Software Size Measurement for Embedded System using Class Diagram

Thandar Zaw
University of Information Technology, Yangon, Myanmar
thandarzaw@uit.edu.mm

Swe Zin Hlaing,
Myint Myint Lwin,
University of Information Technology, Yangon, Myanmar
swezin@uit.edu.mm,
myintmyintlwin@uit.edu.mm

Koichiro Ochimizu
University of Information Technology, Yangon, Myanmar
ochimizu@jaist.ac.jp

Abstract

Todays, software size measurement is the essential roles for project management. It can measure before developing in software to get the accurate size of software. Due to obtain the functional size of software, there are many measurement methods that have been recognized as international standard. Some measurement methods are designed for business application software and a few methods are designed for real-time application software. Thus, in order to measure the functional size of embedded system correctly, the well-defined class diagrams notation and well-designed FSM procedures should be used in embedded system. So, *COSMIC FSM is one of the well-known methods of FSM which is suitable to estimate the size of embedded software. This paper proposes UML class design notations which can be used to estimate the size of embedded software. This paper also shows the mapping rules which mapped between the class diagram and COSMIC FSM to measure the functional size of software. Finally, the functional size of software is calculated by using COSMIC FSM.*

Keywords- Common Software Measurement International Consortium (COSMIC FSM), UML class diagram.

1. Introduction

Software sizing is used to estimate the size of software application. Several size estimation methods have been proposed. The most popular size estimating methods are Source Line of Code (SLOC) which is based on size related measures and Function Points which is based on function-related measures. Functional size measurement is an important way of measuring software in the early stages of development when the effort and cost estimation is most needed. Several Function Points measurement was recognized as an international standard. The IFPUG [1], MKII [2], COSMIC [5,6 ,9], NESMA [3] and FISMA [4] were also defined and standardized. Among them, one of the functional size measurement methods is COSMIC FSM which was designed to be applied in various functional domains such as business application domain and real-time application domain.

Many researchers proposed the software estimation methods which are applicable for UML sequence diagram notations and have not proposed for UML class diagram. To address this limitation, this paper proposed the UML class diagram notations to estimate the functional size of software. These notations have been translated to COSMIC by using mapping rules with a simple case study of cooker system. This paper is organized as follows: the second section provides related work; the third section presents the proposed system; the fourth section explains the case study and final section is conclusion and future work.

2. Related Work

In [7], Symons, C. described the COSMIC concepts that can be applied in any real-time software requirements to measure the functional size of real-time software to understand clearly for any software engineer with alarm example. In [8], 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. In [11], Luigi Lavazza., et al proposed the UML that can be used to build models according to the COSMIC measurement rules. Asma Sellami et al. [12] proposed the measurement method for sizing of sequence diagram that can be measured both the functional and structural size at different level of granularity. In [10], the author proposed the automated functional and structural measurement of software size from XML structure of sequence diagram to calculate COSMIC CFP. In [13], the author presented a COSMIC based FSM procedure for RTS (Real-Time embedded systems) designed with SCADE and manually applied the FSM procedure to an aerospace system and also compared the results obtained with automatically by a prototype tool. This paper proposed the UML class design model notation. Then, the mapping rules define which can map between these notation and COSMIC.

3. Proposed System

This section described the proposed method based on the COSMIC that uses the specification expressed UML class diagram notation in embedded system to obtain the functional size of software. The proposed system has three measurement processes which are shown in Fig 1. In the measurement strategy phase, the proposed system is analyzed from the popular notation of UML class diagram. After analyzing the class diagram with COSMIC, the mapping rule apply between these diagram and COSMIC during the mapping phase. The functional size of software is calculated by using the COSMIC in the final state of measurement process.

