Including the Composition Relationship among Classes to Improve Function Points Analysis

José Antonio Pow-Sang[1] Ricardo Imbert [2]

[1]Pontificia Universidad Católica del Perú

[2]Universidad Politécnica de Madrid

japowsang@pucp.edu.pe, rimbert@upm.es

Abstract

One of the most widely methods used to size estimation is Function Point Analysis. Since the introduction of object-oriented development in industrial practice, many OO and Function Point-like approaches have been presented.

This paper proposes the use of the composition relationship among analysis classes to improve the rules included in many of the FP-like approaches in order to identify Interface Logic Files (ILF) and External Interface File (EIF).

We also present the results obtained when undergraduate students applied our proposal in six case studies. The results of our approach have revealed to be more consistent and accurate than the original FPA technique.

1. Introduction

Function Point (FP) [IFPUG, 2000] [IFPUG, 2004] is a well-known software measurement technique. It was created by Allan Albrecht for IBM [Albrecht, 1979], and has gradually become an alternative to other popular size metrics, such as source lines of code (SLOC), becoming one of the most widely used technique for functional software measurement. The objectives of this technique are:

- To measure user requirements and what the user really gets.
- To provide a software measurement regardless of the technology utilized in the system deployment.
- To provide a size metric for quality analysis and productivity purposes.
- To provide an alternative for software estimation.
- To provide a normalization factor to compare different software.

The analysis of Function Points has been conducted considering five basic external system parameters: External Input (EI), External Output (EO), External Query (EQ), Internal Logic File (ILF) and, External Interface File (EIF).

With the diffusion of the Unified Modeling Language [OMG, 2005], promoted by the Object Management Group, many object-oriented approaches to calculate FP have been proposed. Unfortunately, they do not consider some specifications included in UML, such as the composition relationship among classes. For this reason, we propose in this paper an approach to calculate Logic Files (ILF and EIF) from an analysis class diagram that uses composition. We have also tested our approach against the Function Point Analysis (FPA) proposed by the International Function Points User Group (IFPUG) with good results.

This article has been divided in 7 different parts: Section 2 shows some work related to this paper, Section 3 shows the rules to identify the logical files that we propose, Section 4 presents the information of the empirical study; Section 5 shows the obtained results; Section 6 discusses the results and, finally, conclusions and future work related to this article are presented.

2. Related work

In order to cope with object-oriented systems measurement, several methods to calculate FP have been proposed. These methods reformulate the IFPUG rules in terms of OO concepts to facilitate the function points counting process. The final result of the count using this kind of techniques is similar to what would have been obtained by directly applying IFPUG FPA. Just to mention some of the most remarkable ones, Lehne [Lehne, 2004] presents an experience report in function points counting for object-oriented analysis and design using a method called OOram. Fetcke [Fetcke, 1997] defines rules for mapping OO-Jacobson method [Jacobson, 1992] to concepts from IFPUG Counting Practices Manual and three case studies have confirmed that rules can be applied consistently. Uemura et al. [Uemura, 1999] propose FPA measurement rules for design specifications, based on UML (Unified Modeling Language) and present a FP measurement tool. Cantone et al. [Cantone, 2004] and Caldiera et al. [Caldiera, 1998] present rules to map OO concepts to FPA and include in their papers pilot studies to demonstrate the applicability of their approaches. Finally, Abrahao et al. [Abrahao, 2004a] [Abrahao, 2004b] present a FP-like method called OOmFP and its evaluation through an empirical study.

The proposals presented above calculate FP-Logical Files (LF) from a class diagram, but they consider the aggregation relationship and do not the composition relationship. UML [OMG, 2005] defines composite aggregation or composition as a strong form of aggregation, which requires that a part instance be included in at most one composite at a time and that the composite object has sole responsibility for the disposition of its parts. The multiplicity of the aggregate end may not exceed one (it is unshared). Composition may be shown by a solid filled diamond and aggregation, by a non-filled diamond.

Composition presents a stronger dependence relationship than aggregation. Our approach includes both aggregation and composition relationship in order to identify files and its record element types (RET).

3. Rules to identify logical files

The input to use the rules we propose is the analysis class diagram included in Jaaksi's method [Jaaksi, 1998] or the domain model mentioned by Larman [Larman, 2004].

