Measurement of Functional Size in Conceptual Models: A Survey of Measurement Procedures Based on COSMIC*

Beatriz Marín, Giovanni Giachetti, and Oscar Pastor

Centro de Investigación en Métodos de Producción de Software,
Universidad Politécnica de Valencia,
Camino de Vera s/n,
46022 Valencia, Spain
{bmarin,ggiachetti,opastor}@pros.upv.es

Abstract. Many functional size measurement procedures have been developed for applying the COSMIC measurement method to particular methods of software production. A subset of these measurement procedures is centered on the measurement of the functional size of the applications from their conceptual models, allowing the generation of indicators in early stages of the development cycle of a software product. This paper presents a survey of these functional size measurement procedures in order to provide a guide for practitioners and researchers. Finally, a general analysis focused on the results obtained in the survey is performed to obtain important lessons that must be considered in the development of correct measurement procedures.

Keywords: Functional Size Measurement, Functional Size Procedures, COSMIC, Conceptual Models.

1 Introduction

Nowadays, it is widely accepted that it is essential to know the functional size of applications in order to successfully apply estimation models, effort models, and budget models [33]. This knowledge will allow the project leader to generate indicators to facilitate project management. To measure the functional size of software applications, four measurement methods have been recognized as standards: IFPUG FPA [22], MK II FPA [23], NESMA FPA [24], and Cosmic FFP [21]. The first three methods are based on the Function Point Analysis proposal [1], which takes into account only the functionality of the system that the human user observes. These FPA-based methods have several limitations for the correct measurement of systems: for instance, they only allow the measurement of Management Information Systems, which excludes the measurement of other types of software (such as real time software); they have units that are hard to understand; they do not consider the

* This work has been developed with the support of MEC under the project SESAMO TIN2007-62894 and co financed by FEDER.

R. Dumke et al. (Eds.): IWSM / MetriKon / Mensura 2008, LNCS 5338, pp. 170–183, 2008.

functionality that allows communication between layers in systems with a layer-based architecture, etc. To overcome the limitations of FPA-based measurement methods, the COSMIC measurement method was defined.

In addition, software production processes have evolved from focusing essentially on the solution space (software product) to focusing on the problem space (conceptual models). The new software production processes are based on MDA (Model Driven Architecture) approaches [35], which allow the generation of the applications by means of model transformations. In these technologies, the conceptual models are a key resource that allows the partial or complete generation of the final software product. Consequently, the measurement of the functional size in the conceptual models allows the project leader to generate indicators in early stages of the development cycle of a software product.

Taking into account this situation, many proposals have been defined to measure the functional size of software applications from conceptual models. The aim of this work is to present a broad survey of the existing literature related to functional size measurement procedures based on COSMIC that can be applied to conceptual models. This survey includes the following proposals: Bévo et al. [7], Jenner [25], Diab et al. [15], Poels [39], Nagano et al. [36], Azzouz et al. [5], Condori-Fernández et al. [12], Habela et al. [19], Grau et al. [18], Levesque et al. [29], and Marín et al. [31].

In this paper, we summarize the proposals based on the COSMIC measurement method according to the following criteria [30]: the version of the measurement method, the context of the proposal, the functional domain (i.e., real time systems, management information systems), the input artifact (i.e., a requirements model, an analysis model, and a design model), the rules to apply the procedure, the instrument to apply the procedure, and the verification of the procedure.

The objectives of this paper are (1) to provide researchers with an overview of the current state of the functional size measurement procedures based on COSMIC and (2) to provide practitioners with information about the functional size measurement procedures that are available.

The rest of the paper is organized as follows: section 2 presents the existing proposals of measurement procedures based on COSMIC that allow the measurement of the functional size from conceptual models. Section 3 presents an overall analysis of the proposals. Finally, section 4 presents some conclusions, highlighting the features that must be considered by the functional size measurement procedures.

