International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-1, Issue-8, October 2013

Function Point Analysis

S.Sowmya, N.Vignesh

Abstract— **With the development of software industry, software estimation and measurement is catching much more concern. Scale increasing in applications and a variety of programming languages using at the same time, manual measurement based on the LOC (Line of Code) cannot meet the estimating requirements. Security is becoming more and more important in most of the software construct. The emergence of Function Point resolves these difficult issues. It is extensively used in production analysis and estimation. FPA can also be used in the estimation of embedded and real-time software. In recent years, Object Oriented (OO) technology has emerged as a dominant practice in Software Engineering domain. it is useful to match traditional Function Point (FP) measurement to new OO approaches, including models based on Unified Modeling Language. Web applications can be estimated by Web objects. It is a quite different measure, since it starts from the most classic functional size metric, Function Points, and extends it by taking into account components that can be typically found in a web system.**
This paper gives about the overview of methods available in function point analysis.

Index Terms— **Software estimation, function point analysis, web objects, fuzzy rules.**

I. Introduction

Project progress, cost and quality are the three elements of software projects. Size estimation is the key of software engineering. The accurate estimation not only could promote exploitation resources allocation, but also it deeply affects the development of the software project. The software is invisible and unmanageable, so the software size problem becomes difficult. The facts show that estimating the software effort at the early stages of the project will contribute to the management arrangements for the project development plan, control the cost and ensure the software quality at the same time. On the one hand, software self could change according to function requirements in the course of compilation. On the other hand, the software size easily suffers many factors, such as person, technology, and environment and exploiter strategy. Function Point Analysis has been proposed in the late 70' as an effective means for measuring the size of software systems and forecasting the development and maintenance effort ; its central principle is the use of functional requirements as the main input of size estimation, which overcomes the problems inherent to the use of code-level, physical parameters for dimensioning the software. It has steadily evolved into a functional size metric consolidated in the Well-accepted Standardized International Function Point User Group (IFPUG).

Section II says about the traditional function point analysis. Section III deals with Full Function points being used for estimation of real time software.

II. TRADITIONAL FUNCTION POINT ANALYSIS

A. *Characteristics of function point*

Function Point Analysis is a software estimation method that oriented on function, which measures the size of system mainly from its functionality and usability. In its view, system consists of the following 5 characteristics:

- EI (External Input): user input to the software which provides application-oriented data.
- EO (External Output): Report, screens, error messages and so on.
- EQ (External Query): a real-time response when users enter an online input.
- ILF (Internal Logical File): a logical main document, such as a logical combination of data.
- EIF (External Interface File): A machine-readable interface, such as data files on the tapes or disks, which can be used to transport information from one System to the other.

Actually, function point is the sum of these weighted characteristics.

Type Sizes for Papers

Component	Low	Average	High
External Input	3	4	6
External Output	4	5	7
External Query	3	4	6
Internal Logical File	7	10	15
External Interface File	5	7	10

B. *Calculating function point*

- Determine the type of function point count. Function point counts can be associated with either projects or applications, which has three types: development project, enhancement project and application.
- Identify the counting scope and application boundary.
- Count the data functions to determine their contribution to the unadjusted function point count.

Manuscript received October 14, 2013.

S.Sowmya is working as a Software Development Engineer at ACS, Oracle India
N.Vignesh is working as a Research Associate at Indian Institute of Management, Ahmedabad.

- Count the transactional functions to determine their contribution to the unadjusted function point count.
- Determine the value adjustment Factor. The value adjustment factor (VAF) is based on 14 general system characteristics (GSCs).
- Then VAF (Value Adjustment Factor) is calculated like this:
  VAF = (TDI * 0.01) +0.65
- Calculate the adjusted function point count according to the formula below:
  FP=VAF*UFP
  In which UFP stands for Unadjusted Function Point.

General system characteristics

- Data communications
- Distributed data processing
- Performance
- Heavily used configuration
- Transaction rate
- Online data entry
- End-user efficiency
- Online update
- Complex processing
- Reusability
- Installation ease
- Operational ease
- Multiple sites
- Facilitate change

III FULL FUNCTION POINTS

Full function points is a measure of the functional size of the software, from a user perspective and not taking into account the technical characteristics of the software. A new measurement basis was introduced in FFP, together with corresponding data and functional types closer to the control data characteristics.

The two new Control data function types have a structure similar to that of the IFPUG Data Function types:

Updated Control Group (UCG): It is a group of control data updated by the application being measured. It is identified from a functional perspective.

