Software Size Estimation for Embedded Software using COSMIC Functional Size Measurement

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

Myint Myint Lwin, Koichiro Ochimizu
Swe Zin Hlaing
Faculty of Information Science
University of Information Technology
Yangon, Myanmar
myintmyintlwin@uit.edu.mm,
ochimizu@jaist.ac.jp, swe@uit.edu.mm

***Abstract*— Software sizing estimates the probable size of a piece of software while effort estimation predicts the effort needed to build it. Measurement of software size from user requirements is crucial for the estimation of the developmental time and effort. The generated Function Point methods are designed for business application domain but it is difficult to estimate the real-time software domain such as embedded software system. Therefore, COSMIC Functional Size Measurement (FSM) method is applied to design both in the business application software domain and the real-time software domain. To correctly measure the functional size of embedded system's requirements documented and modelled by using the popular tools, FSM procedures must be designed and well defined in embedded system development. Firstly, this paper proposes SysML design notations which can be used to estimate the size of embedded software. After that these design notations can be applied to the COSMIC FSM. SysML representation involves sequence diagram to represent the functional requirements. This proposal uses one of the embedded systems. It shows the usage of SysML sequence diagram with COSMIC FSM which is applicable to the case study in elevator control system. SysML sequence meta model can be defined depending upon the SysML design diagram by extending stereotype through the specific profiling mechanism .Finally, the mapping rules between the meta model and COSMIC FSM define to support the functional size measurement of software.**

Keywords*— *Function Size Measurement (FSM); COSMIC (Common Software measurement Consortium); SysML (System Modeling Language); Meta Model

I. Introduction

The functional size of software is an important input for planning, buying, developing, improving, and maintaining software systems [1]. In particular, it provides valuable information for estimating the effort required to develop the measured software. Based on that estimation, software managers can successfully plan resources and estimate costs for the software project [2]. To measure the functional size of software applications, five measurement methods have been recognized as standards: IFPUG FPA [2], MK II FPA [3], FISMA [5], NESMA FPA [4] and COSMIC FFP [6,7]. The former four functional size measurements 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 [3]. COSMIC was designed to apply in various functional domains such as business application domain, real-time application domain. After that some of researchers proposed the COSMIC FSM that can be applied in Business Process Modeling such as Business Process Modeling and Notation (BPMN) and Qualigram. Some researchers proposed the embedded system with modeling language such as UML, Simulink. But these aren't applied in various modeling notations using met model in embedded system to estimate the functional size of software. To address this limitation, this paper proposes the general modeling rules from the requirement document of different notations in embedded system. Then, SysML (System Modeling Language) meta model must define for different notations with COSMIC FSM concept. After that the mapping rules must be produced between these met model. Finally, COSMIC can be applied in case studies as complement to the measurement manual to support in industrial development. To improve the applicability of COSMIC in elevator control system this is modeled by SysML sequence diagram. The COSMIC measurement is mapped with SysML met model before it is used as guideline for calculating the COSMIC Function Point (CFP) based on Entry, Exit, Write or Read data movements.

Sizing with COSMIC is an excellent way of controlling the quality of the requirements. It improves estimating accuracy, especially for larger software projects. The rest of the paper is organized 3 sections. Section II presents the related work. Section III presents the proposed system. Section IV provides the conclusion.

II. Related Work

In [8], 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 [9], 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 SysML meta model can be defined with COSMIC concept. Then, define the mapping rules which can map the meta model construct to the COSMIC meta model. This paper proposes the mapping rules that can be used in different types of embedded software system. The resulted software sizing measurement can be used in many software industries to increase effort and productivity.

III. Proposed System

In Fig. 1, the design of the FSM procedure based on the COSMIC method uses the specification expressed popular common tools in embedded system development to obtain the functional size of software. Each element of the design model has to be translated into COSMIC concept by using the rules. The met model must be defined to describe the basic languages such as UML, SYSML etc. used in Embedded software system. After that meta model must be produced by using profiling mechanism for COSMIC FSM. Finally, the mapping rules must be defined between meta modelling construct and COSMIC concepts.

