A Lightweight Size Estimation Approach for Embedded System using COSMIC Functional 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

Functional Size Measurement (FSM) is an important component of a software project that provides information for estimating the effort required to develop the measured software. Although the embedded software is time-consuming to develop, COSMIC FSM can be estimated to get more accurate function size. The traditional Function Point methods are designed to measure only business application domain and are problematic in the real-time domain. As a result, COSMIC Functional Size Measurement (FSM) method is designed to measure both application domains. The design diagrams such as UML, SysML and the well-defined FSM procedure must use to accurately measure the functional size of embedded system. We have already developed the generation model based on SysML metamodel with an example of elevator control system. In this paper, we applied the generation model that is the classification of the instance level of object based on UML metamodel. After that, this paper also showed the mapping rules which mapped between the generation model and COSMIC FSM to estimate the functional size of embedded software with the case study of cooker system. This paper also proposed the light weight generation method of COSMIC FSM by using the generation model.

Keywords- Function Size Measurement (FSM); COSMIC (Common Software measurement Consortium); UML; MetaModel

1. Introduction

Software sizing is used to estimate the size of a software application. As the software development costs are increasing, the early prediction of the size of the embedded software should be achieved in the developmental process.

The functional size of software has become an important task in most of the industrial software development as it offers the valuable input to estimate the development effort model and tools. There are five measurement methods which have been recognized as standards IFPUG FPA [4], MK II FPA [5], FISMA [7], NESMA FPA [6] and COSMIC FFP [8,9] to measure the functional size of software applications. COSMIC FSM has developed the most advanced method of measuring a functional size of software that overcomes several limitations of the traditional FSM methods. The traditional FSM methods are applied in business application domain and they are difficult to apply in real-time application domain. As a result, several methods have been proposed for FSM, one of which is the COSMIC FSM method [8,9]. COSMIC was designed to apply in various functional domains such as business application domain, real-time application domain. Some researchers proposed the embedded system with some modeling languages. But these aren't applied in the generation model by using UML metamodel of embedded system to estimate the functional size of software. To address this limitation, this paper proposes the generation model that is based on UML metamodel with COSMIC FSM concept. After that the mapping rules must be produced between the generation model and COSMIC. Finally, COSMIC calculated with the case study of "Cooker System" to estimate the size of software. The rest of the paper is organized as 5 sections. Section II presents the related work. Section III presents the generation method. Section IV provides the case study. Section V presents the conclusion.

2. Related Work

In [1], SYMONS, C. described the example that COSMIC concepts can be applied in any real-time software requirements to measure the functional size of real-time software to understand clearly any software engineer with alarm example. In [2], Soubra, H., et al. proposed the design of the FSM procedure based on the documentation of the mapping of the Simulink concepts to COSMIC concepts for the embedded real-time software system. The generation model by using UML metamodel can be defined with COSMIC concept. Then, the mapping rules which map the generation model to the COSMIC model define.

This paper proposes the mapping rules that can be used in different types of embedded software system and also proposes a light weight generation method of COSMIC FSM. The resulted software sizing

measurement can be used in many software industries to increase effort and productivity.

3. Generation Method

In Figure 1, the UML sequence notation is used to obtain the functional size of software as an input. After that, the generation model based on UML metamodel has to be translated into COSMIC concept by using the mapping rules. Finally, the result of function size of software is calculated by using COSMIC method.

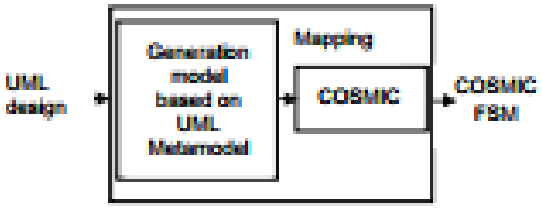

Figure 1. Flow of calculation (Processing)

Figure 2 shows the light generation method of COSMIC FSM based on generation model with three phases. In the generation phase, this paper defines generation model depending on UML metamodel that is based on the COSMIC concepts. In the mapping phase, we analyse the COSMIC concepts with the generation model to define the mapping rules. In the measurement phase, the results of the actual size measurement of the embedded system are calculated by using COSMIC method.