Read-Only Control Group (RCG): It is a group of control data used, but not updated, by the application being measured.

The four new Control transactional function types address the sub process of real time software.

External Control Entry (ECE): It controls the data coming from outside the application's boundary.

External Control Exit (ECX): It controls the data going outside the application's boundary.

Internal Control Read (ICR): It reads the control data. **Internal Control Write (ICW):** It writes control data.

FFP is an extension of IFPUG measurement method; all IFPUG rules can be included in an application of this new Measurement technique. Unadjusted count of an application using the FFP can be expressed as follows:

FFP=Management FP+Control FP
=(FPA-Control information)+Control FP

IV IMPROVED FPA

An improved function point analysis (FPA) method was proposed for analyzing the software size. The method combined the advantage of the fuzzy rules and back propagation (BP) network. Firstly, fuzzy inference system based on the complexity weight matrix of function component was established. Then the adjusted complexity weights were used for modifying the software function point. The adjusted data as samples were transferred to BP network. By the advantage of BP network function approaching, the relationship between software components and software size was established. Finally, BP network was used for estimating software size. The experiment results show that the method could eliminate discontinuity among the different complexity grades, and could make the best of history data, which enhances the accuracy of function point estimation.

Fuzzy inference mainly uses fuzzy logistic method to establish the mapping relationship between input space and output space. It is suitable for describing uncertain phenomena and natural for human understanding. Fuzzy inference contains three main phases: fuzzification,

Fuzzy inference and defuzzification. Fuzzy variables are the basis of fuzzy inference system. Combining the analysis requirement of complexity weight, the DET and RET are the input variable, while the complexity weight of ILF is the output variable.

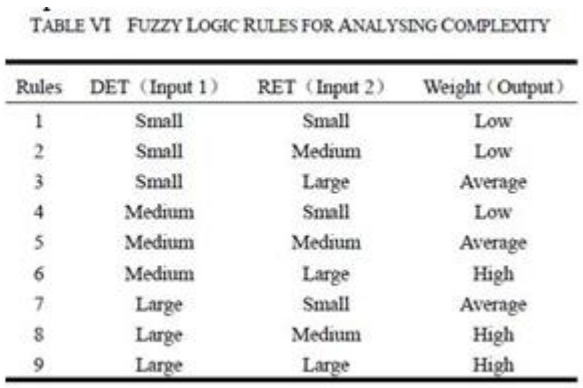
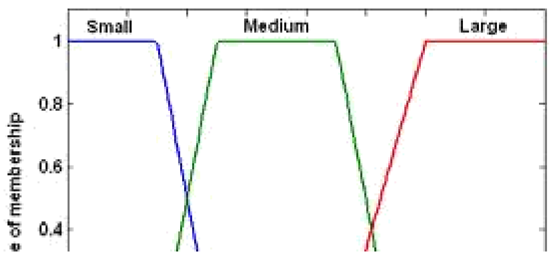
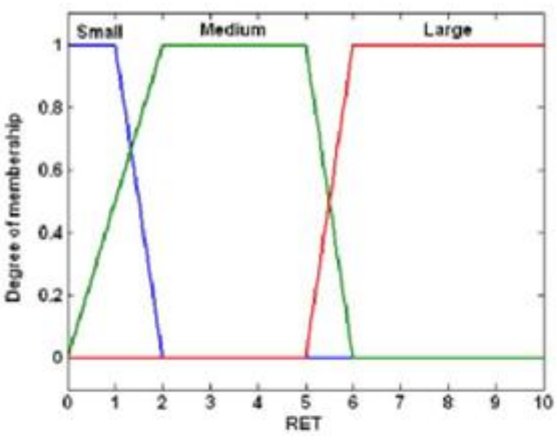
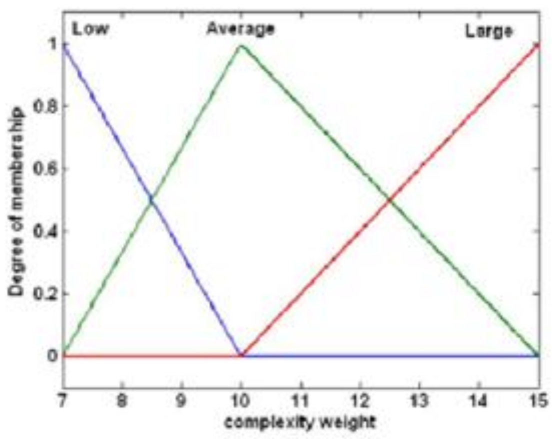