2 Functional Size Measurement Procedures

In this section, we present eleven proposals of functional size measurement procedures based on COSMIC. It is important to note that the proposals by Nagano et al. [36], Condori-Fernádez et al. [12], and Marín et al. [31] were correctly defined as measurement procedures. Even though the rest of the proposals presented in this survey were not originally defined as measurement procedures, they do correspond to measurement procedures according to the definition of the International Vocabulary of Basic and General Terms of Metrology [20], which defines a measurement procedure as: *a detailed description of a measurement according to one or more measurement principles and to a given measurement method.*

2.1 Proposal of Bévo et al. (1999)

Bévo et al. [7] perform a mapping between concepts of UML diagrams (use cases, scenarios, and classes) and concepts of COSMIC. A general description of this proposal is presented below:

- **Version of the Measurement Method.** Cosmic-FFP version 2.0 [2]
- **Context of the Proposal.** Unified Modelling Language (UML) version 1.0
- **Functional Domain.** Management information systems.
- **Input Artifact.** Diagrams of use cases, scenarios, and classes.
- **Rules to Apply the Procedure.** The boundary of the system to measure is included in the use case diagram. Each use case corresponds to a functional process. The data movements are represented in the scenarios, which are sequences of interactions that occur within a use case. Each class of the class diagram corresponds to a data group, and the attributes of those classes correspond to the data attributes. Each actor corresponds to a functional user. The triggering events and the layers are not represented with concepts of UML diagrams.
- **Instrument to Apply the Procedure.** A tool named *Metric Xpert* [6].
- **Verification of the Procedure.** The accuracy of the proposal was verified [6]. To perform this verification, five case studies were measured with the Metric Xpert tool. Then, the results were compared with the measures obtained by experts, obtaining differences that fluctuated between 11% and 33%.

2.2 Proposal of Jenner (2001)

Jenner [25] discusses the granularity aspect of the use cases in the proposal by Bévo et al. presented above. For this reason, the general characteristics of the Jenner proposal are very similar to the characteristics of the Bévo et al. proposal.

- **Version of the Measurement Method.** Cosmic-FFP version 2.0 [2]
- **Context of the Proposal.** UML version 1.0
- **Functional Domain.** Management information systems.
- **Input Artifact.** Diagrams of use cases, sequences, and classes.
- **Rules to Apply the Procedure.** Each functional process is represented by a sequence diagram because Jenner considers that sequence diagrams represent an adequate abstraction level of the use cases. The data movements are represented by the interaction messages of the sequence diagrams. This proposal also uses swimlanes to represent the layers of a system.
- **Instrument to Apply the Procedure.** This procedure has a tool [26].
- **Verification of the Procedure.** The proposal has been verified using case studies.

2.3 Proposal of Diab et al. (2001)

Diab et al. [15] present a set of formal rules that allow the measurement of the functional size of real time applications that are specified with Real-Time Object

Oriented Modelling (ROOM) [42]. The ROOM specifications are used by the Rational Rose Real Time (RRRT) tool for the design and specification of real time systems. The general characteristics of this proposal are the following:

- **Version of the Measurement Method.** Cosmic-FFP version 2.0 [2]
- **Context of the Proposal.** The design of an RRRT model might be observed through two different view points: structure and behavior. The structure of an RRRT model is based on three kinds of entities: actors, protocols, and data objects. An actor is an active object that has restricted visibility of and by other actors. A protocol represents a set of messages that can be exchanged among the actors. A data object is the basic unit of the system data. On the other hand, the dynamic part of an RRRT model is specified with a finite state machine for each actor. Each state machine can be defined with states, sub-states, actions, and transitions between the states.
- **Functional Domain.** Real time systems.
- **Input Artifact.** RRRT model (static and dynamic part).
- **Rules to Apply the Procedure.** The boundary of the system to be measured is represented by a set of actors. The layers correspond to a set of actors with the same level of abstraction, which must be selected by the practitioners using their human judgment. Each transition corresponds to a functional process. The data movements are represented by actions and messages. Actors and protocol classes correspond to data groups, and the attributes and variables of these classes correspond to the data attributes.
- **Instrument to Apply the Procedure.** A tool named *μcRose*[16]. This tool implements the measurement procedure that is updated to version 2.2 of Cosmic-FFP [21].
- **Verification of the Procedure.** The rules of the proposal have been verified by experts of COSMIC. In addition, this proposal has been applied to case studies, and the results have been compared with the measures obtained by experts. Finally, the tool assures the repeatability and consistency of the proposal.

