments lack in one requirement or the other to be a good fit for statistical meta-analysis techniques. For example, Jedlitschka et al [6] observed that meta-analysis disregarded experimental context in its process but context is vital to SE.

Therefore, there is a need for a less rigorous technique that takes the youngness of SE into account in judging and collating its experiments. Most especially, when replications are few or experiments do not pass meta-analysis techniques' fitness tests. Yet, the technique must be meaningful and follow a systematic and repeatable procedure. A method like this will increase the available evidence level on the performance of different SE tools and techniques.

This work proposes a qualitative informal aggregation technique that could serve the purpose when a formal aggregation technique is not yet applicable for a reliable result. The approach discussed here is not a substitute for a formal aggregation technique, but rather a less rigorous, applicable and methodic means of aggregating SE experiment results to get a clue at general sense of direction before a formal technique is actually applicable.

Section 2 of this article describes the background of the study while section 3 presents the tools and techniques. The proposed technique is explained in section 4. Section 5 discusses the validation steps taken so far for the technique and section 6 presents the conclusion and future direction of work.

2. Current Practice

Software engineering is still emerging as an engineering field, thus, there is a continual pressing demand to entrench undisputable facts (laws and theories) as in other engineering fields. Therefore, experiments are aggregated every now and then but with meta-analysis even when we know that SE experiments usually do not fully satisfy all the necessary pre-conditions of the technique [6, 8]. Though, there are other statistical techniques for the same purpose [11-13], meta-analysis is still the most sophisticated [14, 15]. The result of meta-analysis rely on the homogeneity of the experiments involved [10]. This will ensure that all the experiments were taken into consideration before the result was produced. Some researchers [9, 16, 17] have defended the use of meta-analysis in SE studies. Enrique [10] stated the major obstacles to its application in SE as:

- Inadequate number of experiments, replications and homogeneity among the studies.
- Non existence or application of experiment reporting standards.
- Wide ranging measure of quality.
- Non-standardization of response variables.

Aside meta-analysis, other techniques mentioned in [11-13] like vote counting and comparative analysis that are less complicated with reduced constraints may also be applicable but the extent of their application have not been extensively studied [10] and are scarcely applied in SE [18, 19].

The fact remains, from time to time, researchers will want to know which side available studies are tilting. So, rather than settle for any less stringent technique that is not convincingly applicable, then it may be a good moment for a less rigorous technique that evolve from SE studies, that took into account the peculiarities of SE studies and immaturity of the field itself. Such technique may be useful at measuring the state of available study results before a more rigorous meta-analysis is applicable.

Some works have been done in this area [10, 20], suggesting some form of alternative aggregation techniques for SE experiments whenever meta-analysis is found inapplicable. Fernandez [10] proposed an Aggregation Process with multiple evidence levels which still relies on the statistical techniques but choose the most appropriate at any point in time and Oivo [20] also put forward what he called a goal-oriented aggregation of empirical results. This work is also in the direction of proposing another alternative; the result of any of the techniques and others to come found reliable enough and stood the test of time may even replace meta-analysis in SE over time.

3. Tools and Techniques

For the purpose of this study, an extensive work was done on identifying relevant existing knowledge in order to be equipped with appropriate tools and techniques useful for developing this research. They are as follows:

3.1 Tools:

- SE aggregation techniques: Acquire knowledge form formal and less formal aggregation techniques currently in use in SE.
- Replicated Experiments: Set of replicated experiments (8), conducted to study the efficiency and effectiveness of code review by abstraction, decision coverage and equivalence class partition as software evaluation techniques.
- Related Experiments: Also, we gather experiments that have been conducted on similar subjects that are not exact replications.

3.2 Technique

The activities that will be followed to propose the technique is through these steps:

- Extract abstract knowledge for the statistical techniques

- Decide on applicability condition: Determine when this method is more suitable that the more reliable meta-analysis techniques.
- Establish rules of application: Ahead of method application, some issues like identifying the subject of the different experiments and deciding on their relevancy; choose tolerable error level etc. should take place.
- Abstract possible similarities or difference among the different experiment results based on the tolerated error level
- Deduce interpretation from the outcome based on the emerged classification pattern.

