Published in the Proceedings of SMEF 2006

The Comparison between FPA and COSMIC-FFP

Gu Xunmei, Song Guoxin, Zheng Hong

Abstract

Functional Size Measurement (FSM) methods are intended to measure the size of software by quantifying the Functional User Requirements (FUR) of the software. The capability to accurately quantify the size of software in an early or late stage of the development lifecycle is very important to software project indicators. The IFPUG FPA and COSMIC-FFP are the main methods of FSM methods and they have already passed the International Standards. Furthermore the two methods are now used widely in practice. In this paper, we compare the two methods, and then find out the differences and similarities between IFPUG FPA and COSMIC-FFP. Finally we come to a conclusion and point out the applicable scope of each method.

1. Introduction

A Functional Size Measurement (FSM)[1][2][3] method measures the logical external view of the software from the users' perspective by evaluating the amount of functionality to be delivered. The capability to accurately quantify the size of software in an early or late stage of the development lifecycle is critical to software project managers for evaluating risks, developing project estimates (e.g., effort, cost) and having early project indicators (e.g., productivity).

The most widely used FSM method is the IFPUG Function Point Analysis (IFPUG FPA). It is based on the method proposed by Alan Albrecht. This method can be considered as the first FSM method. This technique was developed specially to measure the amount of data that each function accesses as an indicator of functional size. However, this method was developed to measure Management Information Systems (MIS) and assumes the use of traditional software development methodologies such as structured analysis and design.

Although IFPUG FPA has obtained increasing popularity in the industry; the lack of applicability to all software types and the rapid evolution of the development paradigms have created a great number of variations of this method. To overcome this diversity, COSMIC-FFP [8] has emerged as the second generation of the FSM methods. We decide to use this method due to its generic character applicable to several software domains, its compatibility with modern software engineering concepts and its approval as an International Standard FSM method.

The structure of this paper is as follows: Section 2 describes an overview of the IFPUG Function Point Analysis with emphasis on the measurement principle. Section 3 briefly introduces the main concepts of COSMIC-FFP and then presents the measurement principle and model. Section 4 compares the two FSM methods from several aspects and describes the fields tests conducted and highlights the main result. Section 5 points out the applicable scope of each method and presents some observations and topics for further research.

2. IFPUG Function Point Analysis

2.1. IFPUG FPA Measurement Principle

In IFPUG Function Point Analysis [6], the functional size is calculated by quantifying the number of the five types of components that exist, namely: External Inputs (e.g., transaction types entering data into the system), External Outputs (e.g., report types producing information output), External Inquiries (e.g., types of inquiries supported), Internal Logical Files (data maintained by the system), and External Interface Files (e.g., data accessed by the system but not maintained by it). Furthermore, IFPUG FPA's view on functional size is that a software system consists of logical data files and functions. For one thing, logical data files contain Internal Logical Files and External Interface Files indicated respectively by 'ILF' and 'EIF'; for another, functions contain External Inputs, External Outputs and External Inquiries indicated respectively by 'EI', 'EO' and 'EQ'. These components are weighed (according to their complexity-low, average or high), and their weights are summed. This value is then adjusted using fourteen general system characteristics to produce a functional size measure. The IFPUG FPA structure is described in Figure 1.

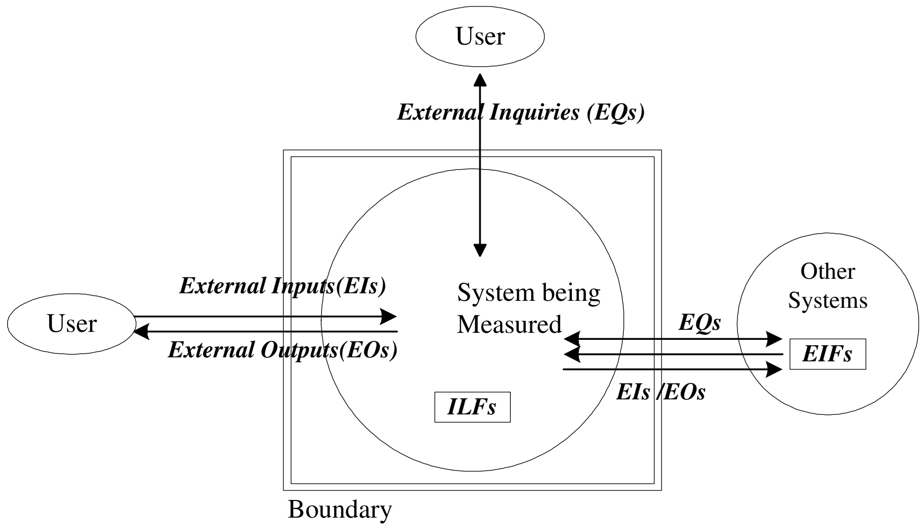

Figure 1: IFPUG FPA Structure

The amount and complexity of the data (types) 'handled' by a logical data file transactional function determines the amount of functionality that this piece of software delivers, hence its functional size. The boundary [1][6] separates the software system being measured from its environment. Within this environment are the users of the system, which may include other systems. Also systems that are used by the system within the scope of measurement are identified.