In this model transformation we are considering the following relationships among classes: association, aggregation and composition. Generalization and association class are not included. We do also not consider the difference between ILF and EIF; the following rules only consider the identification of logical files (LF) and its number of RETs.

- **Rule 1** Classes which are connected through a composition can be mapped together to one LF with 2 RETs. For example, in Figure 1 you must consider one LF and 2 RETs.

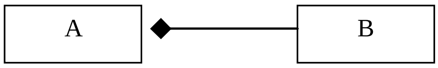

Figure 1. Example of composition

- **Rule 2** If there are three classes and two of them are connected through a composition, in the same way that it is shown in Figure 2, they must be mapped together to one LF for the composition (A and B) with 2 RETs and another LF for C.

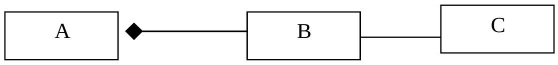

Figure 2. Example of composition and association

- **Rule 3.** If there are two classes, A and B, which are connected through an association or aggregation relationship, and neither of both is connected by composition to a third class, one must follow Table 1 indications. The table is an object oriented adaptation from IFPUG FPA CPM 4.2.1 [IFPUG, 2004]

Multiplicity A	Multiplicity B	When this condition exists	Then Count as LFs with RETs and DETs as follows:
0..*	0..*	A and B are independent	2 LFs
0..1	0..*	A and B are independent	2 LFs
1	1..*	If B is independent of A	2 LFs
		If B is dependent on A	1 LFs, 2 RETs
1	0..*	If B is independent of A	2 LFs
		If B is dependent on A	1 LFs, 2 RETs
0..1	1..*	If A is independent of B	2 LFs
		If A is dependent on B	1 LFs, 2 RETs
0..1	0..*	A and B are independent	2 LFs
1	1	A and B are dependent	1LF, 1RET
0..1	0..1	A and B are independent	2 LFs
1..*	1..*	If B is independent of A	2 LFs
		If B is dependent on A	1 LFs, 2 RETs
1..*	0..*	If B is independent of A	2 LFs
		If B is dependent on A	1 LFs, 2 RETs

Table 1. Rules to identify LF from classes without composition relationships

4. Experiment design and execution

Acknowledging the advantages present in the experiments involving students [Carver, 2003], we present in this section the experiment we have carried out to empirically evaluate the rules that we proposed in order to identify FP logical files in comparison with the IPFUG FPA.

For the experimental design, we have considered the experimental software engineering suggestions made by Juristo & Moreno [Juristo, 2001]. This experiment is similar to one presented by Abrahao et al. [Abrahao, 2004a]. The goal of the experiment was to empirically corroborate which method provides the best functional size assessment to identify logical files.

Using the Goal/Question/Metric (GQM) template for goal-oriented software measurement [Basili, 1994] we defined this experiment as follows:

- **Analyze**: LF measurement
- **For the purpose of**: evaluating our approach and the IFPUG FPA
- **With respect to:** their accuracy
- **From the point of view of:** the researcher
- **In the context of:** junior students (4th year) of Informatics Program at Pontificia Universidad Católica del Perú that were enrolled in the Software Engineering course in Spring 06

The research question is: Does our approach produce more accurate measurements of LF than IFPUG FPA?

4.1 Variables selection

The independent variable is the method used by subjects to size a case study. The dependent variable is the accuracy: the agreement between the measurement results and the true value.

4.2 Students who participated in the experiment

We selected a within-subject design experiment, i.e, the students had to use both methods to determine LF for each case study. The subjects were randomly assigned to two groups using the counterbalancing procedure, with equal numbers in each group. The methods were applied in a different order.

- Group 1 (12 students): our approach first and IFPUG FPA second.
- Group 2 (12 students): IFPUG FPA first and our approach second.

Table 2 shows knowledge and experiences that the students had at the beginning of this experiment.

Characteristic	Knowledge and/or Experience
Programming Language / Programming Environment	Java, C#, Pascal, C and Prolog.
Database modeling techniques	Entity relationship diagram, IDEF 1X
Analysis and Design Techniques	Structured and Object-oriented
Project Management	• Experience in managing developing projects that included short software programming projects (Work Team of 3 or 4 students). • No previous planning and estimation experience.

Table 2. Knowledge and experiences that the students had at the beginning of the experiment.

The vast majority of courses in the Informatics program at PUCP focus software projects as an application of theoretical concepts but they do not demand the application of estimation and planning techniques when developing software projects.