4. Proposed Technique

This informal aggregation technique (using qualitative deduction approach) was a product of a research meant to aggregate results of some existing SE experiments without using any meta-analysis technique but unfortunately, no other existing systematic method to perform this task was found at the time. The, approach was created to systematically synthesize SE experiment results. It is not meant for now, to substitute meta analysis technique. It is meant to be applied only when it has been established that the necessary pre-conditions for applying the formal techniques have not been met. But, in a situation where some of the experiments to be aggregated meet the necessary conditions for applying meta-analysis while some does not, we advice that meta-analysis be applied to those ones even though, they may be few. This method then be applied to all the experiments again, this will give us the opportunity to compare the output of this method to that of a more rigorous technique.

Also, this method is applicable when what we need is a running global result of a set of continuous experiment replications before there are sufficient experiments to apply meta-analysis.

The proposed method – informal aggregation, is divided into four steps:

- Extraction: The primary purpose of this step is to present a combined result of all significant effects in the analysis. This is achieved by extracting the significance value of all the treatments from the various ANOVA tables and present them in one table.
- Classification: At this stage, the intention is to study the table from step 1 and characterize any noticeable pattern. The patterns are classified using some code (e.g., alphabets).
- Classification Ranking: Each class is ranked at this stage based on the homogeneity of the results in each coded class.
- Deduction: The aim here is to study each category and qualitatively deduce evidence from each ranked class.

4.1 The Extraction Step

The main purpose of this step is to present the ANOVA result (significance) of all the experiments in a single table. At this level, it is important that all the experiments have the same number of treatments (main and interaction effects). If all the experiments were not analyzed using the same value for the confidence level, then the researcher needs to make a choice out of two options:

i. Flexible combination: Accommodate the different confidence levels as used. For example, if one experiment used 90% and the other 95% confidence limits. The researcher will apply these two levels to all the experiment and extract significance values that fall within both levels. However, it is advisable to make a distinction between which values were accommodated for which confidence level. This approach can be viewed as downgrading.

ii. Strict combination: The researcher decides to maintain the higher confidence level; therefore he will only extract treatment values that satisfy this condition or better re-analyze the affected experiment. This is more or less an upgrading approach.

So, the main idea of this step is to extract the significance values that satisfy the researcher's confidence criteria from the different experiment analysis results and present them in a table.

Table 1 shows the layout presenting the outcome of applying this step to the experiments of this study. It is advisable to use contrasting color codes to stress the significance level of a treatment for each experiment. For example, if a flexible combination is used, the treatments that are found significant at 99% may be written in black, 95% grey and green to indicate not significant.

4.2 The Classification Step

The classification step basically looks at the available significance options and assigns unique code to each distinct possible combination of values. The possible distinct combination (y) is usually $2^x – 1$, (where x is the number of accommodated confidence level + 1).

Table 1: Extracted ANOVA values for all experiment treatments

Notes	Treatments	Significance			
		Exp 1	Exp 2	…	Exp n
Corrected model (sig/power)					
Model used: Type III Sum of Squares Significance level: 0.01 and 0.05	Treatment 1	0.001	…	…	…
	Treatment 2	…	0.02	…	…
	…………	…	…	…	0.357
	Treatment n	…	…	…	…

For example, in this study, tolerate both 90% and 95% confidence level, as decided by the researcher. Then, x = 3 and thus, y = $2^3 – 1$ = 7. The deducted 1 is usually an impossible situation.

In this step, it is helpful to create a table of 'x' columns and 'y' rows, excluding the header row and code column. The headings will be the different confidence levels and the "*Not significant*" option. The cells will then be filled logically (coded) with 0s and 1s (Yes/No or True/False). Each column will afterwards be coded, say alphabetically, to distinguish them from each other. This step resulted in the creation of a classification table below:

Table 2: Code creation based on combination of different possibilities

Code	Significant at 0.01	Significant at 0.05	Not significant
A	No	No	Yes
B	No	Yes	No
C	No	Yes	Yes
D	Yes	No	No
E	Yes	No	Yes
F	Yes	Yes	No
G	Yes	Yes	Yes

