An Automated Software Size Measurement Tool based on Generation Model using COSMIC Function Size Measurement

Thandar Zaw, Swe Zin Hlaing, Myint Myint Lwin, Koichiro Ochimizu
University of Information Technology, Yangon, Myanmar
thandarzaw@uit.edu.mm, swezin@uit.edu.mm, myintmyintlwin@uit.edu.mm, ochimizu@jaist.ac.jp

Abstract

At present, most of the software industries need the well-defined models and standard measurement methods to measure the size of software. They are useful in the estimation of effort and time that are extensively used at the early stage of software development. COSMIC Functional Size Measurement (FSM) is one of the recognized methods to estimate the size of software project. The several research initiatives have focused on COSMIC FSM procedure with specific design diagram notation by manually and automatically. However, manual measurement of the various diagrams is timewasting and difficult to measure the accurate size of the software. Therefore, the automated measurement of function size which includes the generation model based on three different diagram notations such as UML, SysML and Petri net and general mapping rules between COSMIC FSM and generation model to measure the size of software is proposed in this paper. The manual measurement of cooker system by using the generation model and COSMIC concept is successfully done in our previous paper. This proposed automated tool can handle various types of diagramming and get the size of software correctly.

Key Words- Meta model, Unified Modeling Language (UML) and Systems Modeling Language (SysML) sequence diagram, COSMIC Functional Size Measurement (COSMIC FSM)

1. Introduction

The software size measurement plays the critical factor in developing a software project to estimate costs, efforts and other resources. Size estimation method is very important to get the accurate size. Size methods include source lines-of-code and function points that are valuable size estimation techniques. Traditionally, SLOC is the major software size estimation method by counting the number of source line of code. But, it is difficult to estimate the software size at the early stages of software development and is language-specific.

An alternative method, a Functional Size Measurement (FSM) method measures the amount of functionality to be delivered. Function Point Analysis (FPA) is a method of Functional Size Measurement to estimate software size. The size of software can be estimated at the beginning of a project by using the method. It measures the functional user requirements. It gives the necessary input to estimate the effort in design phase for the industrial software development.

Five function size measurement methods have been recognized ISO standards: IFPUG FPA, MKII, NESMA, FISMA [13], and COSMIC FSM [12]. Among them, COSMIC like Functional Point Analysis (FPA) method that is an internationally standardized functional size measuring method for most application domains, including business application software, real-time software. Although FPA can be measured for business application domain, COSMIC can be measured for both application domains. It can be estimated from statements of requirements early in a project. Most of the researchers proposed the measurement of the software size with specific design diagram notation such as UML, Simulink, Business Process Model and Notation (BPMN), SCADE and so on to collect the software requirements and to measure the functional size of the software. These results may vary based on the different design diagrams. Whenever the measurers use the different diagram notations, the generation model supports the same concepts of the requirements for different measures. Further the manual measurement of the large size of software will take a lot of time and require the specialized experts. The specific automated measurement tool has developed to measure the software size but the general measurement automated tool has not provided yet. To overcome the specific automated measurement tool, it is essential that the automated software measurement tool which can generate the generation model. This generation model offers the different diagram notations to measure the size of software system. The proposed approaches handle different diagram notations by defining mapping rules to resolve the specific diagram notation and to acquire the precise function size of software without calculating the manual measurement. It can be used for estimating development effort and productivity of the project. The organization of the paper is: Section II presents the related work. Section III presents the proposed system of FSM procedure based on generation model. Section IV provides the case study of cooker system. Section V presents the conclusion.

2. Related Work

According to measure the functional size of software, COSMIC is one the standard functional size measurement methods. COSMIC can be measured both business application domain and real time application domain.

Currently, COSMIC-based FSM procedures focus on not only the use of non-real time embedded software [1,7,6,8,9,10,11] but also real-time embedded software [3,4] which is discussed below. The authors presented the functional size of software can be measured by using the COSMIC FSM in any real-time application domain with a simple example of alarm system to understand clearly for software engineers in [2]. The authors proposed the COSMIC method with UML modeling to measure the functional size of software with the case study of cooker system for improving the applicability of the COSMIC method in industry [6]. N.A.S. Abdullah [8] also present the COSMIC measurement procedure by using the UML representations to capture the information for estimating the size of Angry Birds mobile application. The sequence diagram is used for both functional and structural size measurement to estimate the software size.