2.2. IFPUG FPA Meta-model

The IFPUG FPA [6] Meta-model is showed in Figure 2. It illustrated the information we need to gain for representing a software system (called 'project' in IFPUG FPA terminology) that will be measured. Measuring a project includes following activities: determination of the type of count (new project, enhanced project or running application), and identification of the counting scope and application boundary, identification and classification of data and transaction functions (FunctionType).

According to the type of count, the development project function point count measures the functions of a new software application that will be provided to the users, upon completion of the project. The enhancement project function point measures the modifications to an existing application that add, change or delete user functions. Finally, the application function point count measures an installed application.

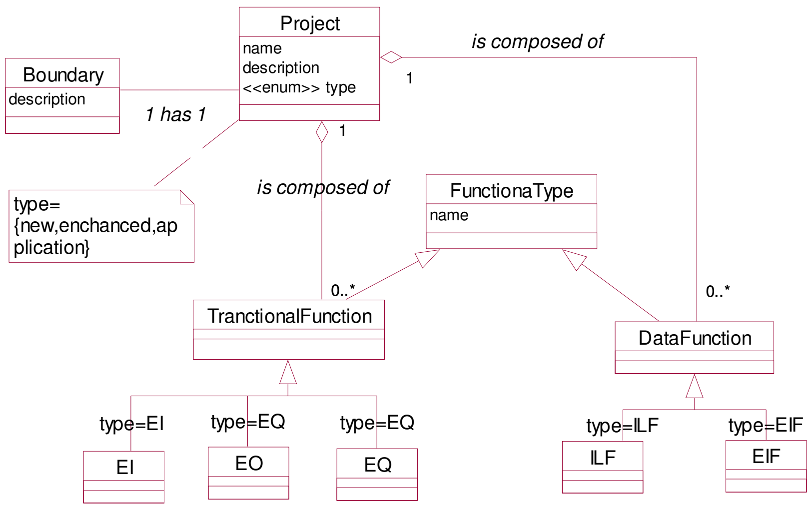

Figure 2: IFPUG FPA Meta-model

3. COSMIC-FFP

3.1. Allocation and Representation of Functional User Requirements

From the perspective proposed by COSMIC (Common Software Measurement International Consortium), software is part of a product or service designed to satisfy Functional User Requirements (FURs) [4][5]. From the high-level perspective, Functional User Requirements can be allocated to hardware, to software or to a combination of both. The Functional User Requirements allocated to software are not necessarily allocated to a single unit of software. Usually, these requirements are allocated to pieces of software operating at different levels of abstraction and cooperating to supply the required functionality to the product or service in which they are included.

In the context of the COSMIC-FFP measurement method, which is aimed at measuring the functional size of software, only those Functional User Requirements allocated to software are considered. The method may be applicable to size the functional requirements for information processing which are allocated to hardware, but this needs further research.

The Functional User Requirements allocated to one or more pieces of software are decomposed into and represented by functional processes. A sub-process can either be a data movement type or a data transform type. The version 2.0 of the COSMIC-FFP [8] measurement method recognizes only data movement type sub-process. Further research is deemed necessary to incorporate data transform sub-process in the measurement method. The approach is illustrated in Figure 3.

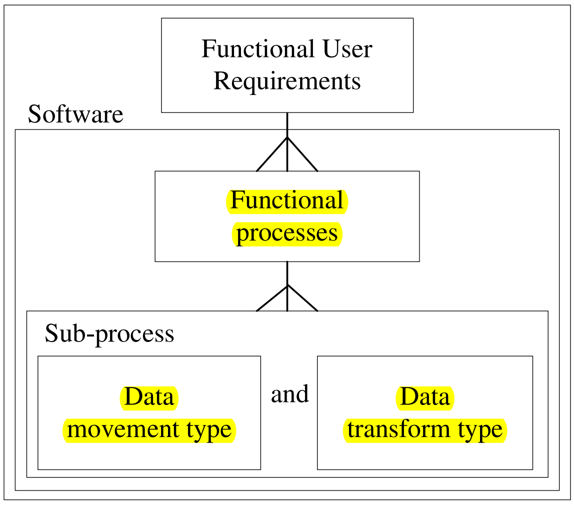

Figure 3: COSMIC Representation Of Functional User Requirement

Given the approximating assumption, a COSMIC-FFP functional process is defined as "a unique and order set of data movements which is triggered by an event outside the software being measured, and which, when complete, leaves the software in a coherent state with respect to the external event".