2.4 Proposal of Poels (2002)

Poels [39] presents a mapping between concepts of COSMIC and the concepts of the business model and the services model of MERODE [14]. Later, this proposal was extended to allow the measurement of multilayer applications [41], specifying that the business model corresponds to a layer, and the services model corresponds to another layer. The general characteristics of this proposal are the following:

- **Version of the Measurement Method.** Cosmic-FFP version 2.1 [4]
- **Context of the Proposal.** The MERODE development method. This method is based on the MERODE conceptual model, which is comprised of a business model and a services model. The business model is composed by a class diagram, an object-event table, and a state transition diagram. The services model specifies the generation of events by the user and their transmission to the business model.
- **Functional Domain.** Management information systems.

- **Input Artifact.** MERODE model (business and services models).
- **Rules to Apply the Procedure.** Poels defines the rules separately for each model of MERODE. The users of the business model correspond to the services model. The boundary of the business model corresponds to the boundary between the business model and the users. Each functional process of the business model corresponds to a set of class methods over all of the enterprise objects, which are invoked by the occurrence of a type of business event. Each data movement corresponds to each class method that composes a functional process. In the business model, the exit data movements are not represented. The data groups correspond to the classes of the business model. On the other hand, the users of the services model correspond to the user interface model (this model is not specified in the MERODE model). The boundary of the services model corresponds to the boundary between the services model and the users. Each functional process of the services model corresponds to a non-persistent service object that is invoked by an input, output or control service request message or by a business event occurrence (for output object only). Again, each data movement corresponds to each class method that composes a functional process, and all the types of data movements are represented in the services model.
- **Instrument to Apply the Procedure.** Manual application of the procedure.
- **Verification of the Procedure.** This proposal has been validated theoretically [40].

2.5 Proposal of Nagano et al. (2003)

Nagano et al. [36] present a measurement procedure to measure the functional size of real time applications specified using xUML [34]. The general characteristics of this proposal are:

- **Version of the Measurement Method.** Cosmic-FFP version 2.0 [2]
- **Context of the Proposal.** The Shlaer-Mellor development method [43]. This method is an object-oriented method that uses xUML to specify systems.
- **Functional Domain.** Real time systems.
- **Input Artifact.** Classes, state transition, and collaboration diagrams.
- **Rules to Apply the Procedure.** The candidate data groups are attributes and relationships between objects of the class diagram. Also, the parameters of messages and control signals are candidate data groups. The triggering events are identified in the collaboration diagrams, which include the relationship between the external entity and the objects of the system. The functional processes correspond to a sequence of data movements. Finally, the data movements correspond to the actions that an object performs to move it from one state to the next state according to the collaboration diagram.
- **Instrument to Apply the Procedure.** Manual application of the procedure.
- **Verification of the Procedure.** This proposal has been applied to the Rice Cooker case study [13], and the results were compared with the results obtained by experts, obtaining a difference of 53%.

2.6 Proposal of Azzouz et al. (2004)

Azzouz et al. [5] present a tool that automates the measurement of the functional size of applications developed with the Rational Unified Process (RUP) [28]. The general characteristics of this proposal are:

- **Version of the Measurement Method.** Cosmic-FFP version 2.2 [21]
- **Context of the Proposal.** Rational Unified Process. This method uses UML to specify the systems.
- **Functional Domain.** Management information systems.
- **Input Artifact.** Use case diagrams, scenarios, and detailed scenarios.
- **Rules to Apply the Procedure.** Azzouz et al. base their proposal on the rules described by Bévo [7] and Jenner [25]. However, Azzouz considers that the layer cannot be represented in the UML diagrams. Therefore, the user of the tool must manually identify the layers of the system. Also, this proposal adds a stereotype to identify the triggering events in the use case diagrams. The measurement is performed in three phases of RUP: in the business modeling and requirement analysis phase, the artifact used is the use case diagram; in the analysis phase, the artifact used is the scenario; and in the analysis and design phase, the artifact used is the detailed scenario.
- **Instrument to Apply the Procedure.** A tool integrated in the Rational Rose tool.
- **Verification of the Procedure.** The tool was verified using the Rice Cooker case study [13].