S.Karim [7] propose the automated measurement of software size for functional and structural from XML representation of the UML sequence diagram which is mapping into the COSMIC FSM for calculating the size of software. A. Sellami [1] propose the manual measurement of the software size based on the UML sequence diagram and COSMIC FSM both structurally and functionality with the three small case studies to improve the effort estimation model. The authors [3] propose the automated measurement tool which is based on COSMIC FSM procedure and Simulink modeling tool and for real-time embedded software. H. Soubra et al. propose a detail description of FSM procedure for both manual and automated measurement of real-time embedded aerospace system software designed in Safety-critical Application Development Environment (SCADE) in [4]. Finally, the results of the automated tool and manual measurement are compared. There are no difference measurement results by comparing the detailed level. In [5], the authors contribute the proposed UML profile and validated the pattern diagrams using the constraints with an example of a Real-Time (RT) design pattern.

T. Zaw [9] propose the automated function size measurement tool which comprise the light weight generation model from UML meta model and COSMIC FSM. This tool supports the generation model for UML met model only. The above papers have focused on the specific notation not only manually but also automatically. T. Zaw [10] presents the automated functional size measurement tool that applied the XML representation of three different diagram notations and mapping rules between these diagram notations and COSMIC to estimate the functional size of software. This tool analyzed and calculate XML structure of sequence diagram of UML, SysML and Petri net diagram individually to search the tag structures. This tool cannot handle the concepts difficult diagram notation and convert these different diagram notations to a common one. Some of the large system may be complex design notations. The different designers can be difficult to understand these complex diagrams. Thus, we proposed [11] the manual measurement of COSMIC FSM in generation model which is from three different diagram notations to measure the functional size of software. The generation model helps researchers to know the complicated diagrams notations. However, manual measurement is time-wasting and occur the chance of human mistake. The automated measurement tool which can handle the concepts of difficult diagram clearly and convert the common design diagram are needed. As a result, this paper proposes the automated measurement tool which applied generation model from three different design diagram notations and converted the common design diagram by using profiling mechanism. This tool can make the mapping between the generation model and COSMIC FSM which can be used in various types of software. A famous case study of the cooker system [6] is used to shows the usefulness of the proposed approach in this research. Many industries use this measurement result to increase effort.

3. Proposed System

The manual measurement of the generation model and COSMIC concept has been defined in our previous paper [11]. It is essential part for the automated measurement tool. Although the manual measurement is proposed to measure the software size, it can get a long period and needs the expert to measure the software size. So, the automated measurement tool which is based on the generation model and COSMIC by combining the three processing phases of COSMIC measurement as one in the manual measurement is proposed in this paper. Firstly, the generation model is defined by using the UML and SysML sequence meta model, Petri net meta model. After that, the sequence diagram of UML and SysML and Petri net are used as an input to calculate the function size of software and use the generation model. Then, the mapping rules between the generation model and COSMIC FSM define. Finally, the measurement result is calculated by using COSMIC method. The design of automated measurement tool is shown in Figure 1.

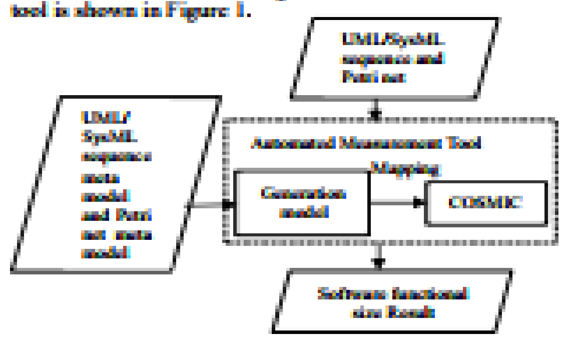

Figure 1. Automated Function Size Measurement Tool

4. Case Study

In this section, in order to get the automation of the FSM procedure, the specification of the Cooker system [6] is used to describe the calculating of COSMIC. When the requirement of Cooker system is defined, the design diagram notations can be established. The functional requirements of this system are as follows:

If the user wants to cook rice, he will choose one of three modes: fast, normal and warm. The heater is controlled depending on the mode and elapsed time. The time will issue that the target temperature will be fixed for each 30seconds and the heater control heat on/off when the target temperature is compared with the actual temperature

Figure. 2. The use case diagram of enhanced rice cooker

for every 5 seconds. The enhanced cooker system has four functional processes. The use case diagram of improved rice cooker has four functional processes as shown in Figure 2.