4. Case Study

4.1. Functional User Requirements

We adopt the specification of simple version of the Cooker system [10] to express the counting of COSMIC. After the specification of Cooker Software has defined, the UML representations can be developed. The basic functional requirements of this system are as follows:

1. When the power is switched on, the cooker software receives the input from the door and from a start button, and sends signals to switch an internal light, and the heater on or off. The software also sends signals to a timer to set the cooking time and can receive a signal from the timer when cooking is complete.
2. Cooking starts with pressing the start button provided the door is closed. If the door is open pressing the start button has no effect.
3. Opening the door during cooking turns the heater off.
4. The timer stops when the door is opened whilst cooking is in progress, or the timer signals that cooking is completed and the timer resets itself to zero.

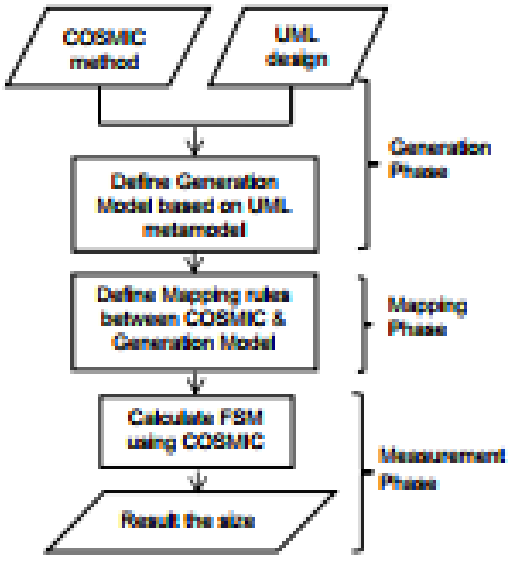

Figure 2. Calculation based on generation model

4.2. UML Model of a Case Study

The use case diagram of cooker system is shown in Figure 3. There are three actors in this system: timer, door sensor and start button that have relationship with four use cases of the system: DoorClosed, ButtonPushed, CookingEnded and DoorOpened by using the functional requirements.

4.3. UML Sequence model for COSMIC FSM

In this paper, the generation model define the classification of instance level of object depending on the partial UML sequence metamodel approach that allows reasoning about meta-elements and their relationships as shown in Figure 4. This generation model uses the UML metamodel through the profiling mechanism for COSMIC concepts. The sequence diagrams of cooker are used to apply the generation model as shown in Figure 5 to 12. The sequence diagram is appropriate to identify functional processes and data movements. Then, the number of data movements for each sequence diagram is calculated. Finally, the total size of system is calculated by aggregating all these data movements.

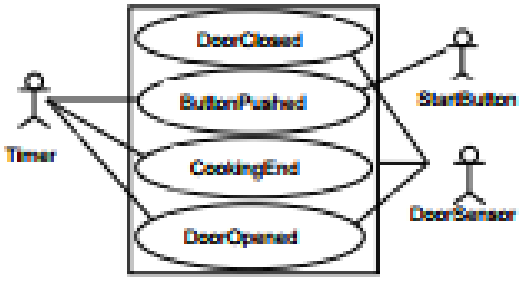

Figure 3. Use case diagram for cooker system

4.4. Mapping Rules

In this section, we define mapping rules that correspond to the COSMIC element and some instance level object of sequence diagram by using the UML metamodel concept through profiling mechanism. Table 1 summarizes the mapping rules from UML notations to COSMIC. Based on the correspondence, we define the mapping rules as follows:

Rule 1: Identification of the application boundary.
The application border in Cooker system corresponds to UML use case diagram.

Rule 2: Identification of the functional users.
COSMIC defines a (type of) user that is a sender and/or an intended recipient of data in Functional User Requirements of a piece of software. The concepts of COSMIC corresponds an objects in the sequence diagram.

Rule 3: Identification of the functional process.
It requires the data from functional user that corresponds to interaction between objects that operate with one another in sequence diagram.

Rule 4: Identifying the data groups.
A COSMIC data group corresponds to the data group that may be represented in sequence diagram by means of the flows of information between objects.

Rule 5: Identifying the four data movements.
Sequence diagram represents these data movements. Each data movements correspond to an interaction messages in sequence diagram.

Rule 5.1: Identifying the Entry data movement.
It moves the message from functional user to boundary.