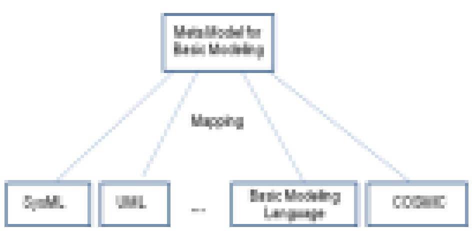

Fig. 1. Design Diagram

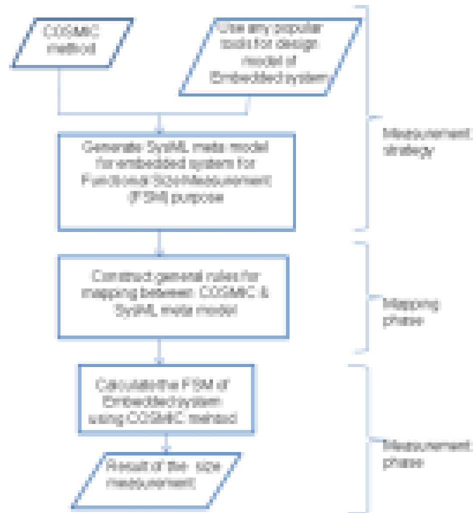

Fig. 2. Proposed System Design

Fig. 2 shows the proposed system with three phases. In the measurement strategy phase, the purpose of this procedure is to apply the COSMIC method and the design model of the embedded system to define the met model. After analysing the COSMIC concepts with the design model constructs, the mapping rules between these COSMIC and the modelling constructs of the design model must be constructed. Finally, the results of the actual size measurement of the embedded system are calculated using COSMIC method.

A. The Elevator Controller: Function User Requirements

In this paper, the specification of simple version of the elevator control is used as a case study to express the counting of COSMIC. The proposed system took the embedded system case study "Elevator System" [10]. Before the SysML representations can be developed, the specifications of elevator control system must be defined. The basic functional requirements of this system are as follows:

1. When the elevator user requests the destination floor service by pressing elevator button(s).

2. The elevator user at a floor presses the request button. The user may go either to a upper floor or a lower one, depending on the current moving direction of the elevator.

4. The user want to stop the car – when the elevator is required to stop, the arrival sensor detects the elevator that is approaching a floor. The system checks and commands to stop the elevator.

6. When the elevator is at floor and there is a floor to move to the next request, the system commands the motor to move the elevator.

B. SysML Model of the Elevator Control System

Firstly, the use case is illustrated for general process of the elevator control system as shown in Fig. 3, which is useful for several purposes:

- It shows the boundary between the functional user and elevator software.
- It indicates the external elements with four data movements.

The two actors in elevator system are the elevator user, which is the role that humans play when interacting with the system and other is arrival sensor to detect the floors of elevator. The user corresponds to the requirements 1, 2 and 4. The arrival sensor corresponds to the requirement 3. Therefore the use case diagram shows that the actor has relationship with four use cases of the system: Process SelectDestination, RequestElevator,StopElevatoratFl-oor and DispatchElevator.

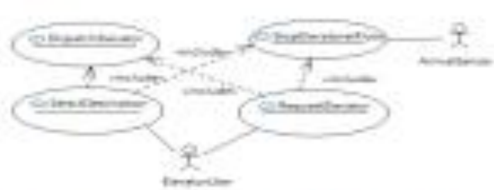

Fig. 3. Use case Diagram for Elevator Control System [10]

C. Sysml Sequence Meta model for COSMIC FSM

The Partial SysML sequence meta model can be defined through the UML profiling mechanism for COSMIC FSM as shown in Fig. 5. That depends on the context class diagram in Fig. 4. The sequence diagram is appropriate to identify functional processes and data movements. And then, the sequence diagram of elevator system [10] can be used to apply the partial meta class as shown in Fig. 6 to 13. 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. The data movement of each sequence diagram is illustrated in TABLE I. TABLE I consists of data movements for elevator system. When the elevator user pressed the button, the 'triggering Entry' will start the functional process.