4.1. COSMIC FSM in Generation Model

Firstly, the generation model is defined from the partial sequence diagram of UML and SysML meta model, Petri net meta model and COSMIC concept by using profiling mechanism. This model uses life line, message, place and transition meta class from existing UML, SysML and Petri net meta model to be extended by using stereotypes and then applies to applicable model elements in the user model. The new extensions are proposed by using each stereotype as follow:

- <<Object>> stereotypes. We use it to extend Lifeline meta-class to show a lifeline. The design creator can add the definite elements consistent with the desired application model.
- <<Entry>>, <<Exit>>, <<Read>> and <<Write>> stereotypes. These stereotypes are used to extend Message or Transition meta-classes to show a message. The design creator can also add the definite elements consistent with the desired application model.
- The <<Entry>> stereotype specifies the entry data movements of the transaction with COSMIC concept.
- The <<Exit>> stereotype specifies the exit data movements of the transaction with COSMIC concept.
- The <<Read>> stereotype specifies the read data movements of the transaction with COSMIC concept.
- The <<Write>> stereotype specifies the write data movements of the transaction with COSMIC concept.
- The <<alt>> and <<opt>> stereotypes are interaction operators.
- The <<Alt>> stereotype (alternative) is used to illustrate the maximum selected operands that are one.
- The <<Opt>> stereotype (optional) is used to illustrate one of each operand occurs or nothing occurs that are selected.

Figure 3 illustrates the structure of the proposed profile for generation model.

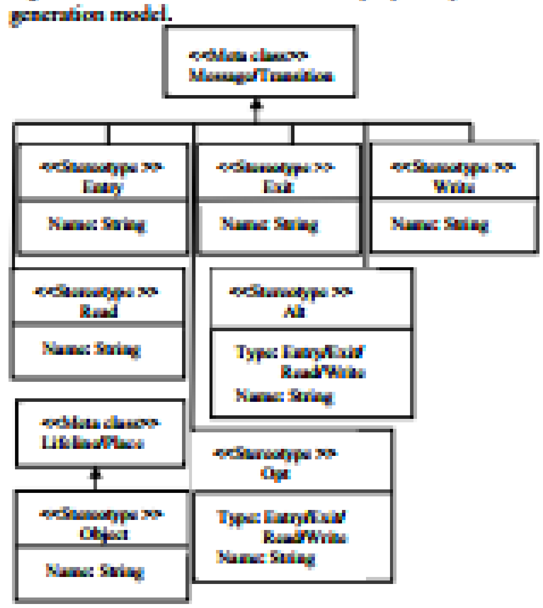

Figure 3. Generation model from partial UML, SysML and Petri net meta model

4.2. Mapping Rules

When the generation model has been constructed, this model is applied to COSMIC concepts by using the mapping rules. The mapping rules are defined between generation model from three different diagram notations such as the sequence diagram of UML and SysML, and Petri net and COSMIC. Table 1 summarizes the mapping rules from the generation model to COSMIC. The mapping rules are defined as follows:

•Rule 1: Identify the application boundary.

The bounding line of application is constituted between the application system and outside elements. The application border in COSMIC corresponds to use case diagram.

•Rule 2: Identify the functional users.

COSMIC defines a user, a transmitter and /or receiver which send and /or receive data in Functional User Requirement of software. The concepts of COSMIC correspond the users or an objects in modeling notations such as UML and SysML sequence diagram and Petri net.

•Rule 3: Identify the functional process.

It requires the data from functional user that corresponds to interaction between objects that operate with one another.

•Rule 4: Identify the data groups.

The data groups identify trigger event that carries data group between objects. A COSMIC data group corresponds to the data group that may be represented the flows of information between objects.

•Rule 5: Identify the four data movements.

Each data movements correspond to an interaction messages in UML and SysML sequence diagram and transition in Petri net.

•Rule 5.1: Identify the Entry data movement.

It moves the message from functional user to boundary. It identifies from Rule 2 to Rule 3.

•Rule 5.2: Identify the Exit data movement.

It moves the message from boundary to functional user. It identifies from Rule 3 to Rule 2.

•Rule 5.3: Identify the Read data movement. It moves the single data group from persistent storage to functional process.

•Rule 5.4: Identify the Write data movement.

It moves the single data group from functional process to persistent storage.