Rule 5.2: Identifying the Exit data movement.
It moves the message from boundary to functional user.

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

Rule 5.4: Identifying the Write data movement.
It moves the single data group from functional process to persistent storage.

Rule 6: Applying the COSMIC measurement function.
Each of the data movement (Entry, Exit, Read and Write) that identified in each functional process is added to obtain the functional size of that process.

Rule 7: Aggregation the functional size measurements.

Table 1. Mapping rules of COSMIC element and UML

COSMIC element	UML diagram
Boundary	Use Case
Functional User	Objects in Sequence Diagram
Functional Process	Interaction between objects
Data Group	Flows of information between objects
Entry Data Movement	Sequence message from Functional User to Functional Process
Exit Data Movement	Sequence message from Functional Process to Functional User
Read & Write Data Movement	Sequence message move single data group from persistent storage to a functional process

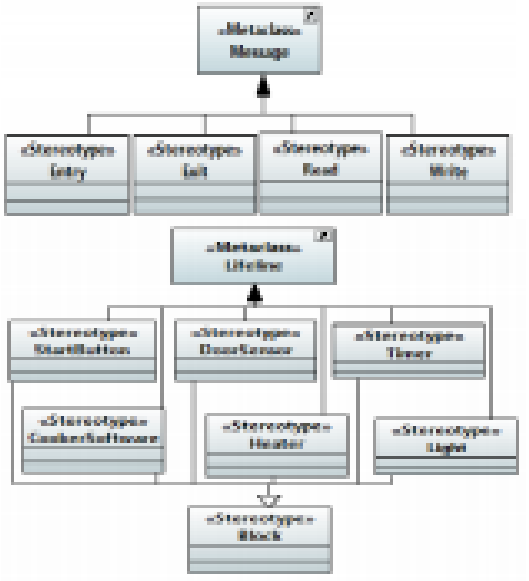

Figure 4. Generation model of classification of instance level by using UML metamodel