3.2. COSMIC-FFP Software Model

The COSMIC-FFP measurement method defines an explicit model of software functionality, derived from the functional user requirement. Based on this explicit model of functionality, relevant functional attributes of software are identified. Regarded as a whole, these functional attributes form a generic model for any type of software, which is not 'algorithm-rich'. The model is illustrated in Figure 4.

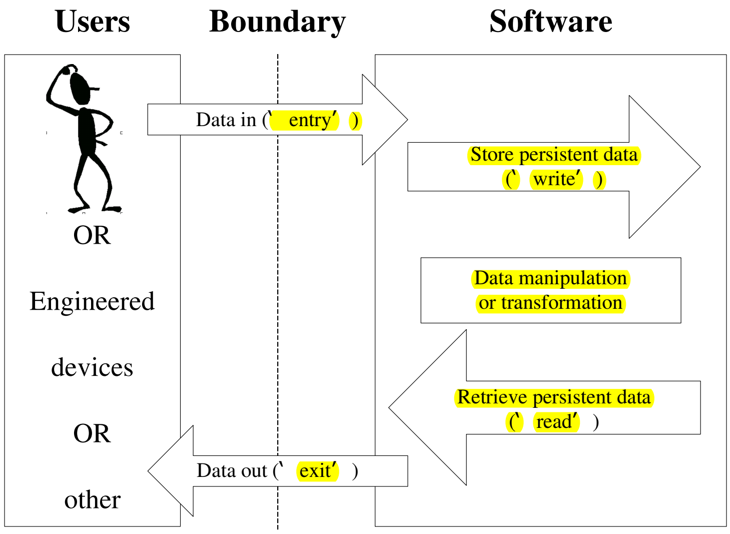

Figure 4: COSMIC-FFP software model

Four types of data movement are defined within this model. They form the basis for defining the standard unit of functional size. The four types of data movement are as follows.

- Entry: A sub-process that receives control data coming from outside the application's boundary. For example, each sub-process receiving data coming from one sensor is considered as an entry.
- Exit: A sub-process that sends control data outside the application boundary. For example, each sub-process sending a signal to a device's control panel display is considered an Exit.
- Read: A sub-process that reads a group of control data. For example, each sub-process that obtains a threshold value from a group of control data is considered as Read.
- Write: A sub-process that writes to a group of control data. For example, each sub-process updating current value of key variables inside the measured application is considered a Write.

In COSMIC-FFP, the standard unit of measurement, that is, 1 Cfsu, defined by convention as equivalent to one single data movement at the sub-process level.

To count the functional size of software, firstly we should count every functional process size by the formula: $size = \sum size(sub-processes)$, then count the size of every software layer by the formula: $size = \sum size(functional\ processes)$, finally add the size of all the software layers.

4. Comparing IFPUG FPA with COSMIC-FFP

4.1. Similarity between IFPUG FPA and COSMIC-FFP

4.1.1. Measurement steps

The International Workshop on software measurement has proposed a generalized representation for FSM methods. According to this representation, FSM requires two steps of abstraction: first, the elements that contribute to the functional size of the system are identified (identification step); and second, these elements are mapped onto numbers (measurement step)[9][10].

- IFPUG FPA method

  In the identification step, (given a Functional User Requirements specification), the data model and process models are specified, and windows, screens and reports are designed. This is an implicit step that is not described as part of the measurement process; however, the IFPUG Count Practices Manual [6] suggests that the measurement may be based on one or more of the following components: user requirements, data models (Entity-Relationship Diagrams), process models (Data Flow Diagrams) and the design of windows, screens, and reports. In some examples, the requirements specification document stands alone as the basic for the measurement, but most of the examples presented in the manual include a data or a process model, and designs of windows, screens, and reports. Significant elements for measurement are identified using these documents, models and designs.

  In the measurement step, the previously identified elements are captured in an abstract model that is specified according to the IFPUG FPA Abstract Model. This 'meta-model' describes all elements that contribute to the functional size of the system. The construction of the abstract model is an implicit process that is performed by applying the IFPUG measurement rules, especially those that are required to identify and classify the five types of components as well as to rate their complexity. The abstract model is then used assigns a numerical value to the system representing its functional size (i.e. translating complexity-rated and classified functions into function point values, aggregating the values).

