A Comparative Study Case of COSMIC-FFP, Full Function Point and IFPUG methods

Vinh T. Ho, Alain Abran, Fetcke Thomas,
Dept. of Computer Science, Université du Québec à Montréal, Canada
vho@lrgl.uqam.ca, abran.alain@uqam.ca

Abstract

Recently, the Full Function Points (FFP) method and its newest development COSMIC-FFP have been developed in order to improve the measurement of functional size for a wide range of software as MIS, Real-Time, Embedded and Technical ones. This paper presents a comparative study of these two functional size measurement methods (Full Function Points and COSMIC-Full Function Points) with respect to the traditional Function Points method (i.e. IFPUG Function Points). The study compares the designs of these three measurement methods through a common framework, from the software models to the measurement processes. A study case on a Warehouse Software Portfolio allows illustrating in detail an empirical comparison on the measurement with these three methods.

1. Introduction

Functional Size Measurement (FSM) methods have been used extensively both in software productivity analysis and in effort estimation. Function Points Analysis (FPA) published in 1979 is an example of a functional measure designed for MIS domain software. The method has been maintained by the International Function Point Users Group (IFPUG) and the current definition of the method is the version 4.0 of the Counting Practices Manual [2]. Though the method is widely used for measuring software size in the MIS domain, it has certain limits for measuring software in real-time and embedded domain. In 1997, Full Function Points (FFP) has been developed as a new functional size measurement method to improve the measurement of various software types: real-time, technical, system and MIS software [6]. Since its introduction (FFP version 1.0, [6]), the method has been passed throughout the measurement field tests and currently used in many organizations [8][9]. The feedback received from the field tests as well as from day-to-day usage of the method, the prospect of ISO certification in accordance with the recently released standard 14143 [7] and the launch of the COSMIC (Common Software Measurement International Consortium) initiative [1][11] in November 1999 have put forward significant improvements in the design of this measurement method. The current definition of the method is documented in the version 2.0 of the COSMIC-FFP Measurement Manual [3]).

A first investigation on the measurement compatibility between the designs of both FFP (version 1.0) and IFPUG (version 4.0) methods was conducted throughout their respective software models and measurement processes [10]. On the software model side (i.e. boundary, user, data object and transactional object), the investigation showed that these software models are compatible either because their concepts are identical or because the concepts proposed by the FFP method are superset or sub-set of the corresponding concepts defined by the IFPUG method. On the measurement process side (i.e., object to be measured, measurement functions and aggregate function), it indicated that compatibility between the two measurement processes is established according to two ranges of value (which are used to determine the contribution): 1) within the range of value introduced by the IFPUG method, both measurement processes are compatible at the "transactional" object level; 2) outside this range of value the compatibility is founded under certain conditions.

This paper extends the previous investigation by examining the COSMIC-FFP (version 2.0) throughout the same comparative framework. It presents the evolution of the designs of functional size measurement methods from the IFPUG to FFP and to COSMIC-FFP methods. In addition, a comparative study conducted on a Warehouse Software Portfolio example is presented to illustrate the measurement using the three methods. This study is helpful for readers interested in applying Function Points methods.

The paper is organized as follows: section 2 compares the designs of the three methods (IFPUG, FFP and COSMIC-FFP); section 3 presents a comparative study case conducted on a Warehouse Software Portfolio example; discussion and future research directions follow in section 4.

2. Comparison of designs

2.1. A common framework for comparison

As a FSM method, three measurement methods in question measure the functional size of a software by quantifying its Functional User Requirements (FURs). First, the FURs are either explicitly (as in case of FFP and COSMIC-FFP) or implicitly (as in case of IFPUG) mapped into a specific software model that captures the concepts and definitions required for the FURs (the so-called mapping phase). Then, a set of measurement rules and procedures defined by the measurement method is applied to measure specific features of this software model (the so-called measuring phase). The result is a numerical figure representing the functional size of the software. For comparative purpose, a common framework was proposed in [10] and illustrated in Figure 1.1. It should be noted that two measurement methods are considered as compatible if their respective software models and measurement processes are compatible.

