COSMIC-FFP – Some results from the field trials

Oligny S., Abran A., Symons C.

Abstract

Field trials of the COSMIC-FFP functional size measurement method were initiated at the end of 1999 with the aim of advancing the method from a 'proposal' status to a 'proven' status by demonstrations and tests with real data on development projects from software from a variety of functional domains in a variety of organizations.

Data has been collected in a number of organizations since then and the analysis of the first results started in July 2000. This paper summarizes the context of the COSMIC-FFP field trials and presents some of the key observations obtained to date. Parts of the analysis focused on the relationship between software size and project variables like effort and schedule while other parts of the analysis focused on the relationship between the components contributing to the functional size of the software. Notably the relevance of considering the count of data attributes as a contributor to functional size and the distribution and variation of the size displayed by the functional processes of real-time software was investigated.

The paper concludes on the status of the COSMIC-FFP measurement method, outlining the key events and further results to be expected by early 2001.

1. Context

Quantifying the size of software is generally recognized as one of the key to adequate estimation of effort, cost and schedule of software projects. Source lines of code was the first generally accepted measure for this purpose and is still used extensively, as demonstrated by the many estimation models that have included this measure as a key parameter [1, 2]. As a measure of software size though, the source code measure carries some inherent limitations, and this has been recognized by software engineering practitioners and researchers alike [3]. Among the practitioners, Allan Albrecht was the first to propose, over 20 years ago, a new way of quantifying software size based on the user's view of the software [4]. Albrecht's 1979 method, now referred to as the IFPUG method, is still used today and provides useful results in many organizations, but it also has some limitations and these have been well documented over the past 15 years. One of these limitations is the difficulty of applying such a method outside the MIS domain, as documented by [3, 5, 6, 7, 9, 10, 11, 12, 13, 14]. In 1996, the industry sponsored the development of an IFPUG extension for real-time and embedded software, which was put into the public domain under the name of Full Function Points [8, 15, 16]. This extension enjoyed fair recognition, notably within the telecommunications industry and the embedded software sector of the automotive industry.

Copyright 2000, Common Software Measurement International Consortium (COSMIC)

Building on the strengths of this work and with the support of the industry, the Common Software Measurement International Consortium (COSMIC) was formed in 1998 to design and bring to market a new generation of software measurement methods. The COSMIC group reviewed existing methods (IFPUG, MarkII [17], NESMA [18] and version 1.0 of the Full Function Point methods [8]), studied their commonalties, and proposed the basic principles on which a new generation of software functional size measurement method could be based [19, 20, 21]. In November of 1999, the group published version 2.0 of COSMIC-FFP [23], a measurement method implementing these principles, and put its measurement manual on the Web for public access. Overall, close to 40 people from 8 countries participated in the design of this measurement method. The Measurement Manual, describing the method, is available in English, French and Spanish; Japanese and Italian versions are in preparation. The purpose of this paper is to introduce the COSMIC-FFP functional size measurement method and to present some results of the field trials, gathered over the past year.

2. COSMIC-FFP, A Summary

2.1. COSMIC Key Concepts

From the perspective proposed by COSMIC, software is part of a product designed to satisfy functional user requirements. From this high-level perspective, functional user requirements can be allocated to hardware, to software or to a combination of the two. The functional user requirements allocated to software are not necessarily allocated to a single unit of software. Often these requirements are allocated to pieces of software operating at different layers of specialization and cooperating to supply the required functionality to the product in which they are included. This is illustrated in Figure 1.

