A Function Point Method for Software Complexity Measurement

De Tran-Cao, Ghislain Lévesque, Jean-Guy Meunier [1]

***Abstract* -This paper introduces a new method for measuring software functional complexity, which is the difficulty degree derived from software functionalities. The method is the type of function point and specification-based as Function Point Analysis (FPA) or COSMIC Full Function Point (COSMIC-FFP). However, unlike other previous proposals that focus on size aspect of software, this paper investigates to software complexity.**

In our previous research, we have proposed a framework in which software complexity is as a vector of three dimensions corresponding to data movement complexity, data manipulation complexity and system complexity. Data movement complexity is measured by the number of data groups (NOD) that move into and out a functional process. Data manipulation complexity is measured by the number of conditions (NOC) on inputs to product different expected outputs and the system complexity is measured by the entropy of system (EOS). The framework and three proposed measures conform to the task complexity model of Wood.

This paper focuses on validating the proposed framework and measures. The empirical study shows that three proposed measures are relevant. Moreover, the effort estimation model, which takes into account both data movement complexity and data manipulation complexity, is better than the model built on the linear regression between maintenance effort and COSMIC-FFP size.

***Keywords*- Complexity measurement, effort estimation, functional complexity measurement, software complexity, task complexity.**

1. Introduction

Function point approach for software sizing was invented by Allan Albrecht in 1979 [ALB79]. The measure of Albrecht - Function Point Analysis (FPA) - is well known because of its great advantages:

- Independent of programming language and technology.
- Comprehensible for client and user.
- Applicable at early phase of software life cycle.

FPA is developed in and designed for Management Information System (MIS). Therefore, there are many other proposals [SYM88, REI90, WHI91, ABR98] that aims to extend it to other kinds of software including real-time, scientific and embedded software.

According to functional measurement methods, software size is measured in terms of *function point* that is the amount of functionalities of software delivered to user. The measurement process to quantify software size begins with identifying the elements (-type) of software from the specification. Then, these elements (-type) are weighted based on their complexity. Therefore, software sizing cannot be independent of software complexity measurement. Unfortunately, software complexity has not been addressed precisely in function point methods. In fact, FPA and some variations (e.g., Feature Point [REI90]) evaluate complexity subjectively. Several methods (e.g., Mark II [SYM88], COSMIC-FFP [ABR98]) try to quantify objectively complexity of element types. But none of them propose a software complexity model or, at least, a theoretical framework for software complexity.

In our previous research [30], we have applied a cognitive approach with which the task complexity model of Wood [31] is used as a theoretical guide to establish a software complexity framework and to propose three simple complexity measures. The complexity investigated in our research is functional complexity. It can be interpreted as the difficulty derived from the functionalities of software and it manifests in development or maintenance effort. Therefore, our measures are likely functional size measures, that is, they take into account the functionalities of software. However, our measures focus on "complexity aspect" rather than "size aspects". In other words, a new function point method for software complexity measurement has been proposed.

The previous research [30] stops at proposing a framework in which software functional complexity is addressed in two parts:

- Component complexity that refers to complexity of the input, output data and the structure of a functional process. This kind complexity is characterized by:
  - Data movement complexity that is measured by the number of data groups (NOD) moved into and out a functional process.
  - Data manipulation complexity that refers to complexity in manipulating inputs to produce outputs and measured by the number of conditions (NOC) on inputs to obtain desired outputs.
- System complexity that refers to complexity derived from relationships between functional processes. It is measured by the entropy of information in the system (EOS).

Therefore, this paper, on one hand, as a continuity of the previous research, aims to validate empirically the framework and measures proposed. On the other hand, the