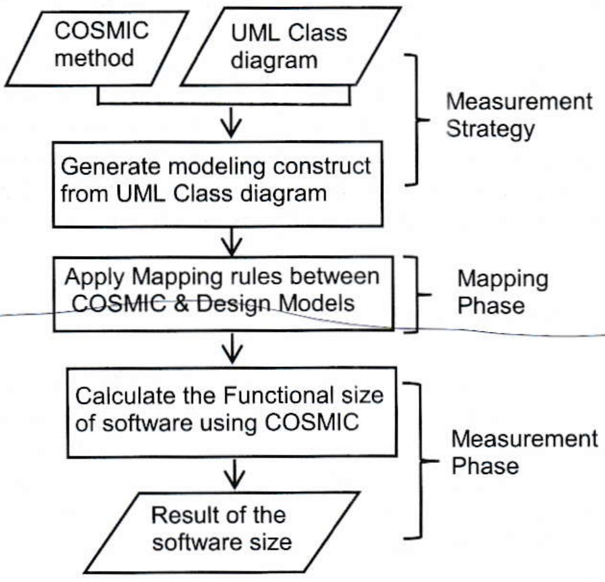
COSMIC method
UML Class diagram
Generate modeling construct from UML Class diagram
Measurement Strategy
Apply Mapping rules between COSMIC & Design Models
Mapping Phase
Calculate the Functional size of software using COSMIC
Measurement Phase
Result of the software size

Figure 1. Proposed System based on UML Class diagram

4. Case Study

The UML use case and class diagram are used in the case study of cooker system. Then the total no of data movements for each class diagram is calculated.

4.1. Functional User Requirement

The paper proposed the specification of a simple version of the cooker system which is used as a case study to determine the counting of COSMIC [13] .Before developing the UML representation, the specification of the cooker system must be defined. The functional user requirements of this system are as follows:

1. The cooker software can get the input from a door sensor and start button. Then it can show the light and heater on/off when the power is switched on.
2. When the start button is pressed and the door is closed, the cooking starts. If the door is open, the start button has no effect.
3. Either the door is open while the cooking is in progress or when cooking is completed, the timer signals will stop.

4.2. Use Case of UML Model

The use case diagram of cooker system is shown in Fig. 2. This system consists of three main functionalities: Pressed Button, Opened Door and End Cooking. The functional users of this system are Door Sensor, Start Button, Timer, Light and Heater.

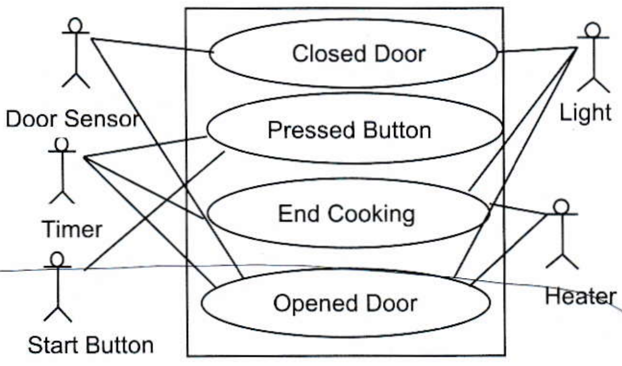

Figure 2. Use case diagram of cooker system

4.3. COSMIC FSM in UML Class diagram

The measurement process of the cooker system mainly comes from the design requirements of UML class diagram. In this system, there are three functional processes of UML class diagrams for the cooker system as shown in Fig. 3 to 5. In Fig. 3, the cooker checks that the door is open or closed. When the door is closed, it sends the signals to start the heater and to switch on the light. The dependency is a relationship between named elements such as class diagram. It is appropriate to identify functional size of class diagram depends on the number and types of messages exchanged. Usage is a dependency in which one named element (client) requires another named element (supplier) for its full definition or implementation. Call is a usage dependency that specifies that the source operation invokes the target operation. This dependency may connect a source operation to any target operation that is within the scope including, but not limited to, operations of the enclosing classifier and operations of other visible classifiers. Call is denoted with the standard stereotype «call» whose source is an operation and whose target is also an operation. Send is a usage dependency whose source is an operation and whose target is a signal, specifying that the source sends the target signal. Send is denoted with the standard stereotype «send».

The types of messages with the corresponding measurement results in CFP units are shown in Table 1. Then, the number of data movements for each class diagram is calculated. Finally, the total size of system is calculated by aggregating all these data movements.