TABLE VI FUZZY LOGIC RULES FOR ANALYSING COMPLEXITY

Rules	DET（Input 1）	RET（Input 2）	Weight（Output）
1	Small	Small	Low
2	Small	Medium	Low
3	Small	Large	Average
4	Medium	Small	Low
5	Medium	Medium	Average
6	Medium	Large	High
7	Large	Small	Average
8	Large	Medium	High
9	Large	Large	High







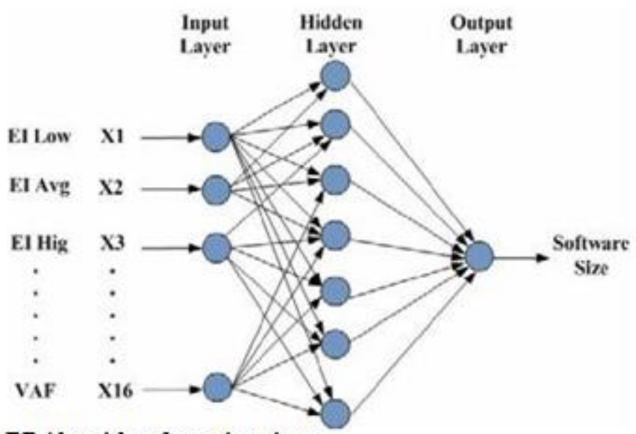
Fuzzy Rules of Complexity Analysis

We use a simple form of fuzzy rules and inference for better human understanding. Each fuzzy rule is of the form "if A then B", where A and B are called an antecedent and a consequent, respectively.

BP Neural Network Theory

The basic idea of BP network is as follows:

BP network is a sort of forward feedback network of owning layer structure. It includes one input layer, one output layer and a lot of connotative layer. It considers the error's square of neural network anticipant output value and the network actual output value as the goal function of study (also called the error function), according to the minimum rule to adjust the network weight, finally the networks' outputs match the pre-specified targets and the goal function reaches the minimum or permitted bound.



BP Algorithm for estimation:

Step 1 Initialization Weights, Learning rate

* the weight value of each layer w_{ij}, max computation epoch R
* the threshold value of each layer θ_j (θ_j is a small random number)
* the learning rate lr, the learning goal

Step 2 Input Samples of BP network

* input samples data $X=(x_1,\cdots,x_{15},x_{16})$, expected output $Y=FP$
* all samples data pretreatment according to Eq. 5.

Step 3 Forward calculating the estimated output of BP network

* calculating output of each layer y_k

$$y_k=f(net_k)=\frac{1}{1+\exp(net_k)} \qquad net_k=\sum_{j=1}^{N} w_{ij}x_j-\theta_j$$

* the nonlinear function f is the typical sigmoid function.
* the practical output $FP=VAF\times\sum\sum Z_{ij}\cdot W_{ij}$ is functionally equivalent to $FP=VAF\times\sum x_i w_i$

Step 4 Calculating the error of each layer

* the error of output layer δ_k $\quad \delta_k=y_k(1-y_k)(y_{pk}-y_k)$
* the error of output layer δ_j $\quad \delta_j=x_j(1-x_j)\sum_{k=1}^{L}\delta_k w_{jk}$

Step 5 Weight update of BP network

* the weight update $\quad w_{ij}(k+1)=w_{ij}(k)+\eta_j\delta_j x_i+x_i(w_{ij}(k)-w_{ij}(k-1))$
* η is a step size to control convergence rate

Step 6 Algorithm Circulation and Termination

* back to step 2, and input another sample to train BP network
* define global error E, When E has fallen within a desire range, the learning process goal is achieved, and the algorithm stops.

$$E=\sum_{n=1}^{N}\frac{w_n}{2}\left[\frac{FP_{estimated}-FP_{actual}}{FP_{actual}}\right]^2$$

V EXTENDED FPA

Software houses are now to keen to provide secure software as requested by customers' desire with respect to security and quality of their products. Significantly security is becoming more and more important in most of the software construct. By engineering security, it will substantially raise the software cost. Security adds a nominal 50 % to the cost,10-90% on effort, and 5-43% on schedules. Therefore estimation should be made earlier on the security costing to overcome the financial shortage in the project management.

Selection of security standards

Software security characteristics formulation:

This formulation considered four common security standards that are widely referred by local software developers and developed over the past decade. It includes Information technology Security Cost Estimation Guide, Common Criteria for Information Technology Security Evaluation, Open Web Application Security Project and Control Objectives for Information and related Technology.