4.3 The Classification Ranking

After the classification, the next step is to rank (numerically) the classification table based on the defined strength or clarity of the knowledge presented by the combination in each column (the row entry). This step becomes tricky, most especially when using the flexible combination. It is usually of three sub steps:

- **Assignation**: Assigning numerical ranks to the different codes. The rank is a positional denotation of the clarity of evidence deducible from the combination of different experiments. Here, some decisions have to be made upfront, concerning the interpretation of the significance values. For example, we need to decide on the weight of contribution of a treatment found significant at say 95% and 99% confidence limit. The higher the limit, the stronger the evidence presented by the result. Such a decision will have to be taken between all the confidence levels tolerated and a relationship must be established between them i.e., if one is stronger than the other. The result of experiment studies from this step is presentable as shown in Table 3. From table 3, the lower the rank, the clearer the message deducible from the pattern presented. The rank is an indication of clarity of evidence inferable from the collection of experiments on each treatment.
- **Annotation**: The summarized ANOVA table produced in step 1 is then interpreted with corresponding codes and ranks. The usefulness of the codes becomes more pronounced in situation where certain treatments have the same rank qualification but different codes. The code will tell us what combination of values lead to the rank.
- **Streamlining**: After the annotation step, it will be possible to make an intermediate decision table, which will show for each treatment, how many experiments fell under each confidence level. This will enable the selection of treatments that have significant effect across all the experiments. For example, in this study, we proposed that an ordinal decision scale ranging from *significant, significance tendency, not significant* or *ambiguous* or based on evidence with *very clear, clear, somehow clear* or *unclear* be used. Consequently, a treatment with all or more than 75% of the experiments not significant is tagged as *not significant*, therefore, the null hypothesis is generally considered rejected. A treatment with 50% significant and 50% not significant is tagged as *ambiguous*, while something stronger is either classified as *significant* with strong knowledge evidence or tagged as having tendency to be significant. Concentration can thus be shifted to those believed to have varying degree of effects across all the experiments (Table 4). That is, the not clear, significant and the significant tendency classes.

Table 3: Ranking of the various codes

Code	Significant at 0.01	Significant at 0.05	Not significant	Rank
A	No	No	Yes	1
B	No	Yes	No	3
C	No	Yes	Yes	4
D	Yes	No	No	1
E	Yes	No	Yes	4
F	Yes	Yes	No	2
G	Yes	Yes	Yes	4

Table 4: Status of each treatment (all experiments)

Code	Rank	Treatment	Numbers of experiments significant at:			Status
			0.01	0.05	Not significant at both	
A	1	Treatment 1	0/z	0/z	z/z	Not significant
	4	Treatment x	0/z	0/z	z/z	Not significant
C	1	Treatment y	0/z	2/z	6/z	Not significant
	4	Treatment p	0/z	2/z	6/z	Not significant
E	4	....	7/z	0/z	1/z	Significant
G	4	....	3/z	3/z	2/z	Significant tendency
	4	....	3/z	1/z	4/z	Ambiguous
Total			**x/z**	**y/z**	**z-(x+y+)/z**	

4.4 The Deduction Step

The idea at this step is to put the pieces of evidence as presented in table 4 together and in perspective then interpret accordingly. For example, other statistical characteristics (profile) like mean values, the confidence interval, the profile plot, the stock plot etc. of those treatments that were of general significant values may be studied to establish facts.

5. Technique Validation

The technique was exercised with eight experiments that are replications. The experiments study the effectiveness and efficiency of three software evaluation techniques. The result look promising, nevertheless, its reliability and correctness need to be well ascertained. This can happen through continuous usage and comparison with the results of a technique with proven reliability. The task of achieving this is divided into two. The first stage will be to get more experiments whose results will be aggregated with this technique. The second stage will be to select of those experiments, the ones that can be aggregated using meta-analysis. The third stage will be to compare the out of the earlier two phases. This will go a long way to reveal the accuracy of the proposed less rigorous technique and suggest possible ways for improvement.