•Rule 6: Applying the COSMIC measurement function.

By counting the four data movements that are distinguished in every functional process.

•Rule 7: Aggregation the functional size measurements.

By adding all of distinguished data movement of every functional process of the entire system into a single function size value to obtain the functional size of system.

4.3. Automated Measurement Tool

The automated measurement tool can measure the functional size of software for any kind of system. In this paper, we measure the automated measurement of a case study of cooker system. At first, the sequence diagrams and petri net diagrams of cooker system are an important input to identify the functional processes and data movements. These diagrams are as shown in Figure 4 to 7. In the measurement tool, the XML representation of the desired file from the browse button is selected by the users. When the calculate button is clicked, the measurement tool can generate the generation method and then these generation model has been transformed to COSMIC by using mapping rules. Then, the generation model and measurement results are shown in this tool.

Table 1. Mapping rules of UML/SysML, Petri net and COSMIC

Rule No	COSMIC element	UML/SysML Diagram	Petri net
1	Boundary	Use Case	Use Case
2	Functional User	Object in Sequence Diagram	Object or initiator
3	Functional Process	Interaction between objects	Interaction between objects
4	Data Group	Flows of information between objects	Flows of information between objects
5.1	Entry Data Movement	Sequence message from Functional User to Functional Process	Transition messages from Functional User to Functional Process
5.2	Exit Data Movement	Sequence message from Functional Process to Functional User	Transition message from Functional Process to Functional User
5.3 5.4	Read & Write Data Movement	Sequence message move single data group from persistent storage to a functional process	Transition message move single data group from persistent storage to a functional process
6	Applying the COSMIC measurement function.	Applying the COSMIC measurement function.	Applying the COSMIC measurement function.
7	Aggregation the functional size measurements	Aggregation the functional size measurements	Aggregation the functional size measurements

In Figure 4 and 5, the cooker software receives start () signal from the StartButton. When instantiating the applicable model to apply the generation model, the start () signal is defined with the <<Entry>> stereotype as shown in Figure 8. It identified 1 data movement. The detailed description of generation model is shown in Figure 3. The DisableItem() signal send from the cooker software to the ModeselectionButton to deactivate the mode. This signal is also defined with the <<Exit>> stereotype in Figure 8. The ResetElapsed() is defined with the <<Write>> stereotypes to reset the time at one second intervals. The CookLampOn() is defined with <<Exit>> stereotype to lamp on the lamp. Enable () is defined with the <<Exit>> stereotypes to activate the temperature sensor. HeaterOn() are defined with the <<Exit>> stereotypes to heat on the heater. If the cooker software sends or receives a signal, this signal is defined with their relatived stereotype as shown in Figure 3 and Figure 8. The total number of start cooking process is 7 CFP (COSMIC Function Point). The other functional processes are also identified like above procedure. This tool supports the total number of function point identified by calculating the total number of four data movements of the system.