Table 1 Message types in a UML class diagram

Usage dependency	Symbol	Standard stereotype	Data Movement	CFP units
Call	-------->	<<call>	R or W	1
Send	-------->	<<send>>	E or X	1

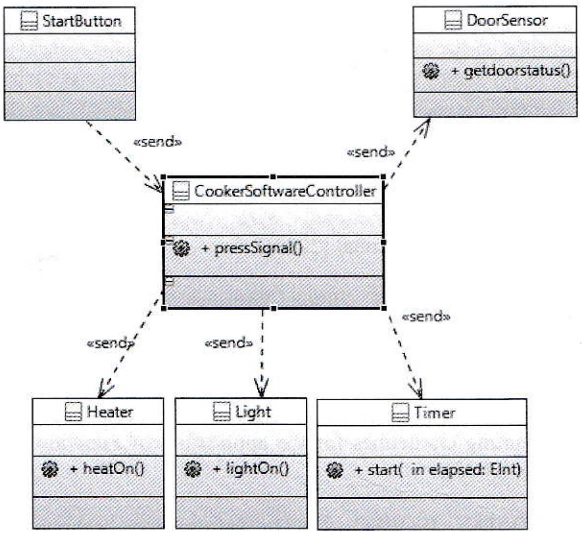

Figure 3. Button Pressed of UML Class diagram

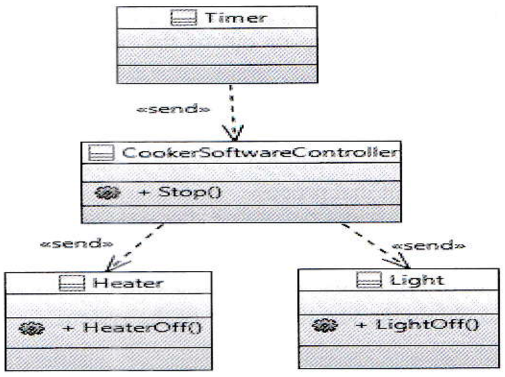

Figure 4. End Cook of UML Class diagram

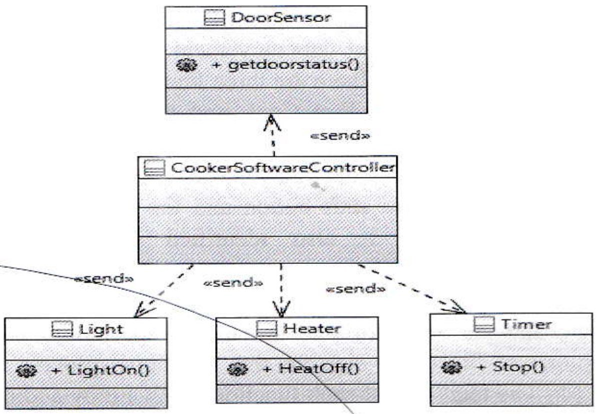

Figure 5. Open Door of UML Class diagram

4.4. Mapping Rules

In this section, the key concepts of COSMIC are mapped to the key concepts of UML class diagram notation.

The mapping rules are described as follows:

Rule 1: Identify the boundary.

The boundary is represented in use case diagram. It shows the application border that is established by identifying the external elements and application system.

Rule 2: Identify the functional user.

The functional user is a class that represents an object or a set of objects that share a common structure and behavior.

Rule 3: Identify functional process.

The functional process identifies use cases in the system.

Rule 4: Identify data groups

The data groups identify that trigger event carries data between objects.

Rule 5: Identify four data movements.

The four data movements are identified as follows:

Rule 5.1: The Entry data movement identify from Rule 2 to Rule 3.

Rule 5.2: The Exit data movement identify from Rule 3 to Rule 2.

Rule 5.3: The Read/Write data movement identify messages that send into or out of the internal persistent storage.
Rule 6: Apply the COSMIC measurement function.
According to COSMIC measurement, each of the data movement in each functional process is added to get the functional size of that process.
Rule 7: Aggregate the functional size measurements.
Aggregate all of the data movements of the functional processes of the whole system into a single functional size value to obtain the functional size of the system.