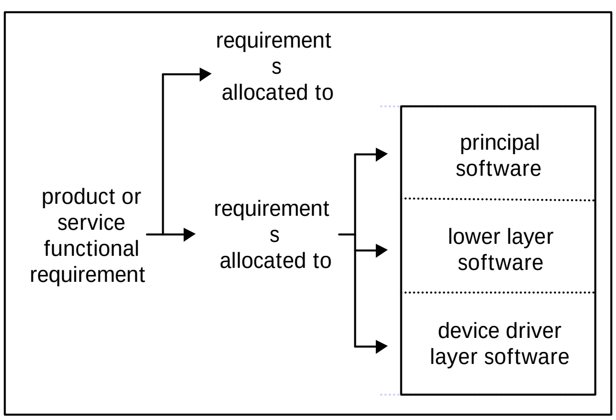


Figure 1 – Allocation of functional user requirements, adapted from [19]

All functional user requirements allocated to any one piece of software can be decomposed into, and represented by, functional processes. In turn, each functional process is represented by sub-processes. A sub-process can be either a data movement type or a data transform type. Version 2.0 of the COSMIC-FFP measurement method recognizes only data movement type sub-processes. Further

Copyright 2000, Common Software Measurement International Consortium (COSMIC)

research is deemed necessary to incorporate data transform sub-process types into the measurement method. In the meantime, an approximation assumption is made such that each data movement his associated with a constant amount of data transformation. This assumption, which should be valid for most MIS, real-time and operating system software, is currently being tested and analyzed through industrial field trials, but will clearly not be valid for algorithm-intensive software as used in, for example, scientific or engineering domains. The COSMIC representation of functional user requirements is illustrated in Figure 2.

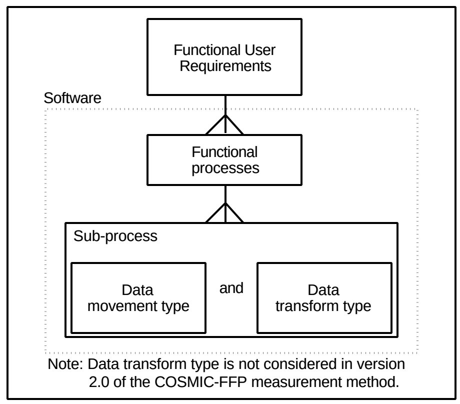


Figure 2 – COSMIC representation of functional user requirements within a piece of software [19]

The COSMIC-FFP measurement method defines four types of data movement which form the basis for expressing the COSMIC-FFP standard unit of functional size. Hence, a generic model is proposed for measuring functional size of any type of software that is not 'algorithm-intensive'. Furthermore, the COSMIC-FFP software model was designed to be compliant with the ISO-14143 standard [22] for software functional size measurement. The COSMIC-FFP generic software model is illustrated in Figure 3.

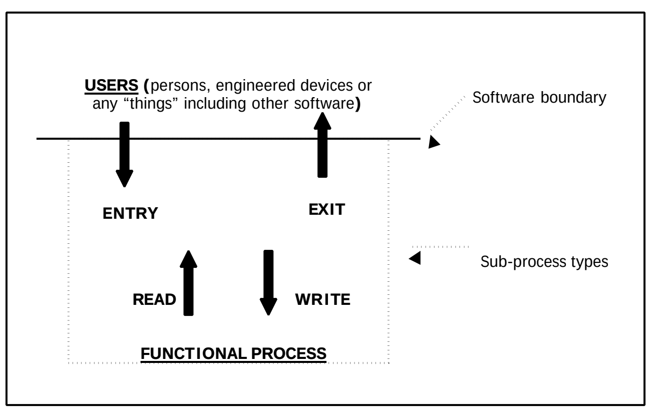


Figure 3 – COSMIC-FFP generic software model

2.2. COSMIC Measurement System

The COSMIC-FFP measurement method explicitly defines a measurement system which includes a measurement principle, base functional components, a standard unit of measure and an aggregation function.

Measurement principle: based on the COSMIC-FFP generic software model (Fig. 3), the method, through a defined set of rules and procedures, produces a numerical figure the magnitude of which is directly proportional to the functional size of this model. These rules presuppose the principle that the functional size of a piece of software is directly proportional to the number of its data-moving sub-processes. By convention, this numerical figure is then extended to represent the COSMIC-FFP functional size of the software itself.