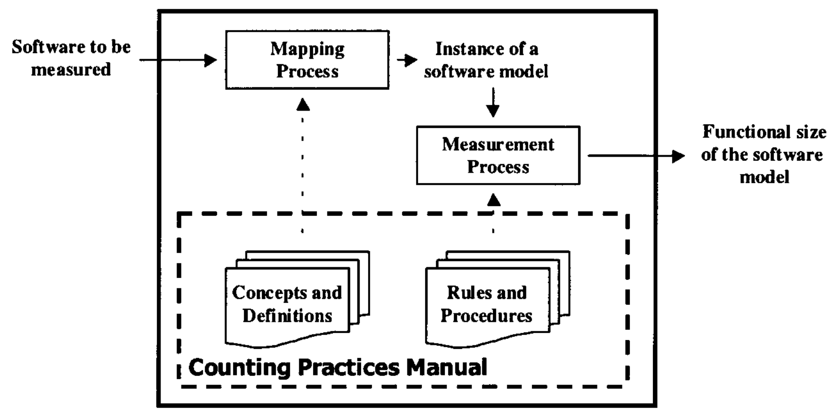

Figure 1.1. *A common framework for comparing IFPUG, FFP and COSMIC-FFP methods [10]*

2.2. Comparing software models

In order to compare the software models of three measurement methods, fundamental concepts characterizing each software model are identified: four concepts for the IFPUG method and five for both the FFP and COSMIC-FFP. Based on their definition according to the Method Manual, these concepts are compared in pair as presented in Table 2.1.

Table 2.1. *Concepts characterizing the software models defined by IFPUG, FFP and COSMIC-FFP methods*

CONCEPTS	IFPUG	FFP	COSMIC-FFP
Boundary	Boundary	Boundary	Boundary
Users	Users	Users	Users
Data objects	Logical Files	Group of Control Data	Group of Data
Process objects	Elementary Process	Control Process	Functional Process
Sub-process objects	N/A	Sub-process	Sub-process

2.2.1. Boundary

For all three, the boundary concepts are clearly defined and used to determine what is included in the software to be measured. Conceptually, they are all compatible.

2.2.2. User

The user concept defined in the COSMIC-FFP and FFP methods includes and extends the concept defined in the IFPUG method to encompassing other software and hardware devices interacting with the software to be measured. Conceptually, they are all compatible.

***Table 2.2.** Data and Process concepts defined by IFPUG, FFP and COSMIC-FFP methods*

CONCEPTS	IFPUG	FFP	COSMIC-FFP
Data objects	ILF: Internal Logical File EIF: External Interface File	RCG: Read-Only Control Group UCG: Updated Control Group	Data Group
Process objects	EI: External Input EO: External Output EQ: External Inquiry	Control Process	Functional Process
Sub-process objects	N/A	ECE: External Control Entry ECX: External Control Exit ICR: Internal Control Read ICW: Internal Control Write	E: Entry X: Exit R: Read W: Write

2.2.3. Data object

All three methods identify data objects from the logical and functional perspectives and distinct them from technical and implementation considerations. Both the FFP and IFPUG distinguish between data objects that are "read-only", i.e. ILF and RCG, and that are updated, i.e. EIF and UCG (Table 2.2). For the FFP method, each of its two data groups can be of two types: multiple-occurrence or single-occurrence. A multiple-occurrence group data displays a typical MIS file structure and is treated accordingly using IFPUG rules. A single-occurrence group data is a newly defined data type that is to capture single-occurrence control data encountered often in real-time software. Conceptually, FFP data object concept is a superset of IFPUG data objects. However, the current COSMIC-FFP does not distinguish anymore types between data groups. It uses the data group concept to represent all functional data groups. In fact, the method defines the data group concept in a higher level of functional abstraction than that is defined in both the FFP and IFPUG methods.