4.3 Materials and case studies

The materials used in the experiment are the following:

- Case studies description.
- Form to fill in the number of LF and its RETs for each case study.
- A questionnaire to know student's opinion about which technique is easier to apply.
- A summary of Coad's patterns [Coad, 1997]. We taught these patterns in a previous semester and we gave them to the students in order they can elaborate diagrams correctly.

The descriptions of six case studies with their analysis class diagrams are the following.

- **Case Study 1 (see Figure 3).** In a Sales System, it is only required to automate the registration of customers and the main data of their invoices. The information of the client is the following: code, name, address and phone number. The information of the invoice is: number, date and total amount. In this case, customers without invoices and invoices without clients can exist in the system.

Figure 3. Class diagram for case study 1.

- **Case Study 2 (see Figure 4).** It is required to develop a Sales System that will register invoices. The information of the invoices is: number, date, taxes, total amount. Additionally, each line of the invoice (detail) will contain: line number, product description, product quantity, product unit price and subtotal. The system also must allow the registry of the types of products that sells and their information: code, description and price.

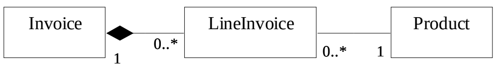

Figure 4. Class diagram for case study 2.

- **Case Study 3 (see Figure 5).** It is required to develop a system that registers the universities and their students. The information of the universities is: code, name, address, Web page and telephone number. Students' information: code, name, address, e-mail and telephone number. The system will have to allow universities without students and students without universities. The students can belong to more than one university.

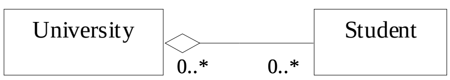

Figure 5. Class diagram for case study 3.

- **Case Study 4 (see Figure 6).** It is required to make a system that allows registering project plans and its activities. The system will be able to allow that the project plan does not have defined activities, but an activity always must be associated to a project plan. An activity is only associated to a specific project plan.

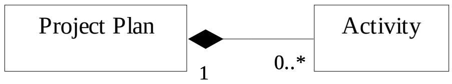

Figure 6. Class diagram for case study 4.

- **Case Study 5 (see Figure 7).** It is required to develop a system that allows registering and maintaining the information of CDs. CD information is code, title and duration. Each track of the CD has the following information: number of track, song name, artist and duration.

Figure 7. Class diagram for case study 5.

- **Case Study 6 (see Figure 8).** It is required to develop a system that allows registering customer orders. Information of each order is: number, date and customer name. Additionally each line of the order (detail) will contain: line number, product code, product description and product quantity. The system also must allow registering product information that the company sells, such as code and description.

Figure 8. Class diagram for case study 6.

Further details of the case studies and instruments used can be found at the following link http://inform.pucp.edu.pe/~jpowsang/pf/composition.html

4.4 Tasks performed in the experiment

Table 3 shows the tasks carried out in the session by the students.

<table>
<tr><th>N°</th><th>Group 1</th><th>Group 2</th></tr>
<tr><td>1</td><td colspan="2">Receive the case studies</td></tr>
<tr><td>2</td><td>Receive material with the explanation of our approach, Coad's patterns and form to fill number of LF and its RETs</td><td>Receive material with the explanation of IFPUG FPA and form to fill number of LF and its RETs</td></tr>
<tr><td>3</td><td>Elaborate analysis class diagrams and identification of LF</td><td>Elaborate E-R diagrams and identification of LF</td></tr>
<tr><td>4</td><td colspan="2">Give completed forms back with the results</td></tr>
<tr><td>5</td><td>Receive material with the explanation of IFPUG FPA and form to fill number of LF and its RETs</td><td>Receive material with the explanation of our approach, Coad's patterns and form to fill number of LF and its RETs</td></tr>
<tr><td>6</td><td>Elaborate E-R diagrams and identification of LF</td><td>Elaborate analysis class diagrams and identification of LF</td></tr>
<tr><td>7</td><td colspan="2">Give completed forms back with the results and receive questionnaire</td></tr>
<tr><td>8</td><td colspan="2">Give questionnaire back</td></tr>
</table>

Table 3. Tasks of the session carried out by the students

The session lasted approximately one hour and the students performed 45 minutes on average in all of the tasks. Although, it was not part of our study to know which technique demands less time, we could observe that they almost used the same time to identify LF in both techniques.