Base functional components: version 2.0 of the COSMIC-FFP measurement method uses only four base functional components: entry, exit, read and write; each one is defined in [27]. Data transform type sub-processes are not used as base functional components due to the approximation assumption described earlier.

Standard unit of measurement: the standard unit of measurement, that is, 1 *Cfsu*, is defined by convention as equivalent to one single data movement type at the sub-process level. A further refinement, still being considered by the COSMIC group, is to define the standard unit of measurement based on the number of data element types moved by a data movement type sub-process. Empirical evaluation of this refinement is analyzed in section 4.3 of this paper.

Aggregation function: The functional size of the data movement base functional components can be combined to obtain the size of any higher-level functional structures such as that of functional processes, components within layers or whole applications. This is performed by arithmetically adding together the functional sizes of the constituent functional structures according to the purpose of the measurement.

The measurement system proposed by the COSMIC-FFP measurement method offers a scalable result, which means that the functional size figure can be constructed at the desired level of abstraction. Furthermore, as demonstrated by Fetcke in [24], the COSMIC-FFP measurement system meets the dominance and the monotonicity properties; two distinctive and desirable measurement properties, violated by some other functional size measurement methods, and contributing theoretically to a) better predictability of effort estimates (in the case of dominance) [24, pp.150] and b) better predictability of functional size itself (in the case of monotonicity) [24, pp. 152].

3. COSMIC-FFP Field Trials

The general aim of the trials was to advance the COSMIC FFP method from a 'proposal' status to a 'proven' status by demonstrations and tests with real data on development and maintenance software

Copyright 2000, Common Software Measurement International Consortium (COSMIC)

projects from a variety of functional domains in a variety of organisations. More specifically, the field trials were organized:

- to test that the COSMIC FFP V.2 documentation is understandable and interpretable in a common and repeatable way across software developers working in different domains, with different methods of expressing requirements and with different development technologies, and can be applied with acceptable effort,

- to test that the functional size measures obtained properly reflect the functionality of the related software requirements as perceived by experts in the requirements and/or that the functional size measures correlate with development effort,

- to enable a full transfer of the COSMIC FFP method to the organisations participating in the field trials, such that they regard it as 'implemented',

- to establish initial benchmark performance levels according to the COSMIC FFP method, against which the participating organisations can compare their own performance.,

- to establish approximation approaches ('estimating rules of thumb') to the full COSMIC FFP method which can be used very early in the life of determining the requirements for an item of software

- to establish, if feasible, conversion rules from existing functional sizing methods.

A protocol was documented to collect specific and standardized data from each participating industrial partner. Collected data was sent to UQAM Software Engineering Management Research Laboratory for centralized analysis.

4. Some Field Trials Results

The preliminary results presented in this section are addressing two specific topics. The first one is an analysis of the variability of the size of functional processes, leading to some insights on the relative limitations of an existing measurement method. The second one is an analysis of the relative contribution of the count of data attributes to the measurement of software functional size. A description of the data sample is presented first.

4.1. Description of the data sample

The data used in the two analysis below are drawn from six software projects within one organization. This organization is a world class manufacturer of real-time systems. Within each measured software only one layer was identified. All projects were completed either in 1999 or early 2000. These software represents a total of 93 functional processes according the COSMIC definition of a functional process. The total size of the six software is 456 *Cfsu*; there was therefore 456 elementary data movements identified at the functional level, out of which it was possible to count the actual number of data attributes moved around in 344 cases.