4.5. Measurement Phase

After defining the mapping rules, the data movements of each functional process have been identified as shown in Fig. 3 to 6. According to COSMIC standard, 1CFP is defined as the size of one data movement.

For the case study, there are three functional processes such as Push Button, End Cook and Open Door. The functional process of Push Button identified 2 Entry data movements. The Entry data movements counted the Push Signal 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. The total size of each function is 12CFP by adding all number of data movements. The data movement of each sequence diagram in this system is as shown in Table 2.

5. Conclusion and Future Work

FSM, an important component of a software project, provides information for estimating the effort required to develop the measured software. The early prediction of the size of the embedded software can be achieved in the developmental process as the software development costs are increasing. In this paper, we proposed the UML class diagram notation with COSMIC FSM which is applicable to the case study in cooker system. Then, the mapping rules between these notation and COSMIC FSM define to support the functional size measurement of software. It has been intended to propose the various notations by extending the COSMIC rules.

Table 2. Measurement of data movements for cooker system

Process	Message Sending		Data Move ments	C FP
	Message	Component of object involved		
Button Pressed	PressSignal( )	Start button	Entry	5 C FP
	Getdoorstatus (Close)	Door Sensor	Entry	
	HeatOn( )	To Heater	Exit	
	LightOn( )	To Light	Exit	
	Start Cooking( )	To Timer	Exit	
End Cook	Stop( )	Timer	Entry	3 C FP
	HeatOff( )	Heater	Exit	
	Lightoff( )	Light	Exit	
Door Opened	Doorstatus (open)	Doorsensor	Entry	4 C FP
	LightOn( )	Light	Exit	
	HeatOff( )	Heat	Exit	
	Stop( )	Timer	Exit	
			Total	12 C FP

6. References

[1] ISO/IEC: ISO/IEC 20926, Software Engineering- IFPUG 4.1 Unadjusted Functional Size Measurement Method- Counting Practices Manual (2009).

[2] ISO/IEC: ISO/IEC 20968, Software Engineering- Mk II Function Point Analysis- Counting Practices Manual (2002).

[3] ISO/IEC: ISO/IEC 24570, Software Engineering- NESMA Function Size Measurement Method version 2.1 – Definitions and Counting Guidelines for the application of Function Point Analysis (2005).

[4] ISO/IEC: ISO/IEC 29881, Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method (2008).

[5] COSMIC. The COSMIC Functional Size Measurement Method Version 3.0.1, Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003).

[6] COSMIC. The COSMIC Functional Size Measurement Method Version 4.0: Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2011).

[7] Symons, C.: "Sizing and Estimating for Real-time Software – the COSMIC-FFP method". In: DOD Software Tech News',

Editor: Data & Analysis Center for Software, USA DOD, Rome NY, vol. 9(3), pp. 5–11 (2006).

[8] 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", Software Measurement, 2011 Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA).

[9] The ''COSMIC Functional Size Measurement Method, version 4.0: Guideline for Sizing Real-time Real-Time Embedded Software", 2016.

[10] Meiliana etal. ,"Automating Functional and Structural Software Size Measurement based on XML Structure of UML Sequence Diagram ", 2017 IEEE International Conference on Cybernetics and Computational Intelligence 20-22 Nov. 2017.

[11] Luigi Lavazza and Vieri Del Bianco, "A Case Study in COSMIC Functional Size Measurement: the Rice Cooker Revisited", IWSM/Mensura 2009.

[12] A. Sellami. etal, "A measurement method for sizing the structure of UML sequence diagrams", Information and Software Technology 59, 2015.

[13] Hassan Soubra, Laury Jacot and Steven Lemaire, "Manual and Automated Functional Size Measurement of an Aerospace Real Time Embedded System: A Case Study Based on SCADE and on COSMIC ISO 19761", International Journal of Engineering Research and Science & Technology, vol.4, No. 2, May 2015.