2.7 Proposal of Condori-Fernández et al. (2004)

Condori-Fernández et al. [12] present a measurement procedure to estimate the functional size of object-oriented systems from the requirements specifications that are defined using the OO-Method approach [37]. The general characteristics of this proposal are:

- **Version of the Measurement Method.** Cosmic-FFP version 2.2 [21]
- **Context of the Proposal.** The development method OO-Method. This method is based on a formal language. It is an object-oriented method that allows the automatic generation of final applications by means of model transformations [38]. The software production process in OO-Method is represented by three models: the requirements model, the conceptual model, and the execution model. The requirement model specifies the system requirements using a set of techniques such as the mission statement, the functions refinement tree, and the use case diagram. To establish the traceability between the requirements model and the conceptual model, the requirements model uses sequence diagrams. The conceptual model captures the static and dynamic properties of the functional requirements of the system (object, dynamic, and functional models). The conceptual model also allows the specification of the user interfaces in an abstract way through the presentation model. The execution model allows the transition from the problem space to the solution space. The software product can be generated in a systematic and automatic way for different platforms.

- **Functional Domain.** Management information systems.
- **Input Artifact.** OO-Method requirements model (functions refinement tree, use case diagrams, and sequence diagrams).
- **Rules to Apply the Procedure.** The boundary of the system to be measured corresponds to the border between the set of use cases and the actors of the use case diagram. Each functional process corresponds to each elementary function of the functions refinement tree (primary use case). Also, each secondary use case corresponds to a functional process. The data groups are identified in the sequence diagram. Each different actor, control class or entity class of the sequence diagram corresponds to a data group. The data movements correspond to the messages of the sequence diagrams. In this proposal a single layer is identified because there is not a functional partition at the requirements level. The triggering events are not represented.
- **Instrument to Apply the Procedure.** Manual application of the procedure.
- **Verification of the Procedure.** This proposal has been rigorously verified in several ways: according to measurement theory [9]; in conformity with COSMIC [9]; using the formal framework DISTANCE [9]; performing empirical studies of its repeatability and reproducibility [11], and evaluating its adoption in practice [10].

2.8 Proposal of Habela et al. (2005)

Habela et al. [19] present an extension of the use case model that allows the measurement of the functional size using COSMIC. The general characteristics of this proposal are:

- **Version of the Measurement Method.** Cosmic-FFP version 2.2 [21]
- **Context of the Proposal.** UML version 1.5
- **Functional Domain.** Management information systems.
- **Input Artifact.** Use case diagrams, and detailed use cases using a template that includes references to business rules, pre-conditions, post-conditions, and a description in steps of the main and alternatives scenarios.
- **Rules to Apply the Procedure.** Each use case corresponds to one or more functional processes. The data movements are identified in each step described in the scenarios. Each step specifies the movement of a set of data attributes. The uses, extends, and generalizations between use cases are taken into account to avoid redundancies in the measurement.
- **Instrument to apply the Procedure.** Manual application of the procedure.
- **Verification of the Procedure.** We did not find studies of validation, verification, or application of this proposal.

2.9 Proposal of Grau et al. (2007)

Grau et al. [18] present a set of mapping rules to measure the functional size of i* models generated by means of reengineering of systems using PR*i*M [17]. The general characteristics of the Grau et al. proposal are the following:

- **Version of the Measurement Method.** Cosmic-FFP version 2.2 [21]
- **Context of the Proposal.** The PR*i*M method, which is a process reengineering i* method that addresses the specification, analysis and design of information systems from a reengineering point of view. In PR*i*M, the i* model is comprised of two models: an operational i* model (that contains the functionality of the system), and an intentional i* model (that contains the non-functional requirements). To generate the operational i* model, scenario-based templates named *Detailed Interaction Scripts* are used. These templates describe the information of each activity of the current process by means of pre-conditions, post-conditions, triggering events, and a list of actions undertaken in the activity.
- **Functional Domain.** Management information systems.
- **Input Artifact.** Detailed interaction scripts, and an operational i* model.
- **Rules to Apply the Procedure.** The boundary of the system to be measured corresponds to the actor of the operational i* model that represents the different pieces of the system. The users are actors of the operational i* model that represent one or more human roles. The data movements are identified in the operational i* model and correspond to any dependency where the *dependum* is a resource. Each functional process corresponds to an activity of the detailed interaction scripts. The triggering events are part of the conditions associated to the activity. Finally, the data groups correspond to the resources of the detailed interaction script.
- **Instrument to Apply the Procedure.** A tool named *J-PRiM*.
- **Verification of the Procedure.** This proposal has been applied to the C-Registration case study [27], and the results have been compared with the results obtained by experts, obtaining a difference of 53%.

