How to cite this paper:

Azizah Suliman, Nursyazana Nazri, & Surizal Nazeri. (2017). Applying a new hybrid model of embedded system development methodology on a flood detection system in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 215-221). Sintok: School of Computing.

APPLYING A NEW HYBRID MODEL OF EMBEDDED SYSTEM DEVELOPMENT METHODOLOGY ON A FLOOD DETECTION SYSTEM

Azizah Suliman[1], Nursyazana Nazri[2], Surizal Nazeri[3]

[1] *Universiti Tenaga Nasional, Malaysia,* azizah@uniten.edu.mymailto:dejl@sunmoon.ac.kr
[2] *Universiti Tenaga Nasional, Malaysia, syazana@gmail.com*
[3]*Universiti Tenaga Nasional, Malaysia, surizal@uniten.edu.my*

ABSTRACT. The embedded system development methodology has not been as well established as the development methodology in software engineering. Involving inter-disciplinary activities in the development of hardware and software need to be further considered when developing an embedded system. This paper presents a new development model for an embedded system by the hybridization of selected development methodologies in software engineering and systems engineering, considering that they are both essential in embedded system. The model is harmonized with embedded system design vital tasks and also non-functional properties following the ISO/IEC 9126 standard. The model is then applied to a flood detection system for verification purposes. With the phases and steps in the new hybrid model for embedded systems development methodology carefully followed, the system was built in a more systematic and structured manner addressing the every needs of an embedded system's requirements.

Keywords: Embedded System, Development Methodology, Flood Detection System.

INTRODUCTION

A general definition given by the Institute of Electrical and Electronics Engineers (IEEE) of embedded system is; it is a computer system that is part of a larger system and performs certain specialized tasks for specific purposes that satisfy some of the requirements of that system. Therefore, every system with embedded software alongside a specific goal can be termed as an embedded system. An embedded system is constrained by its extra-functional properties, and those extra-functional properties for one particular embedded system might differ from another. For example, the reliability property in an automatic chocolate vending machine (small-scale embedded system) might not be in an equal priority like in a military defence system (sophisticated embedded system). Each type of embedded systems has its own concern of quality properties. Therefore, that explains on why embedded system does not have a generic methodology that can be applied into all domains. The drawback of generality is inefficiency. This statement is supported by Jerraya and Wolf (2005) and Crnkovic (2005).

Embedded system does not have its own development methodology. Constructing a development methodology for designing embedded system can be challenging as one must understand that it is an interdisciplinary activity – a hybrid of hardware and software. This research proposes the idea of integrating software engineering and systems engineering development methodology for embedded system, specifically for a medium-scale size (Suliman & Nazri (2014a), (2014b)). An in-depth review was made on both development methodologies to see which elements can be borrowed and which elements can be omitted to suit the need of the medium-scale embedded system. The model was developed by following Recognize-Analyze-Synthesis (RAS), which is the method used to derive phases in a system engineering life cycle. The proposed hybrid model is then harmonized with embedded system design vital tasks and also non-functional properties following the ISO/IEC 9126 standard.

Due to the fact that embedded system has a part of software in it, it is acceptable for the researchers to follow the international standard for software quality. According to IEEE, software quality is the extent to which the software consist a combination of properties, such as reliability, functionality, usability, efficiency, maintainability, portability, etc. In his work of providing taxonomy for the quality properties of embedded system, Ramesh (2012) mentioned that the quality properties selected for the classification are reliability, usability, efficiency, maintainability, and memory consumption. The selection of these quality properties are based on the ISO/IEC 9126 model, which is the international standard for software quality and these are the closest properties that suit an embedded system.

It is now widely recognized that the non-functional or extra-functional properties of a system are at least as important as its somewhat more classical functional properties and that, therefore, they must be considered as early as possible in the development cycle in order to avoid costly failure (Chung, 1999). The model will consider the characteristics at every step of the development process. The model is then verified on the development a flood-detection and warning system as a case study of the proposed model.

HYBRID MODEL OF EMBEDDED SYSTEM DEVELOPMENT METHODOLOGY

The proposed hybrid model is derived by incorporating selected elements from V-model, systems engineering as well as embedded system design tasks. The selected steps are incorporated and a model is derived by using the RAS method. The methodologies are picked and customized to meet the specific needs of a medium-scale embedded system. The process of the model development is depicted in Figure 1.

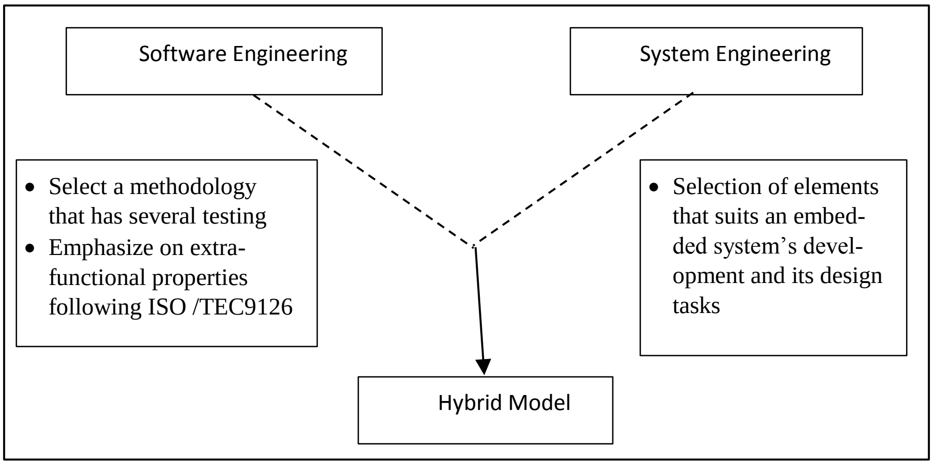
Software Engineering
System Engineering
• Select a methodology that has several testing
• Emphasize on extra-functional properties following ISO /TEC9126
• Selection of elements that suits an embedded system's development and its design tasks
Hybrid Model

Figure 1. The Hybrid Process of the Development Methodologies.