2.2.4. Process object

All three methods imply that software's functionality is implemented on a set of process objects, i.e., transaction, control or functional process (Table 2). The IFPUG distinguishes explicitly three kinds of transactions: EI (External Input), EO (External Output) and EQ (External Inquiry). In contrast, both the FFP and COSMIC-FFF do not classify process objects (they are called control process or functional process respectively) and define process as a set of data movement sub-processes. In both the FFP and COSMIC-FFP methods, the software to be measured is considered in a deeper level of functional abstraction (i.e. at sub-processes level) than in the IFPUG method.

2.2.5. Sub-Process

The sub-process concept defined in both the FFP and COSMIC-FFP is strictly a refinement of the process concept (Table 2.3). With this concept, the software's functionality is considered at a level of granularity below the lowest level defined by the IFPUG method. That makes both the FFP and COSMIC-FFP methods possible to capturing adequately characteristics of real-time software, and it is therefore an important consideration in the measurement of the functional size of real-time software. It fact, it is the main improvement in the design of these two methods over the IFPUG method for measuring functional size of real-time software.

Furthermore, the COSMIC-FFP method has put forward an important improvement, with respect to the FFP method, by simplifying its base functional components (BFC). It includes only four BFCs: Entry, Exit, Read and Write data movement sub-process. This leads to a more rigorous definition of the elements contributing to the functional size of a software.

Table 2.3. *BFC concepts used in the IFPUG, FFP and COSMIC-FFP methods*

IFPUG	FFP	COSMIC-FFP
Transaction Type (EI, EO, EQ)	Control	Transaction Type
	Function Type	Data Movement Type
I/O parts of EI, EO and EQ	Entry / Exit	Entry / Exit
References to ILFs and EIFs	Read / Write	Read / Write

By comparing the software models of three measurement methods, it is observed that:

- The software model of the FFP is conceptually a superset of that of the IFPUG. It takes into account the specific functional characteristics of software at a finer level of granularity, the sub-process level, while the IFPUG considers only at the process level. That is an important improvement in the method's design, making it possible to measure adequately characteristics of real-time software.
- The COSMIC-FFP method has improved the software model of the FFP method and put it in a higher level of functional abstraction. On the one hand, data objects is unified in only one concept: data group. Type of data is no longer distinguished. On the other hand, the definition of the transactions is simplified and only four data movement types are defined as BFCs: Entry, Exit, Read and Write data movement sub-processes. That puts forwards the method to comply with the ISO standards defined for FSM methods [7].

2.3. Comparing measurement process

The measurement processes are evaluated on the basis of three concepts: 1) the objects to be measured or to be received "points"; 2) the measurement functions assigning "points" to the objects to be measured; and c) the aggregation functions.

2.3.1. Objects to be measured

a) Data objects

The data objects are measured in both the IFPUG and FFP methods. In contrast, the COSMIC-FFP method does not account the data objects and they are, in fact, used as a means to identify data movement sub-processes.

b) Process objects

The IFPUG method considers its three transactions (EI, EO, EQ) as elementary objects to be measured, that is, the measurement is conducted at the process level. Both the FFP and COSMIC-FFP do not conduct the measurement at the process level but at the sub-process level because their process objects are composed of a set of sub-processes (section 2.2.5).

c) Sub-Process objects

Sub-processes are considered as elementary objects to be measured in both the FFP and COSMIC-FFP methods – that is the measurement is conducted at a lower level than in the IFPUG method.

It is worth noting that compatibility on the process objects must be evaluated at the same level of granularity – i.e. the process object level.

2.3.2. Measurement and aggregation functions

Based on a detail comparison between the IFPUG and FFP method, it is indicated in [10] that the IFPUG measurement process offers a bounded range of functional values while the FFP measurement process does not display that characteristics. Compatibility of the two methods can therefore be established according to two ranges of value (used to evaluate their contribution):

- Within the range of value defined by the IFPUG method, both measurement processes are compatible at the "process object" level.
- Outside this range, both measurement processes are compatible under the following conditions:
  a) Results are considered at the "process object" level;
  b) The range of values for FFP measurement functions RCG and UCG (Table 2) is an extrapolation of that defined for ILF and EIF by the IFPUG;
  c) The range of values for FFP measurement functions ECE, ECX, ICR, and ICW (Table 2) is an extrapolation of that defined for EI, EO and EQ by the IFPUG.