- COSMIC-FFP method
  In the identification step, the COSMIC-FFP takes as input the specification of Functional User Requirements (FUR). This specification may be at different levels of abstraction, which generates the identification of different software layers as a result of a functional partition of the system. Later the boundary is identified, which is defined as a conceptual interface between the software under study and its users. Then the collection of Functional User Requirements (FUR) can be decomposed into a set of functional processes and each functional process consists of a set of data movements.

  In the measurement step, each data movement should be identified. The objective of this step is to produce a quantitative value based on the measurement principle. Then assign a numerical quantity to a data movement. Finally, the functional size of a software layer is defined as the sum of the functional sizes of its respective functional processes.

4.1.2. Expression of functional requirements

Both of the COSMIC-FFP and IFPUG FPA techniques express the functional requirement in terms of Base Functional Component (BFC). ISO/IEC 14143-1 defines the BFC as "an elementary unit of Functional User Requirement defined by and used by an FSM Method for measurement purpose" [1]. COSMIC-FFP method includes 4 BFCs: Entry, Exit, Read and Write, while IFPUG FPA includes 5 BFCs: External Input, External Output, External Inquiry, Internal Logical File and External Interface File.

4.1.3. Measurement phase

All the two methods can approximately evaluate the size of software at the early stage of software development lifecycle in order to complete the system well, i.e. we can estimate the effort and feasibility according to software specification in requirement phase. As long as the functionality of users is determined, we can measure the function size of the software to be developed.

4.1.4. Convertibility

Convertibility is defined as "an ability to convert the results from applying two or more FSM Methods in the measurement of a Functional Size of the same set of Functional User Requirement" [3]. In some case, IFPUG FPA can be converted to COSMIC-FFP, although further research is necessary to prove general applicability of this conversion. But we can only convert the final results of the two methods, because the BFCs of the two methods are different and every BFC expresses different meanings. By conversion, we want to prove whether the two methods are universal.

4.2. Difference between COSMIC-FFP and IFPUG FPA

4.2.1. Counting rule

Although COSMIC-FFP and IFPUG FPA are all FSM methods, their counting rules are different. IFPUG FPA method considers fourteen system characteristics while COSMIC-FFP method does not take the system characteristics into account. The fourteen system characteristics are data communications (e.g. Web connection), distributed processing (e.g. client/server), performance (e.g. minimum response time), heavily used configuration (e.g. set up likely to change often such as dynamic web content), transaction rates, on line data entry, design for end user efficiency, online updates, complex processing (e.g. calculations or lookups), usable in other applications, installation ease, operational ease, multiple sites, facilitate change. But in IFPUG FPA, the analysis of adjusted factors is not objective.

4.2.2. Different results from field tests

The measurement results presented in this section were collected in field tests subsequent to the release of the COSMIC-FFP measurement method. The industrial sites were in North America and Pacific region. The software products measured were classified as telecommunications software, power utility software and military software. All the measurement procedures were executed by the same measurement expert with twelve years of experience in FSM, so eliminating inconsistencies across measurement expert.

Table 1 presents measurement results of seven distinct software products. These products were measured using both the IFPUG 4.0 rules and the Full Function Points measurement methods. They all come from the same organisation. Distinction between real-time and MIS software products is based on the knowledge of functional experts and practitioners within the organisation. The results of the tests are in Table 1 [7].

Product	Type	Size (IFPUG FPA)	Size (COSMIC-FFP)
A	Real-time	210	794
B	Real-time	115	183
C	Real-time	---	2604
D	Real-time	43	318
E	Mostly MIS	764	791
F	MIS (batch)	272	676
G	MIS	878	896

Table I: Size comparison between FPA and FFP

We can draw a conclusion from the results presented in Table 1. The conclusion is as following:

- The size results are similar when both methods are applied to MIS-type software products, as demonstrated by measurement of products E and G;
- The software product ”C” could not be measured at all using IFPUG FPA because the functionality it supplied could not be reliably categorized into IFPUG FPA function type, nor did it fit the definition of elementary process according to IFPUG FPA 4.0 rules;
- The size difference is significant in all cases where the software product has complex real-time processes. Product A, B, C and D are examples of this. The COSMIC-FFP size is considerably greater than the IFPUG FPA size.

4.2.3. Independence character

COSMIC-FFP is completely independent of software development technology and program language while IFPUG FPA is not. The five components of software are limited in IFPUG FPA when the software reaches some scale.

