Available online at www.sciencedirect.com
SCIENCE DIRECT
Information and Software Technology 47 (2005) 151–166
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

μ_cROSE: automated measurement of COSMIC-FFP for Rational Rose RealTime

H. Diab, F. Koukane, M. Frappier*, R. St-Denis

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke Que., Canada J1K 2R1

Received 24 September 2003; revised 28 June 2004; accepted 28 June 2004
Available online 3 September 2004

Abstract

During the last 10 years, many organizations have invested resources and energy in order to be rated at the highest level as possible according to some maturity models for software development. Since measures play an important role in these models, it is essential that CASE tools offer facilities to automatically measure the sizes of various documents produced using them. This paper introduces a tool, called μcROSE, that automatically measures the functional software size, as defined by the COSMIC-FFP method, for *Rational Rose RealTime* models. μ_cROSE streamlines the measurement process, ensuring repeatability and consistency in measurement while reducing measurement cost. It is the first tool to address automatic measurement of COSMIC-FFP and it can be integrated into the *Rational Rose RealTime* toolset.

Keywords: Measure; Measurement tool; COSMIC-FFP; Function points; Real-time system; Rational Rose RealTime

1. Introduction

This paper describes μ_cROSE,[1] a CASE tool that automatically measures COSMIC-FFP for Rational Rose RealTime (RRRT) models. COSMIC-FFP is a functional size (FS) measure initially designed for real-time systems [4]. It was inspired from Albrecht's function points (FP), widely used in industry for information systems [2]. COSMIC-FFP addresses some shortcomings of FP for real-time systems and simplifies the measurement task. It should be noted that COSMIC-FFP is not restricted to real-time systems; it can also be used for information systems. It is the first method that conforms to ISO/IEC 14143-1, which specifies a set of generic mandatory requirements for a method to be called a *functional size measurement method*; COSMIC-FFP has also been recently adopted as an ISO/IEC standard (19761).

RRRT is a CASE tool for designing embedded, real-time, and distributed systems. It originates from the acquisition and evolution of the *ObjecTime* toolset by *Rational*. *ObjecTime* was created to support the *Real-Time Object-Oriented Modeling* (ROOM) language [17]. RRRT is mainly used in telecommunications, avionics, and process control. The main reasons for selecting RRRT as the first target for COSMIC-FFP measurement automation are its market penetration and close correspondence with COSMIC-FFP.

FS measures like COSMIC-FFP and FP are applied in several key areas of the software process. The first motivation behind the development of FS measures was to estimate the software development cost (effort), because size is the main factor influencing cost. FS is more relevant than *lines of code* (LOC) to estimate cost early on in the software development process, the latter being measured too late. FP and COSMIC-FFP have been successfully used to build cost estimation models. They can also be used to measure productivity (by computing FS unit per effort unit) or quality (by computing defect per FS unit). An organization may also benchmark its productivity across

* Corresponding author. Tel.: +1-819-821-8000; fax: +1-819-821-8200.

E-mail addresses: hassan.diab@usherbrooke.ca (H. Diab), marc.frappier@usherbrooke.ca (M. Frappier), richard.st-denis@usherbrooke.ca (R. St-Denis).

[1] μ_cROSE is pronounced McRose, and derived from the concatenation of μ, c, and Rose, which stand for measure, COSMIC-FFP, and Rational Rose RealTime.

0950-5849/$ - see front matter
doi:10.1016/j.infsof.2004.06.007

several organizations using FS measures. For instance, the *International Software Benchmarking Standards Group* [12] collects and provides data to its members, while preserving anonymity. Members can benchmark their productivity based on FP or COSMIC-FFP.

COSMIC-FFP can be measured from various documents and at various times during the software process. At the beginning of a project, it is measured from the functional requirements and can be applied in a cost estimation model to estimate cost. Costs derived in the requirements definition phase can be adjusted during design or implementation by re-measuring COSMIC-FFP from design documents or the source code, since the functionalities usually evolve during a project. When a project is coming to an end, the final COSMIC-FFP size and actual effort can be stored in a database of completed projects in order to build or improve a cost estimation model.