2.10 Proposal of Levesque et al. (2008)

Levesque et al. [29] apply COSMIC to measure the functional size of systems from use case diagrams and sequence diagrams. This proposal classifies the functional processes in two groups: data movement types and data manipulation types. The general characteristics of the Levesque et al. proposal are the following:

- **Version of the Measurement Method.** Cosmic-FFP version 2.1 [3]
- **Context of the Proposal.** UML version 1.4, and UML version 2.0
- **Functional Domain.** Management information systems.
- **Input Artifact.** Use cases and sequence diagrams.
- **Rules to Apply the Procedure.** For the functional processes corresponding to the data movement type, each use case is a functional process. The actors of the use case are the users. The entities of the sequence diagram are the data groups. The data movements correspond to the messages among the entities of the sequence diagram. On the other hand, the data manipulations correspond to the conditions associated to the error messages of the sequence diagrams. Finally, this proposal obtains the functional size aggregating the messages between the actors and objects of the sequence diagrams.
- **Instrument to Apply the Procedure.** Manual application of the procedure.

- **Verification of the Procedure.** This proposal has been applied to the Rice Cooker case study [13], and the results have been compared with the results obtained by experts, obtaining a difference of 8%.

2.11 Proposal of Marín et al. (2008)

Marín et al. [31] present a measurement procedure to measure the functional size of object-oriented systems generated in MDA environments from their conceptual models. This proposal uses the OO-Method development method [38] as the reference MDA approach. The general characteristics of the Marín et al proposal are the following:

- **Version of the Measurement Method.** Cosmic-FFP version 3.0 [4]
- **Context of the Proposal.** The development method OO-Method version 3.8. This method is composed by three models: the requirements model, the conceptual model and the execution model. The last two models of the OO-Method approach has been implemented in a tool named *Olivanova* [8]. This tool allows the specification of systems with a graphical notation in a conceptual model and allows the automatic generation of software products from this conceptual model. The OO-Method conceptual model is comprised of four models: the object model, the functional model, the dynamic model, and the presentation model. The specification of the systems with these four models allows the automatic generation of fully working applications. The OO-Method applications are generated with a three tier architecture: presentation, logic, and database. Each tier of the architecture is associated with the other tiers in a superior/subordinate hierarchical dependency. Therefore, the presentation tier can use the services of the logic tier because the logic tier is beneath the presentation tier in the hierarchy. In the same way, the logic tier can use the services of the database tier because the database tier is beneath the logic tier in the hierarchy. In addition, the OO-Method applications have at least one software component in each tier of the architecture: the client component, the server component, and the database component. The client component has the graphical user interface of the applications. The server component has the business logic of the application. And finally, the database component has the persistence of the application.
- **Functional Domain.** Management information systems.
- **Input Artifact.** OO-Method conceptual model (object, functional, dynamic and presentation).
- **Rules to Apply the Procedure.** This proposal is structured in the three phases of the COSMIC method: the strategy phase, the mapping phase and the measuring phase. With respect to the strategy phase, the scope of the measurement can be determined by the functional processes, the layers, or the whole application. The layers correspond to the hierarchical tiers of the OO-Method applications: the presentation tier, the logic tier, and the database tier. The pieces of software correspond to the software components: the client component, the server component, and the database component. The users are the human users, the client component, and the server

component of the applications. The users are separated from the pieces of software by a boundary. With respect to the mapping phase, the functional processes are groups of functionality that can be directly accessed by the users. These groups of functionality correspond to the interaction units specified in the menu of the presentation model. The data groups correspond to the classes of the object model that participate in the functional processes. The data attributes correspond to the attributes of the classes identified as data groups. With respect to the measuring phase, the data movements correspond to the movements of data groups between the users and the functional processes. This proposal has 69 rules to identify the data movements that can occur in the OO-Method applications. Finally, this proposal has a set of rules to obtain the functional size of each functional process of the application, of each piece of software of the application, and of the whole application.

- **Instrument to Apply the Procedure.** This procedure has a tool [32].
- **Verification of the Procedure.** This proposal has been verified respect its conformity with COSMIC. Also, the tool has been verified using OO-Method case studies, and the results have been compared with the measures obtained by experts. Finally, the tool assures the repeatability and the consistency of the proposal.

3 General Analysis

In this section, we present an overall analysis of the criteria used in the survey presented above.