Software security characteristics

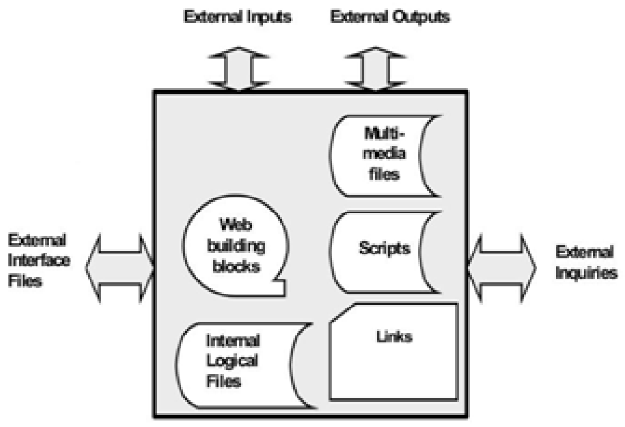
It is organized according to phases in SDLC.

TABLE I. PROPOSED SOFTWARE SECURITY FORMULATION

Step	Security Aspects
Plan (P)	Security Requirements (SR)
Design (D)	Security Features (SF); Functional Features (FF)
Code (C)	Attack Planning (AP); Formal Review and Sign-off (FR); Secure Coding, Review and Audit (SCR)
Test (T)	Software Security Assurance (SSA); Final Security Review (FSR); Infrastructure Application Security Measures (ASM)
Deploy (E)	Software Hardening & Application Security Monitoring (SHA)

VI COMPARISON WITH WEB OBJECTS

Web technologies, once exploited only for creating hypermedia applications (static web sites), have known a huge development, and now they are one of the cutting-edge technologies for very different kinds of systems, from small information centered applications to large enterprise-scale commercial systems. The differences from traditional software, in terms of technology, development model, time-to-market needs, and volatility of requirements and so on, pose serious challenges in adapting traditional size metrics, like Function Points, to measure Web applications. Web Objects [4] size measure can be considered as a significant contribute to web-based applications measurement field. It adds four new elements, related to web development, to the five FP functions then, complexity weighting rules are applied, and the sum of these weights becomes the functional size of the web application. The new elements describe the multimedia content of the pages (Multimedia files), the blocks of different nature that compose the pages (Web Building Blocks), as well as embedded Scripts and Links to external applications and databases.



Web object components

In practice, starting from the requirements of the ten examined systems, we kept the original FP evaluation, and added to

it the specific evaluation of Web application characteristics, according to Web Objects approach (Multimedia files, Web Building Blocks, Embedded Scripts and Links to External Applications and Databases). Clearly, the new estimate using Web Objects is consistently higher than the original FP one, because "points" were added, and never subtracted. To correct for this systematic bias, we computed for each project the percentage increase of Web Object estimate with respect to FP and averaged the ten values. We got a 58% average estimate increase, so we rescaled Web Object estimates, dividing them by 1.58. We kept instead the original productivity coefficient,

varying with the technology adopted in the project.

VII MODEL DRIVEN DEVELOPMENT

Specifically, an algorithm has been implemented for performing the function point count of a software project starting from its conceptual model. Differently from previous approaches, the proposed technique works on the same conceptual model that is used for producing the implementation, eliminating any unnecessary ad-hoc specification task. We evaluate the precision of the FP computation algorithm on a set of real-world projects and de-scribe its implementation within a commercial Model Driven Development tool suite.

With the diffusion of the Unified Modeling Language and of the Model-Driven Architecture, promoted by the Object

Management Group, the software developers' community is paying increasing attention to Model Driven Development (MDD). These methods advocate a stronger automation of the software life-cycle, based on the use of high-level conceptual models of software solutions and on the iterative transformation of high-level models into lower-level platform specific models, until an executable representation of the system is obtained. In the tool market, vendors are

Also committing to MDD by progressively incorporating conceptual modeling and code generation capabilities in their integrated development environments, further contributing to the adoption of model-driven development.