4.2.4. Measurement viewpoint

IFPUG FPA measures software only from the views of end users while COSMIC-FFP can measure the size of software from the views of end users and developers.

4.2.5. The way to deal with technical and quality requirements

A Base Functional Component should not express technical and quality requirements. IFPUG FPA accounts for technical and quality requirements via a Value Adjustment Factor (VAF) while COSMIC-FFP measures the size of software purely on the base of Functional User Requirements.

In IFPUG FPA, each project also has a VAF based on other system characteristics of the project. Each characteristic is given a rating from (0-not important) to (5-very important); the rating is called the Degree of Influence (DI). The value for each system characteristic is summed to gain a Total Degree of Influence (TDI); this provides a VAF of 0 to 70, which is then used in the following formula:

$$VAF = (TDI * 0.01) + 0.65$$

This formula is then used to create the final project point count on the basis of the functional size of project.

4.2.6. Applicable scope

IFPUG FPA is applicable to size business applications software such as MIS and has been used extensively in productivity and estimation. In contrast, COSMIC-FFP is applicable to Real-time, embedded system and almost all the software system. But COSMIC-FFP is not capable of sizing complex mathematics algorithms.

The granularity of COSMIC-FFP allows much better to capture the functional size variations within individual functional processes exhibiting quite distinct averages and standard deviations for each software while the allowed range of the IFPUG FPA method is much smaller and, therefore, less sensitive to the large diversity of functional processes encountered in real-time software.

5. Conclusion

Although both COSMIC-FFP and IFPUG FPA are FSM methods, they are different from each other in some aspects. IFPUG FPA was designed before the publishing of International Standard ISO/IEC 14143 while COSMIC-FFP was after ISO/IEC 14143. So COSMIC-FFP was designed to ensure its full compliance with ISO/IEC 14143. But some items of IFPUG FPA were not adopted by ISO and IEC.

The range of software (specifications) that can be sized with COSMIC-FFP is definitely wider than the range covered by IFPUG FPA. So, the generality of COSMIC-FFP needs to be instantiated through a more specific and systematic procedure in accordance with software development method.

Further research activities have been initiated to address the following themes:

- Convertibility studies with previous methods such as: COSMIC-FFP, Mark II and IFPUG FPA;
- Mapping of measurement rules into the UML-based specifications domain;
- Measurement of Functional Reuse using COSMIC-FFP;
- Web-site (standard & publication): www.lrgl.uqam.ca/ffp;
- Convertibility studies with previous methods such as: COSMIC-FFP Version 2.0, Mark II, and IFPUG FPA.

6. References

[1] ISO/IEC 14143-1:Information Technology - Software measurement – Functional Size Measurement-Part1: Definition of Concepts, 1998.

[2] ISO/IEC 14143-2: Information Technology - Software measurement – Functional Size Measurement-Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1: 1998.2002.

[3] ISO/IEC 14143-3:Information Technology - Software measurement – Functional Size Measurement-Part 3: verification of functional size measurement methods.2003.

[4] ISO/IEC 14143-4:Information Technology - Software measurement – Functional Size Measurement-Part 4: Reference model.2003.

[5] ISO/IEC 14143-5:Information Technology - Software measurement – Functional Size Measurement-Part 5: Determination of functional domains for use with functional size measurement.2002.

[6] Abrahao, S. and Pastor, O., Estimating the Applications Functional Size from Object-Oriented Conceptual Models. In International Function Points Users Group Annual Conference (IFPUG'01), Las Vegas, USA, 2001.

[7] Abrahao, S., Poels, G. and Pastor, O. Comparative Evaluation of Functional Size Measurement Methods: An Experimental Analysis, Working Paper, Faculty of Economics and Business Administration, Ghent University, Belgium, 2004, 43 p.

[8] Abran, A., Desharnais, J.M., Oligny, S., St.Pierre, D. and Symons, C. COSMIC-FFP Measurement Manual, Version 2.1, The Common Software Measurement International Consortium, May 2001.

[9] Abran, A. and Jacquet, J.P., A Structured Analysis of the new ISO Standard on Functional Size Measurement - Definition of Concepts. in 4th IEEE Int. Symposium and Forum on Software Engineering Standards, 1999, 230-241.

[10] Abran, A. and Robillard, P.N. Function Points: A Study of Their Measurement Processes and Scale Transformations. Journal of Systems and Software, 25, 1994, 171-184. 6. Albrecht, A.J., Measuring application development productivity. In IBM Application Development Symposium, 1979, 83-92.