With respect to the version of the COSMIC measurement method, we observed that four proposals (Bévo, Jenner, Diab, and Nagano) use the 2.0 version, two proposals (Poels and Levesque) use the 2.1 version, four proposals (Azzouz, Condori-Fernández, Habela, and Grau) use the 2.2 version, and one proposal (Marín) uses the 3.0 version. It is important to note that the proposal oby Nagano (which was defined in 2003) uses the 2.0 version in spite of the fact that newer versions of COSMIC already exited in 2003. It is also important to note that the proposal by Levesque (which was defined in 2008) uses the 2.1 version in spite of the fact that the version 3.0 of COSMIC already existed in 2008. Our opinion is that newer versions of COSMIC provide improvements and clarifications that help to better understand the measurement method and to obtain accurate measures. Therefore, we consider that the use of the last version of the method is very important for the correct development of measurement procedures.

With regard to the context of the procedure, five proposals (Bévo, Jenner, Azzouz, Habela, and Levesque) measure UML models, one proposal (Diab) measures RRRT models, one proposal (Poels) measures MERODE models, one proposal (Nagano) measures xUML specifications, two proposals (Condori-Fernández and Marín) measure OO-Method models, and one proposal (Grau) measures i* models. The UML, MERODE, and i* models do not have enough expressivity to specify all the functional requirements of the applications (for instance, these models do not allow the specification of the values assigned to the attributes of the classes, the interaction units, etc.). The same situation occurs with the OO-Method requirement models. Therefore, the proposals based on these models only estimate the functional size of the applications. On the other hand, the proposals based on the RRRT model, the xUML specification, and the OO-Method conceptual model have enough semantic formalization to specify all the functional requirements, allowing the measurement of the functional size of the applications.

With respect to the functional domain, we observed that only two proposals (Diab, Nagano) have been developed for the domain of real time systems. The remaining nine proposals have been developed for the domain of management information systems. We did not find any measurement procedure proposal for other domains (such as algorithmic systems, geographical systems, ubiquitous systems, etc.), in spite of the fact that the COSMIC measurement method can be applied to any software system domain.

With regard to the input artifact, all the proposals use more than one input artifact. Seven proposals (Bévo, Jenner, Azzouz, Condori-Fernández, Habela, Grau, and Levesque) use input artifacts obtained in the requirements phase, three proposals (Diab, Poels, Nagano) use input artifacts obtained in the analysis phase, and only one proposal (Marín) uses input artifacts obtained in the analysis and design phase.

With respect to the rules to apply the procedure, only two proposals (Condori-Fernández and Marín) perform the design of the measurement procedure, defining the objectives of the procedure, the characterization of the concept to be measured, the mapping with the concepts of COSMIC, and the measurement rules. The remaining nine proposals only define some mappings between the concepts of COSMIC and the concepts of the conceptual models to be measured. The design of a measurement procedure is a key stage in the development of a measurement procedure (correctly abstracting the elements that will be measured), since, otherwise, the procedure may not measure what should be measured according to the specifications of the base measurement method selected. It is also important to keep in mind the direct influence that the design of a measurement procedure has on the application and possible automations of the procedure.

With regard to the instrument to apply the procedure, six proposals (Bévo, Jenner, Diab, Azzouz, Grau, and Marín) have been automated, and five proposals (Poels, Nagano, Condori-Fernández, Habela, and Levesque) must be applied manually. The manual measurement of functional size is generally very time-consuming and could have many precision errors. Therefore, it is very important to automate the measurement procedures to obtain a solution that can be efficiently applied in academic and industrial environments. In addition, a tool that automates the measurement procedures reduces the measurement costs and measurement training, and ensures perfect repeatability of the measures.

Finally, with respect to the verification of the procedure, we observed that only one proposal (Habela) has not been verified in some way. The remaining proposals have been verified using different techniques: using case studies, performing theoretical validations, performing conformity validations, using empirical studies, etc. Thus, it is important to keep in mind that a high quality design of a functional size measurement procedure is not enough to assure the quality of the measures obtained by this procedure. To ensure the quality of the results obtained, it is also essential to verify the developed procedure.

4 Conclusions

This paper provides an extensive summary of the existing proposals of functional size measurement procedures that are based on the COSMIC measurement method and that use the conceptual models as input artifacts to perform the measurement. The main contribution of this work is the presented survey, which provides researchers with an updated overview of the current state of the functional size measurement procedures that are based on COSMIC. This survey also provides practitioners with valuable information about the functional size measurement procedures that are available.