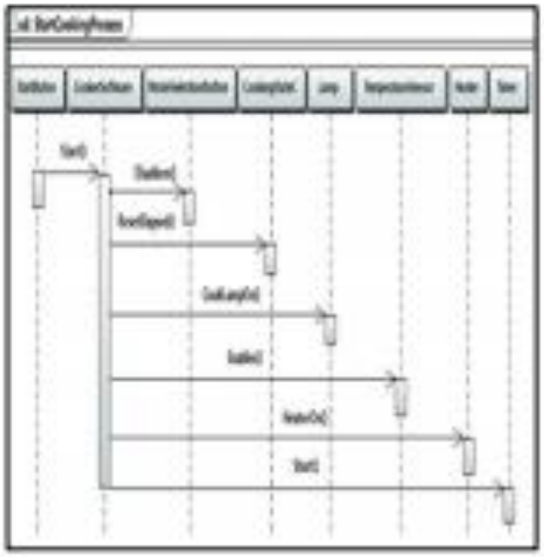

Figure 4. Start Cooking Process of UML and SysML Sequence diagram

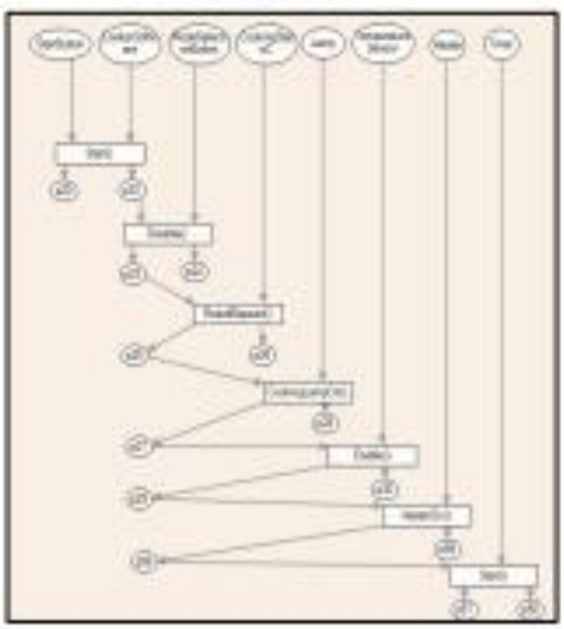

Figure 5. Start Cooking Process of Petri net diagram

It recognized the four functional processes for the total number of function size of 27 CFP. The automated system produce the result of the size of cooker system which are shown in Figure 8.

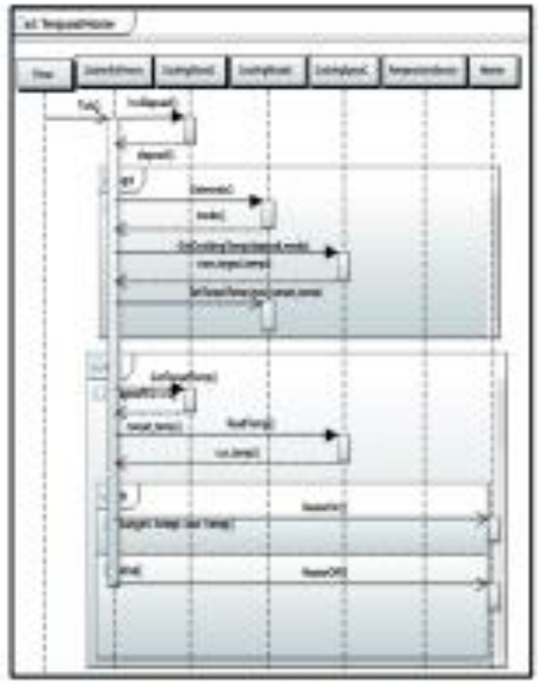

Figure 6. Temperature and Heater of UML and SysML Sequence diagram

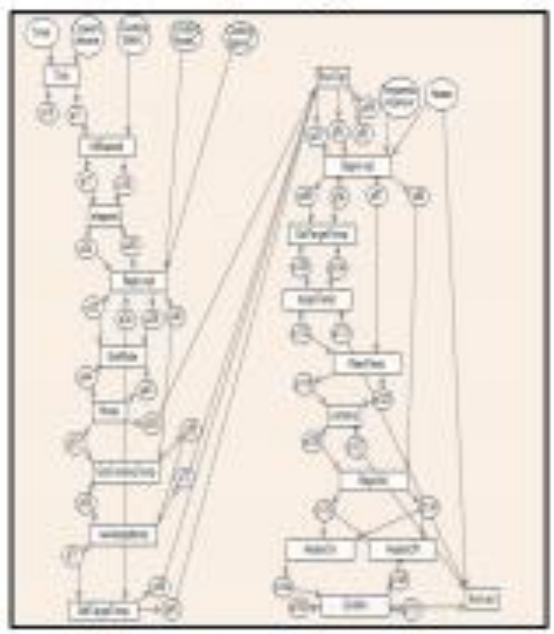

Figure 7. Temperature and Heater of Petri net diagram

Automated Software Measurement					
Functions	Entry	Exit	Read	Write	Sub Total of Function Size
Temperature and heater	2	1	3	1	7
StopCooking	1	5	0	1	7
Start Cooking Process	1	5	0	1	7
Initialization Process	1	4	0	1	6
Total Function Size					27