It is important to mention that we commented to the students that the purpose of the questionnaire is to know their honest opinion about which technique is easier to apply.

5 Results

For each case study, we grade with 1 if the student identified the number of LF and RET correctly and with 0 if he or she did incorrectly. Table 4 presents the results obtained for the case studies. We established a significance level of 0.05 to statistically test the results obtained.

Variable	IFPUG FPA	Proposal
Observations	24	24
Minimum	2	2
Maximum	6	6
Mean	5.125	5.792
Std. Deviation	1.329	0.833

Table 4. Descriptive statistics for functional size

In order to determine if the grades obtained with each technique follow a normal distribution, we applied Shapiro-Wilk test. Table 5 shows the results obtained with this test.

Variable	IFPUG FPA	Proposal
W	0.69780878	0.27715518
p-value	< 0.0001	< 0.0001
Alpha	0.05	0.05

Table 5. Results obtained to test normality distribution with the Shapiro-Wilk test

As the computed p-value is lower than the significance level alpha=0.05 (see Table 5), we can reject the normality distribution hypothesis for both techniques and accept the non-normal distribution hypothesis. Because of these results, we can not use the paired samples t-test and we have to select a non-parametric alternative. The Wilkoxon signed rank test was chosen for this purpose. The statistical hypotheses formulated to test both techniques are:

- H0: The distribution of the two samples is not significantly different.
- Ha: The distribution of the IFPUG FPA sample is shifted to the left of the distribution of the proposal FPA sample.

Variable	Result
V	17.5
Expected value	90
Variante (V)	843.125
p-value (one-tailed)	0.006575978
Alpha	0.05

Table 6. Wilkoxon signed rank test results

As the computed p-value is lower than the significance level alpha=0.05 (see Table 6), we can reject the null hypothesis H0 and accept the alternative hypothesis Ha. It means that we can empirically corroborate that our proposal produces more consistent assessments than IFPUG FPA.

Question	Answer		
	No	**Both are the same**	**Yes**
Rules regarding composition (technique with classes) facilitate LFs and RETs identification compare to technique without classes	0%	0%	100%
The technique with classes is easier to apply than the technique without classes	0%	12.5%	82.5%

Table 7. Results of the questionnaire

As it can be observed in Table 7, all of the students consider that rules proposed in our model transformation facilitate LFs and RETs identification and almost all of the students think our proposal is easier to apply than IFPUG FPA. The questionnaire results do not disagree with the quantitative results presented above.

6. Discussion

In this section, we discuss various threats to validity of the empirical study and the way we attempted to alleviate them.

6.1 Threats to construct validity

The dependent variable: accuracy that we used is proposed in the ISO/IEC 14143-3 [ISO, 2003].

6.2 Threats to internal validity

Seeing the results of the experiment we can conclude that empirical evidence of the existing relationship between the independent and the dependent variables exists. We have tackled different aspects that could threaten the internal validity of the study:

- Differences among subjects. Error variance due to differences among students is reduced by using within-subjects design.
- Learning effects. The counterbalancing procedure (subjects were randomly assigned in two groups) cancelled the learning effect due to similarities and the order of applying both techniques.
- Knowledge of the universe of discourse. We used the same case studies for all subjects.
- Fatigue effects. Each student took 45 minutes on average per session to apply both techniques. So, fatigue was not relevant.
- Persistence effects. The students had never done a similar experiment.

- Subject motivation. The students were motivated because they had to apply FP in order to estimate required effort in their projects assigned in the semester.

6.3 Threats to external validity

Two threats to external validity have been identified which limit the ability to apply any such generalisation:

- Materials. We used representative case studies in which students had to identify association, aggregation and composition relationships among classes. However, more empirical studies are needed using others case studies or software requirements specifications [IEEE, 1998].
- Subjects. We are aware that more experiments with practitioners must be carried out in order to generalize these results.

7. Conclusions and Future Work

This paper presented a conversion model to determine FP logic files using an analysis class diagram. This model considers composition relationship among classes that it is not included in others approaches.

We also described a controlled experiment with undergraduate students in order to determine the accuracy of our model compare to IFPUG FPA. The results of the experiment present that our model produces more accuracy results than the original IFPUG FPA rules and students perceived that our approach is easier to use than IFPUG FPA. Although, results obtained from the experiment are very encouraging, we are aware that more experimentation is needed in order to confirm them.