Copyright 2000, Common Software Measurement International Consortium (COSMIC)

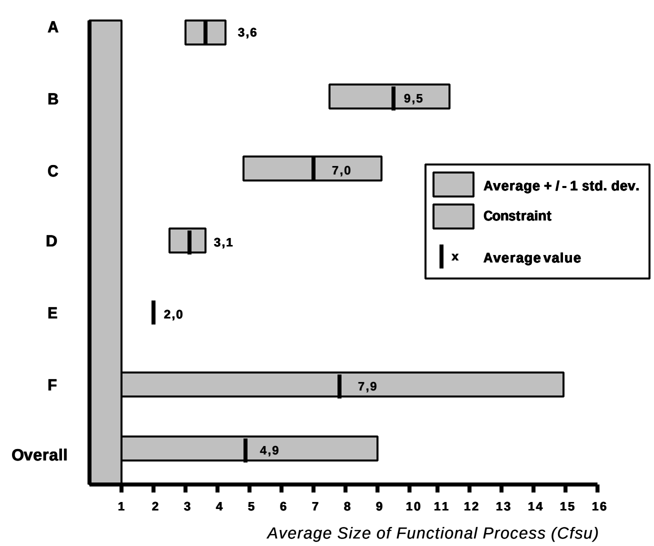
4.2. The size of functional process

An analysis of the average size of these functional processes is presented in Table 1 below. It can be observed that the average size varies significantly from project to project (spread in the order of 5:1). Project E is a singular case, since all four identified functional processes have the same functional size (2 Cfsu). Project B shows the largest average (9,5 Cfsu), while projects F and C also display sizeable functional processes (7,9 and 7,0 Cfsu respectively).

Project ID	No. of Func. Pr.	Software size (Cfsu)	Avg. size of Func. Pr. (Cfsu)	Standard deviation
A	9	32	3,6	0,5
B	8	76	9,5	1,9
C	8	56	7,0	2,1
D	46	142	3,1	0,7
E	4	8	2,0	0,0
F	18	142	7,9	7,1
Overall	93		4,9	4,1

Table 1 – Functional size of the functional processes sample

A further analysis of the distribution of functional size is presented in Figure 4 below. Ranges of size are illustrated by a spread of plus or minus one standard deviation around the average value, and a constraint is illustrated by a smallest size of 1.



Copyright 2000, Common Software Measurement International Consortium (COSMIC)

Figure 4 – Size distribution of functional processes sample

A useful comparison between COSMIC-FFP and the IFPUG measurement methods is based on the following hypothesis: Part of the IFPUG elementary process definition which states that "an elementary process is significant to the users" is equivalent to the COSMIC-FFP definition of functional process which states that "a functional process starts with a triggering event… and ends when producing a result identifiable by the users." If accepted, this hypotheses equates the IFPUG elementary process with the COSMIC-FFP functional process.

Under such circumstances, the range of values that would have been assigned to the functional processes is illustrated in Figure 5 by the vertical shaded box (spanning the sizes of 3 to 7 Cfsu). It can then be observed that, for project B, the IFPUG method would have systematically "undersized" all functional processes, while at the same time "under-sizing" a certain number of functional processes in projects C and F. Yet, in a project like A, all functional processes would have been sized similarly using the IFPUG method. Overall, Figure 5 contributes to illustrating empirically why non-MIS software is reported to be undersized when the IFPUG method is used.

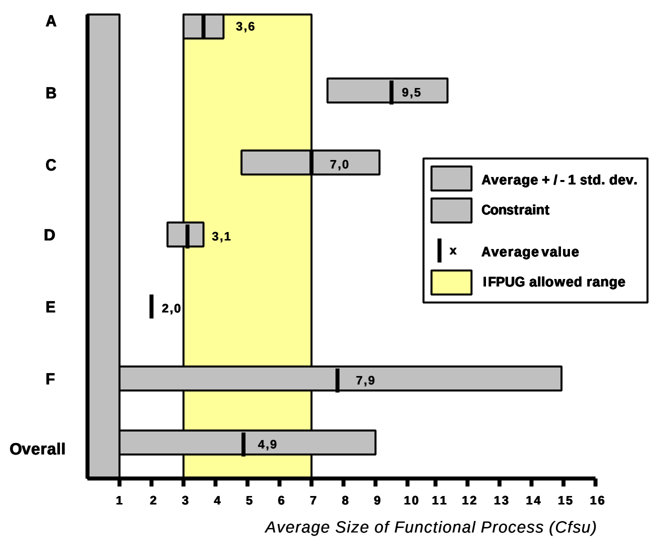


Figure 5 - Size distribution of functional processes sample showing IFPUG limitations