At this point, the COSMIC-FFP has put forwards an important improvement. Unlike the IFPUG and FFP methods, the COSMIC-FFP method defines that the functional size of a software is directly proportional to the number of its BFCs (i.e. entry, exit, read and write data movement sub-processes). By virtue of this definition, the method offers a scalability for the measurement results throughout its software model's levels of granularity: sub-process, functional process, layer and software. Therefore, the measurement result (i.e. the functional size) of an "object" at a level of granularity is simply defined as the arithmetic sum of the measurement results of its constituent objects at an immediately lower level of granularity. For example, the functional size of a functional process is defined as the arithmetic sum of the values of the measurement function, as applied to each of its constituent sub-processes.

By convention proposed in the COSMIC-FFP method, each data movement sub-process is assigned 1 CFSU (Cosmic Functional Size Unit). Since each functional process must have at least two data movement sub-processes: 1 entry and 1 exit or write, its smallest theoretical functional size is 2 CFSUs. However, there is no upper limit to the functional size of a functional process as that fixed in both the IFPUG and FFP methods.

2.4. Evolution of FSM methods from IFPUG to FFP and to COSMIC-FFP

The FFP method has been designed in alignment with the fundamental concepts of the IFPUG one. New data and transactional functional types are introduced making the FFP method possible to measure adequately characteristics of real-time software while reserving its ability to measure effectively characteristics of MIS software. Conceptually, the FFP method can be considered as a superset of the IFPUG method.

The COSMIC-FFP method is a result of the improvement of the FFP one in order to satisfy, among other,

- the need to simplifying the set of BFC used to represent software functionality,
- the need to increasing the flexibility of the measurement methods in order to offer a scalable result.

On the one hand, its design is put forwards onto a higher level of functional abstraction than that of the FFP method. Therefore, it offers a "better" flexibility for the measurement of software through many levels of functional abstraction as functional process and layer. On the other hand, the simplification of the BFCs (only four BFCs are used) allows consolidating its theoretical basis and clarifying the elements contributing to the functional size of a software. Conceptually, its four BFCs (entry, exit, read and write sub-processes) are equivalent to the four control sub-processes ECE, ECX, ICR, and ICW defined in the FFP method. Since the COSMIC-FFP method does not account the contribution of data objects in the measurement results, one can expect some closeness on the contributions of the process objects identified and measured with both methods (FFP and COSMIC-FFP) when the range of values proposed in the FFP method is not applied to assigning the points.

In the next section, a measurement exercise is conducted on the same MIS software example with the three methods IFPUG, FFP and COSMIC-FFP in order to illustrate the previous analysis.

3. Study Case: Measuring an Warehouse Software Portfolio example with IFPUG, FFP and COSMIC-FFP methods

3.1. Description of the Warehouse Software Portfolio (WSP) system

The system is a portfolio of software for warehouse management [5]. For this system, three data entity types are identified: customers, items and storage places (Fig. 3.1) where key attributes are set in bold and foreign keys are set in italic and underline.

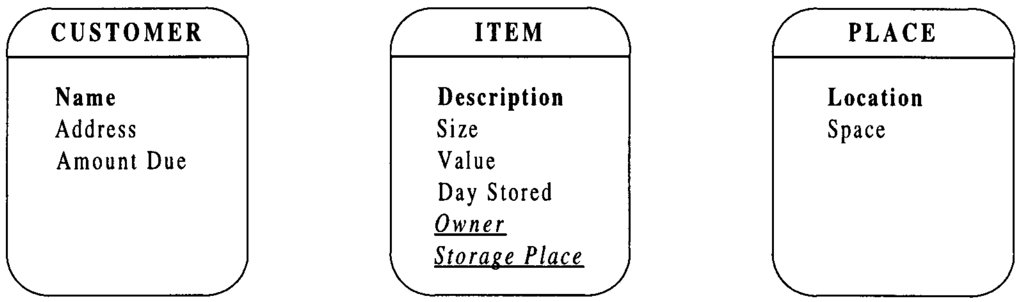

***Figure 3.1**. Three entities of the warehouse software [5]*

Two relationships are defined between three entity types (Fig. 3.2):

