Chapter 7.10
Formalization Studies in Functional Size Measurement

Barış Özkan
Middle East Technical University, Turkey

Onur Demirörs
Middle East Technical University, Turkey

ABSTRACT

Functional size has been favored as a software size attribute that can be measured early in the software development cycles. Its characteristics of being independent of implementation language, technique and technology promoted the use in software cost estimation and other project management practices. It has been about three decades since Albrecht introduced the concept of functional size and a variety of measurement methods have been developed, some of which have been published by International Organization for Standardization (ISO). Although the concept is recognized in the software community, and there is a growing interest in Functional Size Measurement (FSM), the applications in software organizations have not been common as expected. The problems with FSM method structures and practices have been discussed to be the major factors to explain this situation. This chapter reviews the research papers that propose solutions to the problems with FSM via formalizations in FSM practices or related concept definitions. The associations of the formalization ideas to the abstract software models that represent the view of functionality for FSM methods are of particular interest of the chapter.

INTRODUCTION

Software project managers require knowledge on software product for effective management. Size is one of the key software product attributes and size measurement supports project management processes in various ways. In scope management, the quantified product size can be set as the baseline and the changes in scope can be measured via measuring change in size. In cost management, software effort and time estimates for a software product can be quantitatively figured out following a productivity model based on size attributes.

In contract management, the acquisitions terms of bids, delivery, progress payments and reimbursements can be based on size. The uses can be extended to other project management knowledge areas such as performance, quality and risk measurement (Ozkan, Turetken & Demirors, 2008). When the Service Level Agreements (SLAs) are based on size measurements, then it can be thought that the potentials for uses of size measurements are not limited to project management but can extend to organizational processes like IT demand management. The applicability and uses of such the measures increase as the measure is independent from implementation language, tool, measurer and the organization.

Functional size is a measure of software functional requirements and It has been recognized as one size attribute that shows these characteristics in software research and community. It has been almost three decades since the introduction of the concept and there have been various functional size measurement method and technique developments. Nevertheless, FSM has not been without problems and today, FSM has not been as widely practiced in software community as expected, when its promises are concerned (Symons, 2001). This indicates that there is still work to be done in FSM concept definitions, method structures and the identification of the uses in today's software paradigms.

Some of the problems of FSM have been responded by proposals that introduce formalism into FSM methods and practices. The studies mainly have the common motivations of facilitating a better understanding of FSM methods and their structures and automating the measurements. In this chapter, our primary objective is to give the reader an insight into the challenges in formalizing the FSM methods and an evaluation of the uses of the formalization proposals in FSM. In section two, we familiarize the reader with the concepts of functional size measurement independent from any FSM method. In section three, we give the results of our analysis from a secondary research study, a review of the formalization studies in FSM. We discuss their applicability such that we explore where and how they can help. In the final section, we conclude and investigate the future work opportunities in FSM through formalizations.

BACKGROUND ON FUNCTIONAL SIZE AND FUNCTIONAL SIZE MEASUREMENT

Software size has been associated with several attributes of software artifacts, documents and deliverables and software development practitioners have measured size using a wide range of metrics and methods. Fenton and Pfleeger (1996), defines software size as a multi attribute and describes it in terms of length, functionality and complexity. Among the various approaches developed to software size measurement, the measures and methods on quantifying the 'functionality' attribute have been widely accepted in practice. Software functional size measures the amount of functionality provided to the users. Functional Size Measurement (FSM) methods are mostly utilized in effort and cost estimation for software development and maintenance projects. Estimation errors are reported to be essential causes of poor management (Glass, 2002; Hughes, 2000) and the need for well established estimation models is so imperative that the relation between FSM and software cost, effort and time estimation can easily cause the misinterpretation of FSM methods as estimation models. Although Albrecht (1979) proposed the original idea as an approach to software estimation, the common approach in FSM methods and software community today is to separate FSM and software estimation as related but different concepts. Detailed explanation of size based cost estimation models and techniques can be found in (Matson, 1994; Abran, 2003; Boehm, 2003).

Albrecht introduced "Function Points" (FP) metric and Function Points Analysis (FPA) method for measuring software size. After that, variants

Related Content

Consistency Checking of Specification in UML
P. G. Sapna, Hrushikesha Mohantyand Arunkumar Balakrishnan (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications (pp. 993-1010).*
www.irma-international.org/chapter/consistency-checking-of-specification-in-uml/192910

Hybrid MCDM (Multi-Criteria Decision-Making) Approaches for Sustainable Energy Management
Rachna Ranaand Pankaj Bhambri (2025). *Modern SuperHyperSoft Computing Trends in Science and Technology (pp. 149-184).*
www.irma-international.org/chapter/hybrid-mcdm-multi-criteria-decision-making-approaches-for-sustainable-energy-management/365471

Non-Visual Programming, Perceptual Culture and Mulsemedia: Case Studies of Five Blind Computer Programmers
Simon Hayhoe (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications (pp. 1933-1951).*
www.irma-international.org/chapter/non-visual-programming-perceptual-culture/62554

A Satiated Method for Cloud Traffic Classification in Software Defined Network Environment
Mohit Mathur, Mamta Madanand Kavita Chaudhary (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming (pp. 1509-1528).*
www.irma-international.org/chapter/a-satiated-method-for-cloud-traffic-classification-in-software-defined-network-environment/261087

Lattice Boltzmann Method for Sparse Geometries: Theory and Implementation
Tadeusz Tomczak (2018). *Analysis and Applications of Lattice Boltzmann Simulations (pp. 152-187).*
www.irma-international.org/chapter/lattice-boltzmann-method-for-sparse-geometries/203089