The COSMIC-FFP measurement method was designed to better capture the amount of functionality within a functional process. The above analysis, illustrated in Figure 5, provides corroborative evidence that the design of COSMIC-FFP actually meets its goal: 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 method is much smaller and, therefore, less sensitive to the large diversity of functional processes encountered in real-time software.

4.3. The impact of the number of data attributes

A total of 456 individual data movements were identified within the six projects. During the measurement exercises, effort was expended to identify the number of data attributes moved by each individual data movements. For some data movements this number was estimated, while for others it was counted. It was thus possible to count (not estimate) the number of data attributes involved in 344 data movements (more than 75% of the sample). These are retained here for further analysis. The average number of data attributes involved in a single data movement, along with the standard deviation and number of observations is presented for each data movement type in Table 2 below.

Data Movement type	Average	Standard deviation	Number of observation
ENTRY	3,1	2,9	96
EXIT	2,9	2,7	121
READ	3,5	4,1	63
WRITE	4,7	3,3	64

Table 2 – Basic demographics of the number of data attributes per data movement type

A first distribution analysis, based on a range of plus or minus one standard deviation around the average value for each data movement type, is illustrated in Figure 6 below. In this diagram, the range of possible values is constrained on the lower side by the value 1, since a data movement involves at least 1 data attribute.

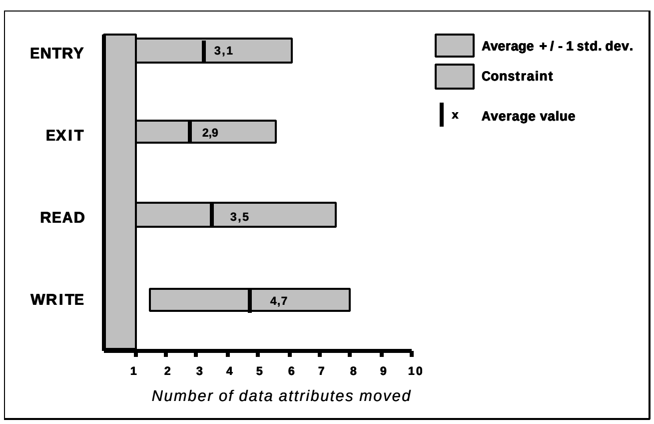


Figure 6 – Basic demographics of the number of data attributes per data movement type

A first statistical test was performed in order to determine whether or not there is a significant difference between the average number of data attributes moved by each type of data movement (null hypotheses). The null hypotheses is rejected at the 0.05 level ($P(H_0) = 0.0025$.

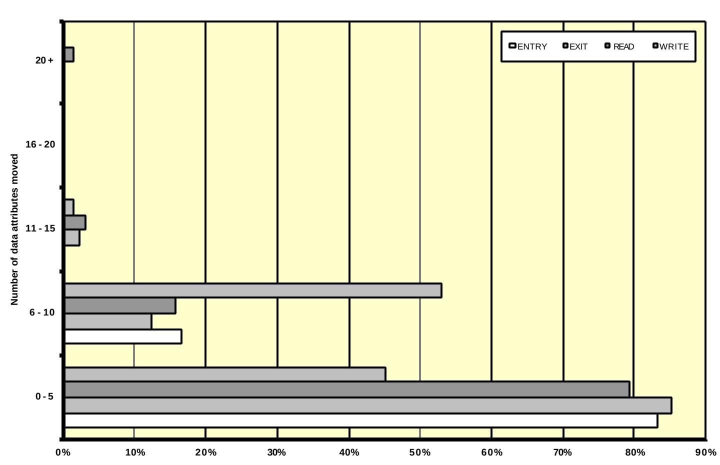
It was therefore appropriate to investigate which data movements type differed from which others regarding the number of data attributes they moved. A prerequisite for a paired t-test is the assumption that the variances are equal across each type of data movement (null hypotheses). This hypotheses was verified using Levene's test successfully, at the 0.05 level (Pr. > F = 0.1882). A paired t-test could thus be applied.