- *Owns* relationship: A CUSTOMER may *own* several ITEMs and each ITEM belongs to a single CUSTOMER. This relationship is represented by the foreign key attribute *Owner* in the ITEM entity type.
- *Stored at* relationship: Each ITEM is *stored at* a single storage place, while a PLACE can accommodate several ITEMs, provided that it has enough SPACE. This relationship is represented by the foreign key attribute *Storage Place* in the ITEM entity type.

In the WSP, sixteen transactions are defined for managing the warehouse (Table 3.5) [5].

***Figure 3.2**. Entity-Relationship defined in the warehouse software portfolio [5]*

3.2. Illustrative measurement details

For illustrative purpose, the measurement details are presented through the comparative framework used in the previous analysis.

3.2.1. Boundary and user

All entities and transaction are included in the application and therefore are considered for measuring. The user is identified as a person or a software that interacts with the application.

3.2.2. Data object

All entities are identified as data objects (Table 3.1). It should be noted that, hereafter, all results to be presented are the unadjusted function points. Both the IFPUG and FFP method assign the same points to the identified data objects. However, the COSMIC method does not account the data objects in the measurement results.

***Table 3.1**. Function points of the data objects*

Data Object	**IFPUG**				**FFP**			
	Type	**#RET**	**#DET**	**Point**	**Type**	**#RET**	**#DET**	**Point**
Customer	ILF	1	3	7	UCG	1	3	7
Item	ILF	1	6	7	UCG	1	6	7
Place	ILF	1	2	7	UCG	1	2	7
Total				**21**				**21**

3.2.3. Process objects

The measurement details of three process objects related to data input, output and query operators are presented as example.

a) *Add Customer* transaction (Table 3.2): receives the name and address of the customer, verifies if a customer with the name entered already exists, displays an error message if the operation fails, updates the Customer file. For the IFPUG method, it is identified as an EI. For both the FFF and COSMIC-FFP, the following sub-processes are identified: receive (*ECE* or *E*), verify (*ICR* or *R*), display (*ECX* or *X*) and update (*ICW* or *W*). In terms of points, the transaction is assigned the same points (4) by both FFP and COSMIC-FFP, but 3 by the IFPUG.

***Table 3.2**. Measurement details of the* Add Customer *transaction*

Process Objects	**IFPUG**				**FFP**			**COSMIC-FFP**	
	Type	**#DET**	**#FTR**	**Point**	**Type**	**#DET**	**Point**	**Type**	**Point**
Process	EI	4	1	**3**	N/A		**4**	N/A	**4**
Sub-process	N/A				ECE	2	1	E	1
					ICR	1	1	R	1
					ECX	1	1	X	1
					ICW	3	1	W	1

b) *Print Bill* transaction (Table 3.3): prints the bill for a customer, including Name, Address, Amount due and the total number of items owned by the customer. For the IFPUG method, it is identified as an EO. For both the FFF and COSMIC-FFP methods, the following sub-processes are identified: receive customer's Name (*ECE* or *E*); retrieve Customer data including Name, Address, Amount due (*ICR* or *R*), read Item data (*ICR* or *R*); display error message (*ECX* or *X*); print bill including Name, Address, Amount due and Total Items (*ECX* or *X*) . In terms of points, the transaction is assigned the same points (5) by both FFP and COSMIC-FFP, but 4 by the IFPUG.