It is important to remark that the measurement procedures presented in this paper have been developed to apply the COSMIC measurement method to the conceptual models in order to obtain the functional size of final applications in early stages of the software development process. Therefore, some of the important lessons taken from this work are: 1) the measurement procedure must be based on the last version of the measurement method; 2) the input artifact used must have enough semantic formalization to allow the specification of all the functional requirements; 3) the design of the measurement procedure must be carried out, clearly defining rules to specify the strategy of the measurement, rules to perform the mapping between the concepts of COSMIC and the concepts of the conceptual models, and rules to identify the data movements and perform the measurement; 4) the automation of the procedure must be carried out to reduce the cost of performing the measurement and to increase the efficiency of the measurement process; and finally, 5) the verification of the procedure must be carried out to assure the quality of the results obtained.

References

1. Albrecht, A.: Measuring Application Development Productivity. In: IBM Applications Development Symposium, pp. 83–92 (1979)
2. Abran, A., Desharnais, J.M., Oligny, S., St-Pierre, D., Symons, C.: COSMIC-FFP Measurement Manual, version 2.0. Software Engineering Management Research Laboratory, Université du Québec à Montréal - UQAM, Canada (1999)
3. Abran, A., Desharnais, J.M., Oligny, S., St-Pierre, D., Symons, C.: COSMIC-FFP Measurement Manual, Version 2.1. The Common Software Measurement International Consortium (2001)
4. Abran, A., Desharnais, J.M., Lesterhuis, A., Londeix, B., Meli, R., Morris, P., Oligny, S., O'Neil, M., Rollo, T., Rule, G., Santillo, L., Symons, C., Toivonen, H.: The COSMIC Functional Size Measurement Method, version 3.0. GELOG web site (2007), http://www.gelog.etsmtl.ca
5. Azzouz, S., Abran, A.: A proposed measurement role in the Rational Unified Process (RUP) and its implementation with ISO 19761: COSMIC FFP. In: Software Measurement European Forum 2004, Rome (2004)
6. Bevo, V.: Analyse et Formalisation Ontologique des Procédures de Mesure Associées aux Méthodes de Mesure de la Taille Fonctionnelle des Logiciels: de Nouvelles Perspectives Pour la Mesure. Doctoral thesis, Université du Québec à Montréal - UQAM, Montréal (2005)

7. Bévo, V., Lévesque, G., Abran, A.: Application de la méthode FFP à partir d'une spécification selon la notation UML: compte rendu des premiers essais d'application et questions. In: 9th International Workshop Software Measurement, Lac Supérieur, Canada, pp. 230–242 (1999)
8. CARE Technologies, `http://www.care-t.com`
9. Condori-Fernández, N.: Un procedimiento de medición de tamaño funcional a partir de especificaciones de requisitos. Doctoral thesis, Universidad Politécnica de Valencia, Valencia (2007)
10. Condori-Fernández, N., Pastor, O.: An Empirical Study on the Likelihood of Adoption in Practice of a Size Measurement Procedure for Requirements Specification. In: 6th International Conference on Quality Software – QSIC, Beijing, pp. 133–140 (2006)
11. Condori-Fernández, N., Pastor, O.: Evaluating the Productivity and Reproducibility of a Measurement Procedure. In: ER Workshops, pp. 352–361 (2006)
12. Condori-Fernández, N., Abrahão, S., Pastor, O.: On the Estimation of Software Functional Size from Requirements Specifications. Journal of Computer Science and Technology 22(3), 358–370 (2007)
13. COSMIC Group: Rice Cooker – Cosmic Group Case Study. École de technologie supérieure, Université du Québec à Montréal - UQAM, Montréal (2003)
14. Dedene, G., Snoeck, M.: M.E.R.O.DE.: A Model-driven Entity-Relationship Object-oriented Development Method. ACM SIGSOFT Software Engineering Notes 19(3), 51–61 (1994)
15. Diab, H., Frappier, M., St-Denis, R.: Formalizing COSMIC-FFP Using ROOM. In: ACS/IEEE International Conference on Computer Systems and Applications, Beirut (2001)
16. Diab, H., Koukane, F., Frappier, M., St-Denis, R.: μcROSE: Automated Measurement of COSMIC-FFP for Rational Rose Real Time. Information and Software Technology 47(3), 151–166 (2005)
17. Grau, G., Franch, X.: ReeF: Defining a Customizable Reengineering Framework. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 485–500. Springer, Heidelberg (2007)
18. Grau, G., Franch, X.: Using the PRiM method to Evaluate Requirements Model with COSMIC-FFP. In: Proceedings of the IWSM-MENSURA 2007, Mallorca, pp. 110–120 (2007)
19. Habela, P., Glowacki, E., Serafinski, T., Subieta, K.: Adapting Use Case Model for COSMIC-FFP Based Measurement. In: 15th International Workshop on Software Measurement – IWSM 2005, Montréal, pp. 195–207 (2005)
20. ISO: International vocabulary of basic and general terms in metrology – VIM (2004)
21. ISO/IEC: ISO/IEC 19761, Software Engineering – CFF – A Functional Size Measurement Method (2003)
22. ISO/IEC: ISO/IEC 20926, Software Engineering – IFPUG 4.1 Unadjusted Functional Size Measurement Method – Counting Practices Manual (2003)
23. ISO/IEC: ISO/IEC 20968, Software Engineering – Mk II Function Point Analysis – Counting Practices Manual (2002)
24. ISO/IEC: ISO/IEC 24570, Software Engineering – NESMA Functional Size Measurement Method version 2.1 – Definitions and Counting Guidelines for the application of Function Point Analysis (2005)
25. Jenner, M.S.: COSMIC-FFP and UML: Estimation of the Size of a System Specified in UML – Problems of Granularity. In: 4th European Conference on Software Measurement and ICT Control, Heidelberg, pp. 173–184 (2001)