Currently, the COSMIC-FFP measurement process is manual, performed by experts who must gather the appropriate information from project documentation. COSMIC-FFP is defined in plain natural language and therefore not specific to a particular functional specification notation. It is formulated in terms of simple concepts like functional process, data group, and data movement. A measurer must interpret these concepts for the specification notation at hand. This interpretation can be subjective (i.e. the measurement may vary depending on the measurer) and may lead to repeatability problems (measurement variance). A recent report on field trials of COSMIC-FFP notes that nearly perfect repeatability is obtained by experienced engineers, but junior engineers show poor repeatability [1]. The measurement rules are subject to interpretation, as noted in a preliminary study based on an experiment with a substantial system [8]. FP is also subject to repeatability problems and interpretations. Studies revealed differences varying from 11 to 30% between several measurements of the same specification by different measurers [9,13–15]. These figures hold when the count boundary is well-defined, but larger variations are observed when boundaries are not clearly delineated. There are various causes for measurement variance: different (acceptable) interpretation of a measure, lack of proper training on measurement, and ambiguous or incomplete functional requirements document.

μ_cROSE improve the COSMIC-FFP measurement process in two ways. First, it almost eliminates measurement costs and advanced measurement training, because measurement is automatic. The measurer only has to identify the subset of the system, expressed as a list of capsules, that has to be measured; μ_cROSE handles all the analysis and calculations of COSMIC-FFP by processing an RRRT model. Second, under the assumption that measurers select capsules that belong to the same layer, μ_cROSE removes measurement variance and ensures perfect repeatability, because the measurement algorithm is completely automated. This algorithm is based on the author's interpretation of the COSMI-FFP definition and has been validated by experts who participated in defining the COSMIC-FFP measure. This interpretation has been formalized in a mathematical definition that is publicly available in a companion paper [5]. μ_cROSE implements this mathematical definition.

The current version of μ_cROSE requires an all-inclusive RRRT model, so that it can be used from the end of the programming phase to measure the actual value of COSMIC-FFP for the corresponding system. This means that μ_cROSE can assist in building a database of completed development projects (containing actual COSMIC-FFP size, actual cost, and other measures) in order to build estimation and defect prediction models, and manage software maintenance and outsourcing, all of which are based on functional software size.

Automation of FS measurement has attracted the interest of both case tools vendors and researchers. According to Mendes et al. [16], most commercial tools provide clerical support for storing FP components and calculating FP from a list of FP components. Eight tools claim to measure FP directly from functional requirements models (e.g. data flow diagrams, entity-relationship diagrams, or object models), but their accuracy has not been independently validated. To the best of our knowledge, no measurement algorithm implemented in any of these tools has been published by vendors or researchers. *HierarchyMaster* FFP is a clerical tool that allows the recording of functional processes and data movements, and the computation of COSMIC-FFP [11]. In contrast, μ_cROSE automatically extract the list of functional processes, data groups, and data movements from an RRRT model.

The rest of this paper is organized as follows, Sections 2 and 3 summarize the main concepts of COSMIC-FFP and RRRT, respectively. They also introduce the terminology used in this paper. Section 4 presents a tour of μ_cROSE, particularly its main functionalities, architecture, and interface. Section 5 contains a discussion about problems inherent to the formalization of the COSMIC-FFP measurement rules, the solutions selected with their consequences, and alternatives. Section 6 gives two class models that capture concepts of COSMIC-FFP and RRRT related to the measurement process and establishes connections between them. Section 7 concerns the validation and verification of μ_cROSE. Finally, Section 8 concludes with a summary of the key points presented in this paper, μ_cROSE limitations, and future directions.

2. A brief overview of COSMIC-FFP

In COSMIC-FFP, a system can be decomposed into layers. Each *layer* represents a level of abstraction that performs a set of functional processes. In this context, software in one layer exchanges data with software in another layer through their respective functional processes.