The essential ingredient of the proposed approach is a formal modeling notation suitable both for code generation and size estimation. In this paper, we have adopted the Web Modeling language, a UML profile for representing interactive systems, especially suited for describing Web and Web Service applications. WebML exploits general-purpose UML class diagrams for representing the business objects underlying the application, and a domain specific notation, called hypertext diagrams, for expressing the structure of the application front-end (be it a user interface or a Web Service interface). WebML has enough expressive power to allow the specification of multi-actor, distributed Web and Web service applications and the complete generation of their code for

the JAVA2EE platform. A WebML conceptual model can also be exploited as the input for the computation of the application's function points, according to the well-known IFPUG counting rules.

I. CONCLUSION

The most effective way of evaluating the validity of a size measure is to assess its performance in estimating the development effort required by a software project. This is

Why we chose to apply Function Points. WO estimates are done web applications. The next step of the research will be the development or the choice of a cost model more suited to take as input a size expressed in Web Objects. In that way, we will also be able to compare the obtained results with those of the unique. The future work comprises the application of the developed system to a large collection of projects developed using WebML and Web Ratio, with a twofold purpose: further evaluating and optimizing the precision of the automatic counter, and evaluating the productivity of model-driven development. The latter issue is very promising, as no quantitative data are available on the true benefits of MDD compared to traditional system development. Determining a statistically sound estimation of the FP delivered per staff-months with the MDD approach could foster the adoption of this promising methodology by traditional developers.

An improved function point analysis method for software size estimation is proposed. It mainly is applied to program early. The method uses the fuzzy logical reference to modify the function point. It could eliminate the uncontinuous of the complexity weight. Modified function point as new samples are transmitted to the input layer of BP network. According to BP network, some similar software programs are estimated. The results of research show that the improved method could effectively deal with the flaw in the weight complexity analysis. UML has been recognized as a powerful tool to model the object-oriented software systems. Verification of Function Point Analysis for object-oriented software estimation is done through a case study of Web-based Document Management System. The concept of variable productivity and building a framework based on best unbiased linear estimator. The equivalent weighted least mean square problem is derived and solved it to arrive at an accurate estimate of needed effort for future projects based on the delivered function points. The individual function point elements were not independent. Not all the function point elements were related to effort, an effort prediction model based on two function point elements (input function points and output function points) was just as good as a effort

model based on total function points, an effort prediction model based on the raw counts of the number of files and number of outputs was only slightly worse than an effort model based on total function points. It might be that this dataset is atypical, for example, the results might be different if the data were all from the same company. However, the results indicate that in this case function points do not exhibit the characteristics that would be expected of a valid size metric (which requires the component elements to be independent). In addition, the results suggest that simple counts may be as effective as more complex size models as effort predictors. This is particularly useful because the basic counts are likely to be known reasonably accurately earlier in the life cycle than the weighted counts (which rely on knowing details such as the number of data items involved and tiles accessed by each input). However, use of simple counts does rely on the ability of individual organizations to collect data and generate their own effort prediction models. For example, in this dataset simple counts of number of inputs and number of master files was a reasonable effort predictor but in another dataset different counts might be better. Another advantage of using raw counts might an improvement in counting consistency as a result of simpler counting rules. Kemerer investigated the consistency of function point counts and found that the differences in function point's counts of the same system averaged 12.2%. It is likely that simpler counts would reduce this counting error.

REFERENCES

[1] "Inter-item Correlations among Function Points" by Barbara Kitechenham and Kari Kansala

[2] "Performance of function point analysis through uml modelling"by Dr.Vipin Saxena and Manish Shrivatsava

[3] "Variable productivity adjustment estimation for function point project delivery" by Saeed Bageri,Krishna Ratakonda,Rakesh Mohan

[4] [4]"Extended function point analysis prototype with security costing estimation" by Nur Atuqah Sia Abdullah ,Rusli Abdullah,Mohd Hasan Selamat,Azmi Jaafar.

[5] "Estimation of software project efforts based on function point" by Yinhuan Zheng,Yilong Zheng,Beizhan Zheng,Liang Shi.

[6] "Using function points to measure and estimate real time end embedded software" by Luigi Lavazza and carla garavaglia

[7] "Automating function point analysis with model driven development" by Piero Fraternali,Massimo Tisi,Aldo Bongio

[8] "A software size estimation method based on improved FPA" by

[9] FU Ya-fang, LIU Xiao-dong, YANG Ren-nong, DU Yi-lin, LI Yan-jie

S.Sowmya is working as a Software Development Engineer at ACS, Oracle India. She received her B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. Her research interest includes Natural Language Processing and Computer Networks.

N.Vignesh is working as a Research Associate at Indian Institute of Management, Ahmedabad. He received his B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. His research interest includes Natural Language Processing and Computer Networks.