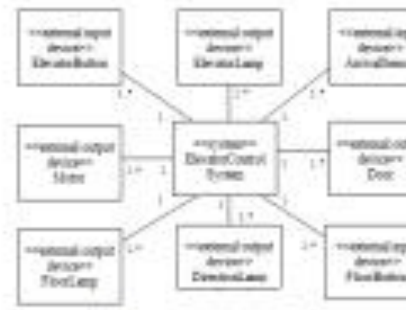

Fig. 4. Context class diagram for elevator control system [10]

Fig. 5. Partial Sequence Meta model for COSMIC FSM concepts

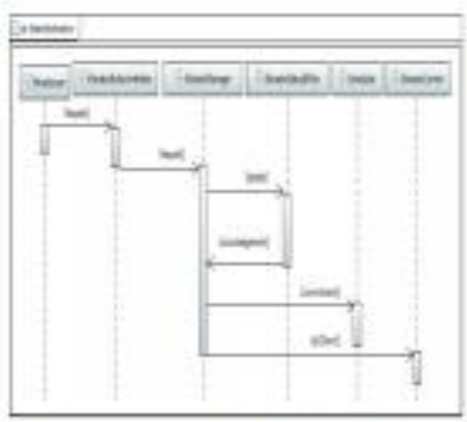

Fig. 6. Sequence diagram for Select Destination [10]

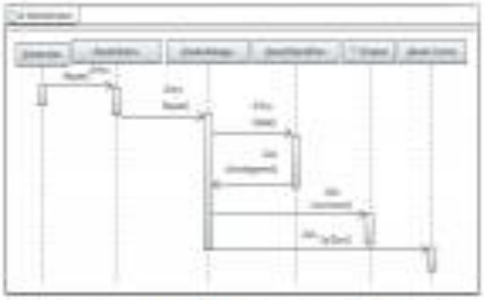

Fig. 7. Sequence diagram for Select Destination using meta class

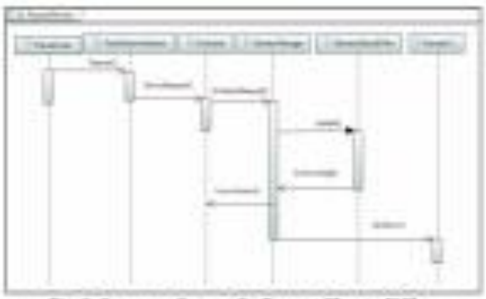

Fig. 8. Sequence diagram for Request Elevator[10]

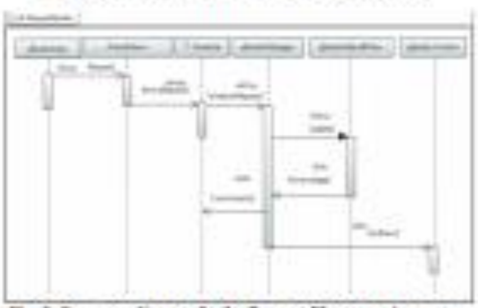

Fig. 9. Sequence diagram for for Request Elevator using meta class

Fig. 10. Sequence diagram for Dispatch Elevator [10]

Fig. 11. Sequence diagram for Dispatch Elevator using meta class

Fig. 12. Sequence diagram for Stop Elevator[10]

D. Mapping Rules

The case study is the description of COSMIC rules and mapping of SysML profile for the elevator control system. The calculation of COSMIC uses the SysML sequence representation.

Fig. 13. Sequence diagram for Stop Elevator using meta class

Rule 1: The scope of functional size measurement is the functional user requirements. For case study, the specification of elevator system is described.

Rule 2: Identify the application boundary in measurement. The application border in case study can be traced in SysML use case diagram.

Rule 3: Identify 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 user are actor in SysML sequence diagram. For the elevator case study, the functional user is the passenger as shown in sequence diagram. The passenger could stay be either humans who press the keys or hardware device.

Rule 4: The functional process requires data from functional user is shown in sequence diagram.

Rule 5: Identifying the data groups. According to COSMIC "a data group is a distinct, nonempty, non ordered and non redundant set of data attributes" that describes an "object of interest". A data group may be represented in sequence diagram by means of the flows of information between objects. For example, "Stop" in Figure 11: the flow information from the Door Time to elevator control includes the data group "After".

Rule 6: Identifying the four data movements. Sequence diagram represents these data movements. Each data movements correspond to an interaction message. Entry data movement moves the message from functional user to boundary. Exit data movement that moves the message from boundary to functional user. Read/Write data movements that move the single data group from persistent storage to functinal process.

Rule 7. The COSMIC measurement function can be applied to measure the functional size of application. According to COSMIC measurement function, each of data movements (Entry. Exit, Read and Write) are required to be added, changed and deleted.

Rule 8. Aggregate the results of the measurement function, as applied to all identified data movements, into a single functional size value.