The paired t-test results, shown in Table 3 below, brings out two group of data movement type regarding the number of data attributes moved: on the one hand ENTRY, EXIT and READ can not be statistically distinguished from each other and, on the other hand, WRITE do differ significantly (at the 0.05 level) from the other three data movement types.

	ENTRY	EXIT	READ	WRITE
ENTRY	n.a.	0.6098	0.5188	0.0022
EXIT		n.a.	0.2620	0.0003
READ			n.a.	0.0271
WRITE				n.a.

Table 3 – Pair wise probability of equality between averages (null hypotheses)

A further analysis of the distribution of data movement types was performed, this time to study the proportion of data movement types involving ranges of data attributes. Results are illustrated in Figure 7 below. It can be seen from this analysis that all (100%) of the ENTRYs moved at most 10 data attributes, 98% of the EXITs moved at most 10 data attributes, 95% of the READs moved at most 10 data attributes and 98% of the WRITEs moved at most 10 data attributes.



Copyright 2000, Common Software Measurement International Consortium (COSMIC)

Figure 7 – Distribution of the data movement types for groups of data attributes

Based on these two analyses, it is postulated that assigning an equal weight to all type of data movements is indeed justified given the small magnitude of the differences between the WRITE and the other three types of data movement.

5. Conclusion

Although the quantification of software size is key to an appropriate usage of most estimation model, it is still the object of much research. Technical measure of software size, like lines of code, appeared more than 20 years ago. Although still in use today, they display some limitations. Functional measure of software size, like Albrecht's Function Point, was proposed during the '70s as an alternative to mitigate those limitations but suffer from other shortcomings. In 1998 the COSMIC Group proposed a new functional size measurement method (COSMIC-FFP) to "push the envelope" beyond those known shortcomings.

Notably, The COSMIC-FFP measurement method was designed to better capture the amount of functionality within a functional process. This paper bought forward corroborative evidence that the design of COSMIC-FFP actually meets this goal: 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 method is much smaller and, therefore, less sensitive to the large diversity of functional processes encountered in real-time software.

Furthermore, the analysis of the number of data attributes moved by a sub-process type show that, on average, three types of data movement are undistinguishable in this regard while the magnitude of the difference with the fourth type is small, favoring equal weights between each sub-process types and, therefore, providing an approach to functional size measurement closer to direct observation.

6. ACKNOWLEDGMENTS

The authors of this paper wish to acknowledge the specific contributions of Pam Morris, Grant Rule, Peter Fagg and Denis St-Pierre in the elaboration of the COSMIC-FFP measurement method, the support of Moritsugu Araki, Reiner Dumke and Risto Nevalainen, and the thoughtful and generous comments from all the reviewers of the COSMIC-FFP Measurement Manual [23].

7. BIBLIOGRAPHY

[1] B.W. Boehm, R.W. Wolverton, "Software cost modelling: some lessons learned", *Journal of System and Software*, 1:195-201, 1980.

[2] V. Côté, P. Bourque, S. Oligny, N. Rivard, "Software metrics: an overview of recent results", *Journal of System and Software*, 8:121-131, 1988.

[3] C. Jones, *Applied software measurement - Assuring productivity and quality*, 2nd Edition. New York, NY: McGraw-Hill Inc., 1996.

[4] A.J. Albrecht, "Measuring Application Development Productivity," presented at *IBM Applications Development Symposium*, Monterey, CA, 1979.