The future work in regard to this article is:

- To conduct experiments with industrial practitioners in order to get more results and opinions about the applicability of our model in the industry.
- To include generalization and association-class in our conversion model.
- To define rules to transform UML diagrams into IFPUG FPA transactions (EI; EO and EQ).

Acknowledgements

The authors would like to thank Dra. Ana María Moreno, professor from Universidad Politécnica de Madrid, for her invaluable comments to prepare the experiment. We would also like to thank Otto Pflücker for his suggestions and comments to this paper.

References

[Abrahao, 2004a] Abrahão, S., Poels, G., Pastor, O., Assessing the Reproducibility and Accuracy of Functional Size Measurement Methods through Experimentation, Proceedings ISESE 2004, IEEE Computer Society, 2004.

[Abrahao, 2004b] Abrahão, S., Poels, G., Experimental evaluation of an object-oriented function point measurement procedure, Information & Software Technology, Elsevier, 2007.

[Albrecht, 1979] Albrecht, A. J., "Measuring Application Development Productivity", in IBM Applications Development Symposium, Monterey, CA, 1979.

[Basili, 1994] Basili, V. R., Caldiera, G. and Rombach, H. D., "Goal Question Metric Paradigm", Encyclopedia of Software Engineering, ed. J. J. Marciniak, Wiley 1994.

[Caldiera, 1998] Caldiera, G., Antoniol, G., Fiutem, R., Lokan, C., Definition and Experimental Evaluation of Function Points for Object-Oriented Systems, Proceedings METRICS'98, IEEE Computer Society, 1998.

[Cantone, 2004] Cantone, G., Pace, D., Calavaro, G., Applying Function Point to Unified Modeling Language: Conversion Model and Pilot Study, Proceedings of METRICS'04, IEEE Computer Society, 2004.

[Carver, 2003] Carver, J., Jaccheri, L., Morasca, S., Issues in Using Students in Empirical Studies in Software Engineering Education, Proceedings METRICS'03, IEEE Computer Society, USA, 2003.

[Coad, 1997] Coad, P., North,D., Mayfield,M., Object Models: Strategies, Patterns and Applications, Prentice-Hall, USA, 1997.

[Fetcke, 1997] Fetcke, T., Abran, A. and Nguyen, T., "Mapping the OOJacobson Approach into Function Point Analysis", IEEE Proceedings of TOOLS-23'97, 1997.

[IEEE, 2000] IEEE Computer Society, IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications, The Institute of Electrical and Electronics Engineers, USA, 1998.

[IFPUG, 2000] IFPUG, "Function Points Counting Practices Manual (version 4.1.1)", IFPUG: International Function Point User Group, Westerville Ohio, 2000.

[IFPUG, 2004] IFPUG, "Function Points Counting Practices Manual (version 4.2.1)", IFPUG: International Function Point User Group, Westerville Ohio, 2004.

[ISO, 2003] ISO. ISO/IEC 14143-3 - Information technology -- Software measurement -- Functional size measurement -- Part 3: Verification of functional size measurement methods, 2003.

[Jaaksi, 1998] Jaaksi, A., A Method for Your Object-Oriented Project, Journal of Object-Oriented Programming, Vol 10. No 9, 1998.

[Jacobson, 1992] Jacobson, I., Object-Oriented Software Engineering. A Use Case Driven Approach, Addison-Wesley, USA, 1992.

[Juristo, 2001] Juristo, N., Moreno, A.M., Basics of Software Engineering Experimentation, Kluwe Academic Publishers, Boston, 2001.

[Larman, 2004] Larman, C., Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition, Addison-Wesley, 2004

[Lehne, 1997] Lehne, O.A., Experience Report: Function Points Counting of Object Oriented Analysis and Design based on the OOram method. in Conference on Object-Oriented Programming Systems, Languages,and Applications (OOPSLA'97), 1997.

[OMG, 2005] Object Management Group, OMG Unified Modeling Language, http://www.uml.org, USA, 2005.

[Uemura, 1999] Uemura, T., Kusumoto, S. and Inoue, K., Function Point Measurement Tool for UML Design Specification. in 5th International Software Metrics Symposium (METRICS'99), Florida, USA, 1999, 62-69.