E. COSMIC Data Movements in Elevator Control System

After mapping phases, the data movements for each function can be calculate to estimate the size of software. These data movements are illustrated in Figure 6,8,10 and 12. For case study, the functional size for select destination, request elevator, dispatch elevator and stop are calculated. By calculating the sizes from these functions, the elevator control system is estimated at 45 CFP by adding all number of data.

TABLE I. Measurement of Data movements for Elevator System

Process	Message sending		Data Movement(s)	CFP
	Message	Component or object involved		
Select Destination	Request	From Elevator User	Entry	6 CFP
	Request	From Elevator Button Interface	Entry	
	Update	From Elevator Manager	Entry	
	Acknowledge	To Elevator Manager	Exit	
	Commitment	To Scheduler	Exit	
	UpEvent	To Elevator Control	Exit	
Request Elevator	Request	From Elevator User	Entry	7 CFP
	Service Request	From Floor Button Interface	Entry	
	Scheduler Request	From Scheduler	Entry	
	Update	From Elevator Status & Plan	Entry	
	Acknowledge	To Elevator Manager	Exit	
	Commitment	To Scheduler	Exit	
	UpEvent	To Elevator Control	Exit	
Dispatch Elevator	UpRequest	From Elevator Status & Plan	Entry	8 CFP
	Close Door	To Door Interface	Exit	
	Off Up Floor Lamp	To Floor Lamp Interface	Exit	
	Floor Lamp	To Floor Lamp	Exit	
	Close Door	To Door	Exit	
	Up	To Motor Interface	Exit	
	Off Up Direction Lamp	To Direction Lamp Interface	Exit	
	Start Up Motor	Motor	Exit	
	Motor Response	Motor Interface	Exit	
	Elevator Started	Elevator Control	Exit	
	Departed	To Elevator Status&Plan	Exit	
Stop	Input	From Arrival Sensor	Entry	[illegible] CFP
	Approaching Floor(Floor#)	From Arrival Sensor Interface	Entry	
	Check this Floor(Floor#)	To Elevator Status&Plan	Read	
	Approaching Requested Floor(Floor, Direction)	To Elevator Status&Plan	Entry	
	Stop	To Motor Interface	Exit	
	Direction Lamp	To Direction Lamp Interface	Exit	
	Direction Lamp	To Direction Lamp	Exit	
	Stop Motor	To Motor	Exit	
	Motor Response	To Motor Interface	Exit	
	Stop	To Elevator Status&Plan	Exit	
	Open Door	To Door Interface	Exit	
	Off Elevator Lamp	To Elevator Lamp Interface	Exit	
	Arrived Floor	To Elevator Status&Plan	Exit	
	Arrived Floor	To Scheduler	Exit	
	Open Door	To Door	Exit	
	Door Response	To Door Interface	Exit	
	Door Opened	To Elevator Control	Exit	
	Start	To Door Timer	Entry	
	After	To Elevator Control	Read	
	Check Next Destination	To Elevator Status&Plan	Entry	
	UpRequest	To Elevator Control	Exit	

IV. Conclusion

This paper will be proposed the mapping rules that can be used in different types of embedded software system. The resulted software sizing measurement can be used in many software industries to increase effort and productivity. FSM, an important component of a software project, 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 the software size to get more accurate function size. 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 these partial SysML meta model of sequence diagram with COSMIC FSM which is applicable to the case study in elevator control system. Then, the mapping rules between the SysML meta model and COSMIC FSM define to support the functional size measurement of software. We succeeded in calculating the COSMIC measure from the SysML sequence diagrams using meta model and mapping rules. In future work, we aim at automating the functional size using COSMIC and developing the evaluation method of our system.

ACKNOWLEDGEMENT

The author would like to express her appreciation to professors for their guidance during this research. The author would also like to show her gratitude to the rector and teachers of UIT.

REFERENCES

[1] ISO/IEC: ISO/IEC 19761, Software Engineering- CFF- A Functional Size Measurement Method (2011).

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

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

[4] 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).

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

[6] COSMIC. The COSMIC Functional Size Measurement Method Version 3.0.1, Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003). The Common Software Measurement International Consortium (COSMIC), 2009.

[7] The 'COSMIC Functional Size Measurement Method, version 4.0: Measurement Manual'. (The COSMIC Implementation Guide for ISO 19761: 2011).

[8] 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).

[9] 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).

[10] Hassan Gomaa, "Designing Concurrent, Distributed and Real-Time Applications with UML", Addison-Wesley, Object Technology Series, 2000.