Generation Model by using Stereotype extension for Meta Model			
Meta Class	Lifeline/Place	Message/ Transition	
Stereotypes	Objects Power Supply Heater Temperature Sensor ModeSelectionButto n CookingModeC Start Button Lamp Timer CookingStateC CookingSpeedC Cooker Software	Entrys Boot () Start () Stop () Tick () Read Temp () Read GetCooking Temp (elapsed, mode) GetTarget Temp () GetMode () Alternatives Heater On () Heater Off () Writes Set Mode () Reset Elapsed () Set Mode(Norma l) SetTargetTe mp(new target-temp) IncElapsed ()	Exits Lamp Off () Heater Off () Disable () Enable () Disables () CookLampOn () Heater On () Enables () Start () Heater Off () Cooking Lamp Off() Disabled() Stop () Enable Mode () WarmLampOff () Heater Off () Heater On () Optionals GetMode () Heater On () GetCookingTemp (elapsed, mode) GetTargetTemp () ReadTemp () SetTargetTemp (new target-temp) Heater Off ()

Figure 8: Automated Measurement Results of Generation Model

5. Conclusion

Manual measurement is timewasting and uncertain. The automated measurement tool can measure the software size from the generation model of different diagram notations and COSMIC FSM that are proposed in this paper. In order to handle not only specific diagram but also the notation problem, firstly we developed the generation meta model. The automated measurement tool can generate COSMIC FSM from the XML representation of UML, SysML and Petri net by using generation model and mapping rules. It is expected that measurement result can be supported by many industries to increase effort and productivity. The proposed system offers function point values specially to estimate the effort and cost for the software developments.

6. References

[1] A. Sellami, H. Hakim, A. Abran, and H. Ben-abdallah, "A measurement method for sizing the structure of UML sequence diagrams", Information and Software Technology, Elsevier, 2015, vol. 59, pp. 222–232.

[2] C. Symons, "Sizing and Estimating for Real-time Software–the COSMIC-FFP method", DOD Software Tech News', Editor: Data & Analysis Center for Software, USA DOD, Rome NY, 9(3),2006, pp.5-11.

[3] H. Soubra, A. Abran, and A. Ramdane-Cherif, "A refined functional size measurement procedure for real-time embedded software requirements expressed using the simulink model", Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, IEEE, 2012, pp. 70-77.

[4] H. Soubra, L. Jacot, and S.Lemaire, "Manual and Automated Functional Size Measurement of an Aerospace Realtime Embedded System: A Case Study based on SCADE and on COSMIC ISO 19761", International Journal of Engineering Research and Science & Technology, May, 2015, vol.4, No. 2 .

[5] H. Marouane, C. Duvallet, A. Makni, R. Bouaziz and B. Sadeg, "An UML profile for representing real-time design patterns", Journal of King Saud University-Computer and Information Sciences, Elsevier, November, 2018, pp 478-497.

[6] L. Lavazza, V. Del Bianco, "A case study in COSMIC functional size measurement: The rice cooker revisited", International Workshop on Software Measurement, Springer, Berlin, Heidelberg, 2009, pp. 101-121.

[7] S. Karim, S. Liawatimena, A. Trisetyarso, B.S. Abbas, and W. Suparta, "Automating functional and structural software size measurement based on XML structure of UML sequence diagram", IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), IEEE, November, 2017, pp. 24-28.

[8] N.A.S. Abdullah, N.I.A. Rusli, and M.F Ibrahim, "A case study in COSMIC functional size measurement: angry bird mobile application", IEEE Conference on Open Systems (ICOS), IEEE, December, 2013 pp. 139-144.

[9] T. Zaw, S. Z. Hlaing, M. M. Lwin, K. Ochimizu, "A Lightweight Size Estimation Approach for Embedded System using COSMIC Functional Size Measurement", Proceedings of the 1st International Conference on Advanced Information Technologies, University of Information Technology, Yangon, Myanmar, November, 2017, pp 112-118.

[10] T. Zaw, S. Z. Hlaing, M. M. Lwin, K. Ochimizu, "Automated Size Measurement of Embedded System based on XML using COSMIC FSM", Joint Conference of the 28th Int'l Workshop on and 13th Int'l Conference on Software Process Measurement (IWSM-MENSURA), CEUR Workshop Proceedings, Beijing, China, September, 2018, pp 14-19.

[11] T. Zaw, S. Z. Hlaing, M. M. Lwin, K. Ochimizu, "The Measurement of Software Size based on Generation Model using COSMIC FSM", The 23rd International Computer Science and Engineering Conference, IEEE Xplore, 2019.

[12] The COSMIC Measurement Practices Committee, The "COSMIC Functional Size Measurement Method, version 4.0.2: Measurement Manual", 2017.

[13] https://www.iso.org/obp/ui/#iso:std:iso-iec:14143:-6:ed-2:v1:e.