6. Conclusion

The focus of this work is to develop a less rigorous aggregation technique that will be useful enough to fill the aggregation vacuum created by the rules surrounding formal aggregation techniques which make SE experiments not suitable for aggregation. The technique will thus serve as an aggregation tool to software engineers before they have enough experiments for formal aggregation most especially when it is created out of SE experiments. The research has shown that aggregation of empirical works is difficult but it is possible to have a less rigorous tool for aggregating SE studies.

The tool has been used on some replications, work will continue on more replications as well as comparing its output with that of a formal aggregation technique.

References

[1] B. A. Kitchenham, et al., "Preliminary guidelines for empirical research in software engineering," IEEE Trans. Softw. Eng., vol. 28, p. 14, Aug. 2002.

[2] D. I. K. Sjoberg, et al., "The Future of Empirical Methods in Software Engineering Research," in Future of Software Engineering, 2007. FOSE '07, 2007, pp. 358-378.

[3] N. Juristo and A. M. Moreno, Basics of software engineering experimentation: Springer, 2001.

[4] N. Juristo, et al., "Limitations of empirical testing technique knowledge," in Lecture notes on empirical software engineering, ed: World Scientific Publishing Co., Inc., 2003, pp. 1-38.

[5] S. L. Pfleeger, "Soup or Art? The role of evidential force in empirical software engineering," Software, IEEE, vol. 22, pp. 66-73, 2005.

[6] A. Jedlitschka and M. Ciolkowski, "Towards Evidence in Software Engineering," presented at the Proceedings of the 2004 International Symposium on Empirical Software Engineering, 2004.

[7] D. E. Perry, et al., "Empirical studies of software engineering: a roadmap," presented at the Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, 2000.

[8] J. Miller, "Can results from software engineering experiments be safely combined?," in Software Metrics Symposium, 1999. Proceedings. Sixth International, 1999, pp. 152-158.

[9] J. Miller, "Applying meta-analytical procedures to software engineering experiments," Journal of Systems and Software, vol. 54, pp. 29-39, 2000.

[10]E. Fernández, "Aggregation Process with multiple evidence levels for experimental studies in Software Engineering," in

2nd International Doctoral Symposium on Empirical Software Engineering (IDoESE'07) ed, 2007.

[11]M. Dixon-Woods, et al., "Synthesising qualitative and quantitative evidence: a review of possible methods," Journal of health services research & policy, vol. 10, pp. 45-53B, 2005.

[12]R. K. Yin and K. A. Heald, "Using the case survey method to analyze policy studies," Administrative Science Quarterly, pp. 371-381, 1975.

[13]C. C. Ragin, The comparative method: Moving beyond qualitative and quantitative strategies: Univ of California Pr, 1989.

[14]D. L. Sackett, Evidence-based Medicine: Wiley Online Library, 2005.

[15]F. Davidoff, et al., "Evidence based medicine," Bmj, vol. 310, pp. 1085-1086, 1995.

[16]B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele University, vol. 33, p. 2004, 2004.

[17]S. MacDonell, et al., "How Reliable Are Systematic Reviews in Empirical Software Engineering?," Software Engineering, IEEE Transactions on, vol. 36, pp. 676-687, 2010.

[18]P. Mohagheghi and R. Conradi, "Vote-counting for combining quantitative evidence from empirical studies-an example," 2004.

[19]L. M. Pickard, et al., "Combining empirical results in software engineering," Information and Software Technology, vol. 40, pp. 811-821, 1998.

[20]M. Oivo, "New opportunities for empirical research," Empirical Software Engineering Issues. Critical Assessment and Future Directions, pp. 22-22, 2007.

Babatunde Olorisade obtained M.Sc degree in Software Engineering in 2009 and B.Sc in Computer Science in 2005. He currently works as a lecturer at Fountain University, Osogbo and as IT consultant to Dezeem Link Services Nigeria Limited. He has worked as software developer and was Assistant Manager, IT at Kasno Ventures, Nigeria (2006). His current research interest include empirical/search based Software Engineering, software measurement, software quality, e-commerce, security and cloud computing. He has published papers in referred conference proceedings and international journals. He is a member of the Nigeria Computer Society (NCS), IEEE, ACM and SEI.