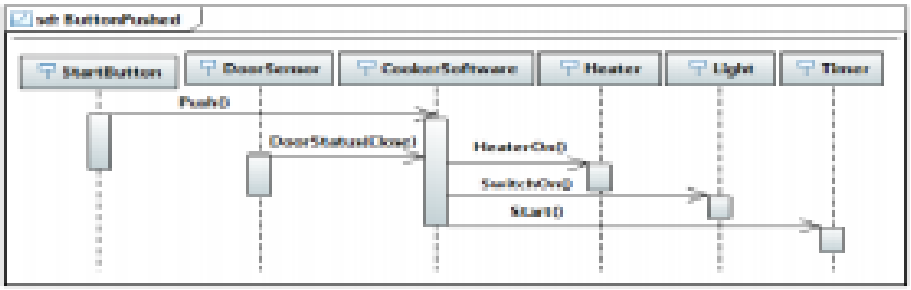

Figure 5. Sequence diagram for ButtonPushed

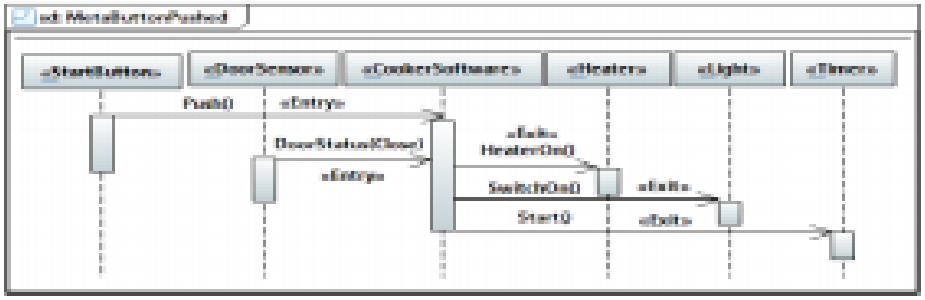

Figure 6. Sequence diagram for ButtonPushed using the generation model

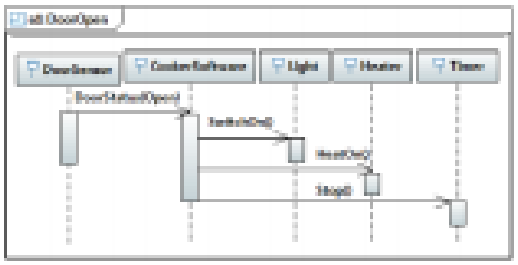

Figure 7. Sequence diagram for DoorOpen

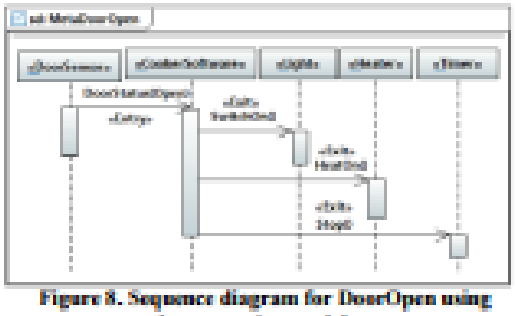

Figure 8. Sequence diagram for DoorOpen using the generation model

Figure 9. Sequence diagram for DoorClosed

All of the identified data movements of the functional processes of the whole system must be aggregated into a single functional size value by adding them together to obtain the functional size of the system.

4.5. COSMIC Data Movements

After defining the mapping phases, the data movements for each function can be calculated to estimate the size of software. These data movements are illustrated as shown in Figure 6,8,10 and 12. For the case study, the functional size identified 4 functional processes for DoorClosed, ButtonPushed, CookingEnd and DoorOpened.

Figure 10. Sequence diagram for DoorClosed using the generation model

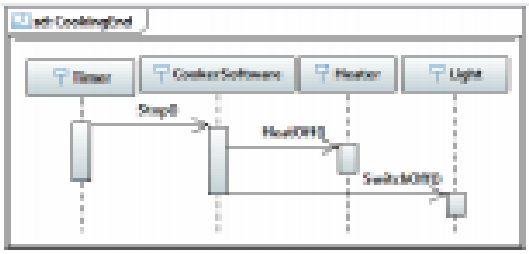

Figure 11. Sequence diagram for CookingEnd

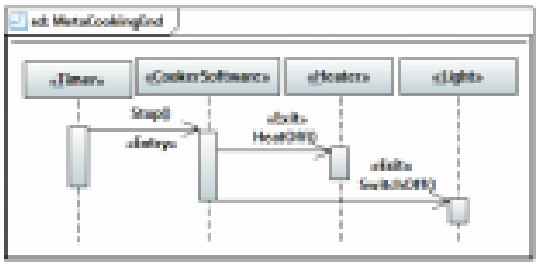

Figure 12. Sequence diagram for CookingEnd using the generation model

The COSMIC measurement standard, 1CFP is defined as the size of one data movement. In the functional process of ButtonPushed, it identified 2 Entry data movements. The Entry data movements counted the PushSignal attribute from start button and the Getdoorstatus from DoorSensor. It also identified 3 Exit data movements. The Exit data movements also counted the HeaterOn attribute to Heater, the LightOn attribute to Light and the Start attribute to Timer respectively. The subtotal of functional size for that functional process is 5CFP. By calculating the sizes from each function, the cooker system is estimated at 14CFP by adding all number of data. The data movement of each sequence diagram in this system is as shown in Table 2.

Table 2. Measurement of data movements for cooker system

Process	Message sending		Data Move_ments	CFP
	Message	Component of object involved		
Door Closed	DoorClose()	From Doorsensor	Entry	2 CFP
	Switchoff()	To Light		
Button Pushed	PushSignal()	From Start button	Entry	5 CFP
	Getdoorstatus (Close)	FromDoor Sensor	Entry	
	HeatOn()	To Heater	Exit	
	LightOn()	To Light	Exit	
	Start()	To Timer	Exit	
Cooking Ended	Stop()	From Timer	Entry	3 CFP
	Heatoff()	To Heater	Exit	
	Lightoff()	To Light	Exit	
Door Opened	Doorstatus (open)	From Doorsensor	Entry	4 CFP
	Lighton()	To Light	Exit	
	Heatoff()	To Heat	Exit	
	Stop()	To Timer	Exit	
Total				14 CFP

4.6. Automated Measurement

An automation tool will help in applying and using the COSMIC method in many industries. We develop the prototype tool for the automatic measurement of the functions as an example in Java. This paper uses the latest version of COSMIC measurement, version 4.0.1[8].