[5] S.A. Whitmire, "3D Function Points: Scientific and Real-Time Extensions to Function Points," presented at *Pacific Northwest Software Quality Conference*, 1992.

[6] D.J. Reifer, "Asset-R: A Function Point Sizing Tool for Scientific and Real-Time Systems," *Journal of Systems Software*, Vol. 11, p. 159-171, 1990.

[7] T. Mukhopadhyay and S. Kekre, "Software effort models for early estimation of process control applications," *IEEE Transactions on Software Engineering*, Vol. 18, p. 915-24, 1992.

[8] A. Abran, J.M. Desharnais, M. Maya, D. St-Pierre, P. Bourque, "Design of a functional size measurement for real-time software", Research report no. 13, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, November 1998.

[9] S.D. Conte, H. E. Dunsmore, V. Y. Shen, *Software engineering metrics and models*. Menlo Park: The Benjamin/Cummings Publishing Company, Inc., 1986.

[10] R.B. Grady, *Practical software metrics for project management and process improvement.* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1992.

[11] B. Hetzel, *Making Software Measurement Work - Building an Effective Measurement Program.* Boston: QED Software Evaluation Series, 1993.

[12] S.H. Kan, *Metrics and models in software quality engineering*. Readings, Massachusetts: Addison-Wesley Publishing Company, 1995.

[13] D.C. Ince, "History and industrial application", in N.E. Fenton, *Software metrics: a rigorous approach*, Chapman & Hall, UK, 337 pages, 1991.

[14] S. Galea, "The Boeing Company: 3D function point extensions, v. 2.0, release 1.0", Boeing Information and Support Services, Research and Technology Software Engineering, June 1995.

[15] A. Abran, M. Maya, J.M. Desharnais, D. St-Pierre, "Adapting Function Points to real-time software", *American Programmer*, Vol. 10, no. 11, p. 32-43, November 1997.

[16] D. St-Pierre, M. Maya, A. Abran, J.M. Desharnais, P. Bourque, "Full Function Points: Function Points Extension for Real-Time Software - Counting Practices Manual", Technical Report no. 1997-04, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, September 1997. Downloadable at http://www.lrgl.uqam.ca/ffp.html.

Copyright 2000, Common Software Measurement International Consortium (COSMIC)

[17] C.R. Symons, *Software sizing and estimating – MkII FPA (function point analysis)*, John Wiley & sons, Chichester, UK, 1991.

[18] The Netherlands Software Metrics Users Association (NESMA), “Definitions and counting guidelines for the application of function point analysis, version 2.0”, 1997.

[19] C.R. Symons, P. G. Rule, “One size fits all – COSMIC aims, design principles and progress”, Proceedings of *ESCOM ’99*, p. 197-207, April 1999.

[20] A. Abran, “FFP Release 2.0: An Implementation of COSMIC Functional Size Measurement Concepts”, Proceedings of *FESMA ‘99*, Amsterdam, Oct. 1999.

[21] C.R. Symons, “ COSMIC aims, design principles and progress”, Proceedings of *IWSM ’99*, pp. 161-172, September 1999. Proceedings are downloadable at http://www.lrgl.uqam.ca/iwsm99/index2.html.

[22] International Organization for Standardization (ISO), “ISO/IEC 14143-1:1997 – Information technology – Software measurement – Functional size measurement – Definition of concepts”, October 1997.

[23] A. Abran, J.-M. Desharnais, S. Oligny, D. St-Pierre, C. Symons, “COSMIC-FFP Measurement Manual, version 2.0”, Ed. S. Oligny, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, Oct. 1999. Downloadable at http://www.lrgl.uqam.ca/ffp.html

[24] T. Fetcke, “A Generalized Structure for Function Point Analysis”, Proc. of *the International Workshop on Software Measurement (IWSM ‘99)*, Lac Supérieur, Canada, Sept. 1999, Downloadable at http://www.lrgl.uqam.ca/ffp.html.