26. Jenner, M.S.: Automation of Counting of Functional Size Using COSMIC-FFP in UML. In: 12th International Workshop Software Measurement, pp. 43–51 (2002)
27. Khelifi, A., Abran, A., Symons, C., Desharnais, J.M., Machado, F., Jayakumar, J., Leterthuis, A.: The C-Registration System Case Study with ISO 19761 (2003)
28. Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley, Reading (2000)
29. Levesque, G., Bevo, V., Cao, D.T.: Estimating software size with UML models. In: Proceedings of the 2008 C^3S^2E Conference, Montreal, pp. 81–87 (2008)
30. Lother, M., Dumke, R.: Point Metrics-Comparison and Analysis. In: Current Trends in Software Measurement, Aachen, pp. 228–267 (2001)
31. Marín, B., Condori-Fernández, N., Pastor, O., Abran, A.: Measuring the Functional Size of Conceptual Models in a MDA Environment. In: 20th International Conference on Advanced Information Systems Engineering Forum, Montpellier, pp. 33–36 (2008)
32. Marín, B., Giachetti, G., Pastor, O.: Automating the Measurement of Functional Size of Conceptual Models in a MDA Environment. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 215–229. Springer, Heidelberg (2008)
33. Meli, R., Abran, A., Ho Vinh, T., Oligny, S.: On the Applicability of COSMIC-FFP for Measuring Software Throughout its Life Cycle. In: 11th European Software Control and Metrics Conference, Munich (2000)
34. Mellor, S., Balcer, J.: Executable UML: A Foundation for Model-Driven Architecture. Addison Wesley, Reading (2002)
35. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1 (2003)
36. Nagano, S., Ajisaka, T.: Functional metrics using COSMIC-FFP for object-oriented real-time systems. In: 13th International Workshop on Software Measurement, Montreal (2003)
37. Pastor, O., Gómez, J., Insfrán, E., Pelechano, V.: The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming. Information Systems 26, 507–534 (2001)
38. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)
39. Poels, G.: A Functional Size Measurement Method for Event-Based Object-Oriented Enterprise Models. In: 4th International Conference on Enterprise Information Systems – ICEIS, Ciudad Real, pp. 667–675 (2002)
40. Poels, G.: Definition and Validation of a COSMIC-FFP Functional Size Measure for Object-Oriented Systems. In: 7th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, Darmstadt (2003)
41. Poels, G.: Functional Size Measurement of Multi-Layer Object-Oriented Conceptual Models. In: 9th International Object-Oriented Information Systems Conference, Geneva, pp. 334–345 (2003)
42. Selic, B., Gullekson, G., Ward, P.T.: Real-time Object Oriented Modelling. Wiley, Chichester (1994)
43. Shlaer, S., Mellor, S.: Object Lifecycles: Modelling the World in States. Yourdon Press, Prentice-Hall (1992)