4.7. Result of Cooker System

We also develop the automated function size measurement from the requirements with cooker system. It is from the generation model based on UML metamodel. The measurement result of the cooker system is as shown in Figure 13. It gives the total function point number identified by adding the total number of Entries, Exits, Reads and Writes of the system. We have already applied our method by describing to another example. So, we apply our light weight generation method to both case studies successfully in generation model by using the SysML and UML metamodels.

4.8. Result of Elevator System

We have succeeded in automated calculation of the COSMIC measure with elevator system that is from the SysML generation model by using SysML metamodel [3].
Figure 14 shows our prototypes tool's interfaced and describes the measurement results of the elevator system.

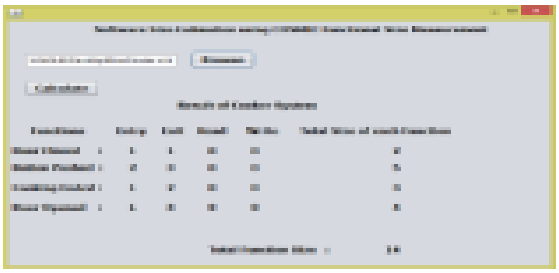

Figure 13. Automated measurement results of Cooker System

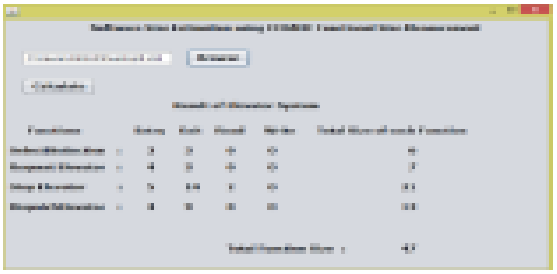

Figure 14. Automated measurement results of Elevator System

5. Conclusion

Sizing with COSMIC is an excellent way of controlling the quality of the requirements. It improves estimating accuracy, especially for larger software projects. COSMIC FSM is a great importance in industrial software development, since it provides the necessary input to effort estimating models. In this paper, we showed the automated generation method of COSMIC FSM by using generation model based on UML metamodel. This generation method uses profiling mechanism and is also called the light weight generation method. We perform the case study of the cooker system which shows the usefulness of our light weighted approach.

I intend to develop the generation model depending on metamodel for basic modeling languages such as UML, SysML etc. which is useful for estimation the size of embedded system. After that, the general rules which allow mapping these generation model to COSMIC concepts must be defined. I hope to develop the automatic measurement of COSMIC size for different models designed in embedded system.

6. References

[1] Symons, C.: "Sizing and Estimating for Real-time Software – the COSMIC-FFP method", In: DOD Software Tech News, Rome NY, vol. 9(3), 2006, pp. 5–11.

[2] Soubra, H., Abran, A. , Stern, S. , Ramdan-Cherif, A., "Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed using the Simulink Model", Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), 2011.

[3] Thandar Zaw, Myint Myint Lwin, Koichiro Ochimizu, Swe Zin Hlaing, "Software Size Estimation for Embedded Software using COSMIC FSM", Second Workshop on Advanced Technology, Myanmar, December 2016.

[4] IFPUG (The International Function Point Users Group), Software Engineering- IFPUG 4.1 Unadjusted Functional Size Measurement Method- Counting Practices Manual, Switzerland,2009.

[5]UKSMA(United Kingdom Software Metrics Association), Software Engineering-Mk II Function Point Analysis- Counting Practices Manual, Switzerland, 2002.

[6]NESMA(Netherlands Software Metrics users Association), Function Size Measurement Method version 2.1 – Definitions and Counting Guidelines for the application of Function Point Analysis , Switzerland , 2005.

[7] FiSMA (Finnish Software Measurement Association), Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method, Switzerland, 2008.

[8] COSMIC (Common Software Measurement International Consortium), The COSMIC Functional Size Measurement Method Version 3.0.1: Measurement Manual, 2009.

[9] COSMIC, The COSMIC Functional Size Measurement Method version 4.0.1: Measurement Manual, 2015.

[10] COSMIC, The COSMIC Functional Size Measurement Method, version 4.0.1: Guideline for Sizing Real-time Real-Time Embedded Software, 2016.