***Table 3.3**. Measurement details of the* Print Bill *transaction*

Process Objects	**IFPUG**				**FFP**			**COSMIC-FFP**	
	Type	**#DET**	**#FTR**	**Point**	**Type**	**#DET**	**Point**	**Type**	**Point**
Process	EO	4	2	**4**	N/A		**5**	N/A	**5**
Sub-process	N/A				ECE	1	1	E	1
					ICR	3	1	R	1
					ICR	1	1	R	1
					ECX	1	1	X	1
					ECX	4	1	X	1

c) *Query Customers* transaction (Table 3.4): displays the CUSTOMER data for a given customer Name entered by the user. If a record of customer exists for the Name, the

Address and Amount Due for this customer are displayed. Otherwise, an error message is displayed. For the IFPUG method, it is identified as an EQ. For both the FFF and COSMIC-FFP, the following sub-processes are identified: receive customer's Name (*ECE* or *E*); retrieve Customer data including Name, Address, Amount due (*ICR* or *R*); display error message (*ECX* or *X*); display customer data including Name, Address and Amount due (*ECX* or *X*) . In terms of points, the transaction is assigned the same points (4) by both FFP and COSMIC-FFP, but 3 by the IFPUG.

Table 3.4. *Measurement details of the* Query customers *transaction*

Process Objects	IFPUG				FFP			COSMIC-FFP	
	Type	#DET	#FTR	Point	Type	#DET	Point	Type	Point
Process	EQ	5	2	**3**	N/A		**4**	N/A	**4**
Sub-process	N/A				ECE	1	1	E	1
					ICR	3	1	R	1
					ECX	1	1	X	1
					ECX	3	1	X	1

For the sake of simplicity, the measurement results for all sixteen transactions in the WSP are summarized in Table 3.5. It should be noted that, in the context of the WSP, the transactions from the requirements of the WSP correspond to transactional (process) function types in both the FFP and COSMIC-FFP methods. It implies that sixteen transactions of the WSP are functionally identical from the view of three measurement methods.

Table 3.5. *Measurement results of sixteen transactions in the WSP*

No.	Name	IFPUG	FFP	COSMIC-FFP
1	Add Customer	3	4	4
2	Change Customer	3	6	6
3	Delete Customer	3	5	5
4	Receive Payment	3	4	4
5	Deposit Item	6	5	5
6	Retrieve Item	4	6	6
7	Add Place	3	4	4
8	Change Place Data	3	6	6
9	Delete Place	3	4	4
10	Print Customer Item List	5	4	4
11	Print Bill	4	5	5
12	Print Stored Items List	4	3	3
13	Query Customers	3	4	4
14	Query Customer's Items	3	5	5
15	Query Places	3	4	4
16	Query Stored Items	3	5	5
Total		**56**	**74**	**74**

Throughout the example, it is observed that:

- On the data side: both the IFPUG and FFP methods identified the same data objects and they are consequently assigned the same points because the same rules are applied. The COSMIC-FFP method also identifies these data objects but it does not account them for measurement result.
- On the process side: both the FFP and COSMIC-FFP methods identified an identical set of sub-processes and, consequently, the points assigned for the processes are identical if the number of DET is not out of the range of value (i.e. <19). However, the IFPUG method assigns less points to the process than both other two methods. This can be explained by the fact that two other methods capture the functionality at a lower level of functional abstraction. This implies that, generally, both method will give more points than the IFPUG method. In addition, the functionality is

represented more explicitly with the FFP and COSMIC-FFP, and therefore it is easier for the measurer to validate.

3.3. Measurement results for applications extracted from the WSP

From the WSP, many applications can be extracted according to specific requirements from clients. Each application includes some elements of the WSP (entities and transactions) depending on its requirements. For this study, five application are identified and used as examples:

- *WSP* (named **W**): the WSP is an application itself. Therefore, all entities and transactions must be considered for measuring with FSM methods.
- *Application* **M**: manage the items stored in the warehouse: deposit and retrieve items; add, modify and delete place; query places and stored items.
- *Application* **C**: manage the customers and their ownership of items: add, modify and delete customer data; deposit and retrieve items; query customers and customer's items.
- *Application* **LC**: manage customer data, send bills to customers and handle the storage and retrieval of items.
- *Application* **LS**: can add, modify and delete storage places, query places or the items stored at a specific location, or print an inventory of a warehouse.

The measurement results for these 5 applications are summarized in Table 3.6. The functional sizes of the applications are all different depending mostly on the transactions which are included in the applications. It is observed that:

- On the data side: both the IFPUG and FFP methods distribute the same points to the data objects.
- On the process side: both FFP and COSMIC-FFP distribute the same points for all five applications. However, the IFPUG method distributes less points, in order of 25 %, to the processes (transactions) than other two methods.

Table 3.6. Measurement results of 5 applications extracted from the WSP

No.	Applications	IFPUG			FFP			COSMIC-FFP		
		Data	**Process**	**Total**	**Data**	**Process**	**Total**	**Data**	**Process**	**Total**
1	**W**	21	56	77	21	74	95	N/A	74	74
2	**M**	14	26	40	14	34	48		34	34
3	**C**	14	35	49	14	47	61		47	47
4	**LC**	19	37	56	19	48	67		48	48
5	**LS**	12	19	31	12	26	38		26	26

4. Discussion and conclusion

This paper presents a comparative study on the design of three functional size measurement methods: IFPUG, FFP and COSMIC-FFP. The study is conducted through a well-defined framework: from the software models to the measurements processes. It shows that the software models of both FFP and COSMIC-FFP are conceptually close and they capture the functionality of software at a lower level of functional abstraction in comparison to that of the IFPUG method. It implies theoretically that both FFP and COSMIC-FFP would distribute more functional size points to software than the IFPUG method. This difference is observed through the study case on a MIS software system. In addition, the study case illustrates in detail the measurement excise with three measurement methods allowing readers to have a practical comparison.

Other empirical studies [8][9] have also demonstrate that the FFP method adequately measures the functional size of software in both domains: MIS and real-time. Presently, many field tests are conducting with the COSMIC-FFP in many industrial organizations in order to evaluate its properties [Recent report on the field test]. It is worth noting that a measurement method, as FFP or COSMIC-FFP, which can measure adequately a wide range of software (e.g. MIS, real-time, embedded and technical software) will offer great benefits in the economics of an organization's software process management.

Acknowledgment

5. Reference

[1] Abran, A., "FFP Release 2.0: An Implementation of COSMIC Functional Size Measurement Concepts", Presented at FESMA 99, Amsterdam, Oct. 4-7, 1999.

[2] Abran, A., et al., 1998, "Design of a Functional Size Measurement for Real-Time Software", Research Report No13, Dept. of Computer Science, Université du Québec à Montréal

[3] Abran, Desharnais, Oligny, St-Pierre, Symons, "COSMIC-FFP Measurement Manual, version 2.0", Ed. S. Oligny, Software Engineering Management Research Lab., Université du Québec à Montréal (Canada), Oct., 1999.

[4] UQAM, "*Full Function Points: Counting Practices Manual*", Software Engineering Management Research Laboratory, Université du Québec à Montréal, Technical Report no. 1997-04, September 1997.

[5] Fetcke, T., 1998, "The Warehouse Software Portfolio", First Draft, ...

[6] IFPUG, 1994, "*Function Points Counting Practices Manual – Release 4.0*", International Function Point Users Group (IFPUG), Westerville, Ohio, USA, January 1994.

[7] ISO/IEC 14143-1: Information technology – Software measurement – Functional size measurement – Definition of concepts, October 1997.

[8] Morris P., Desharnais J.M., "Measuring all software, not just what the business uses", Proceedings of the IFPUG Fall Conference, Orlando, Florida, September 21-25, 1998.

[9] Oligny S., Abran A., Desharnais J.M., Morris P., "Functional size of real-time software: overview of field tests", Proceedings of the 13th International Forum on COCOMO and software cost modeling, Los Angeles, California, October 6-8, 1998.

[10] Oligny, S., Abran, A., "On the Compatibility between Full Function Points and IFPUG Function Points", Proceedings of the 10th European Software Control and Metric Conference (ESCOM SCOPE 99).

[11] Oligny, S., Abran, A., St-Pierre, D., "Improving Software Functional Size Measurement", Proceedings of COCOMO and Software Cost Modeling International Forum 14, USA, Oct., 1999.