There is no universally accepted agreement on how many phases exist in a system life cycle or what those phases are called (Faulconbridge & Ryan, 2002). Thus, the steps and phases in the proposed hybrid model are named in such a way that suits a medium-scale embedded system. The proposed model is as shown in Figure 2. In this hybrid model, the three phases in system engineering are still kept due to the similarity with embedded system that involves hardware. It is also harmonized with some of the embedded system design tasks proposed in Wolf (2000) in order to maintain the concept of embedded system. Details on the model development can be referred to in the researchers' earlier papers (Suliman & Nazri (2014a), Suliman & Nazri (2014b)).

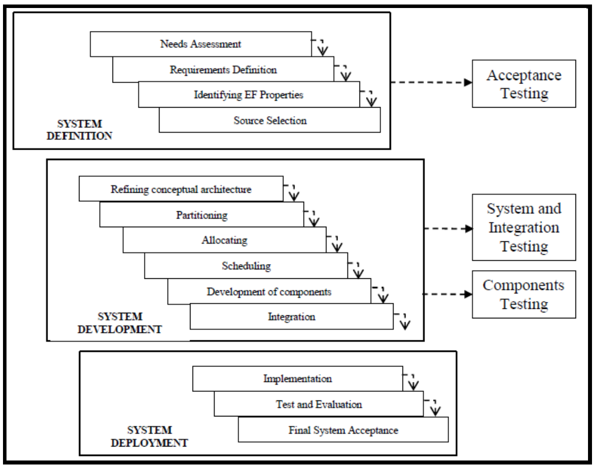

Figure 2. The Hybrid Model of Embedded System Development Methodologies.

THE FLOOD DETECTION SYSTEM

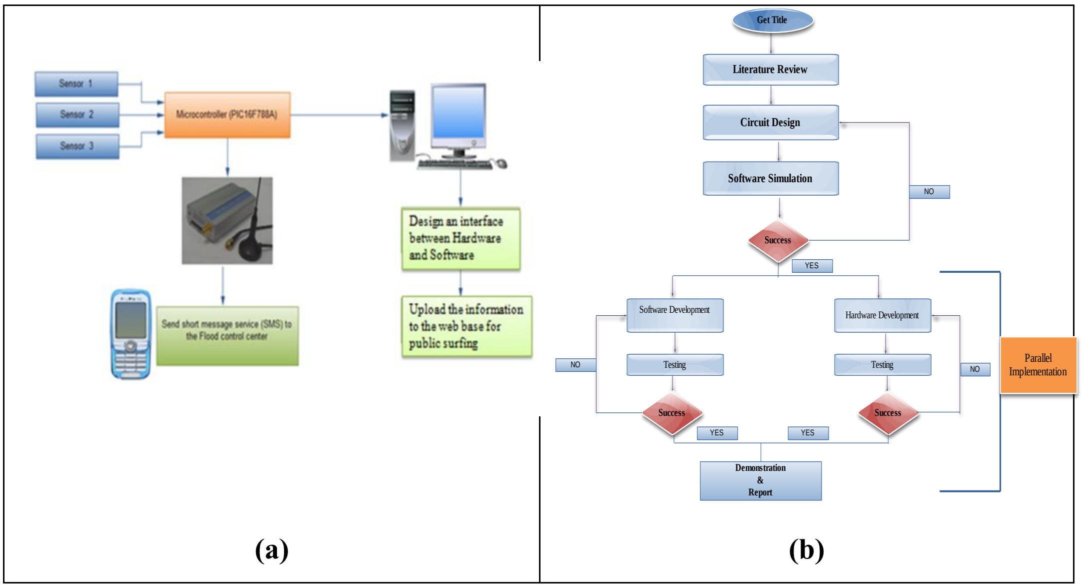

Figure 3. Flow of flood detection system (Ibrahim, 2011).

The flood detection system detects the water level by using water level sensors. When the water level reaches a stage that would cause flooding, notifications will be sent out to the flood control centre and the general public. The system development requires a combination of several hardware devices and software. The equipment includes water level sensors, a microcontroller, a PIC programmer, and GSM Modem. Software codes will be needed for the

PIC to interpret the signals as received from the water sensors and then to send the information via GSM as connected by a serial port to the PIC, to the flood control centre and registered users. Another set of software system will be needed at the control centre to receive the notifications and subsequently upload appropriate messages to the website for public consumption. The water level sensors identify three levels of water, i.e. precaution, warning, and danger.

As the flood detection system uses a combination of hardware and software in its development, it makes a perfect case study for the new hybrid model of embedded system methodology. The design flow of the system as developed previously, before using the new hybrid model, is shown in Figure 3. Figure 3(a) illustrates the general flow whereas Figure 3(b) shows the process flow of the system development using a flow chart. The flow as shown in Figure 3(b) is general and both software and hardware development are divided separately. Even before going into the development of software and hardware, the previous steps are general and do not consider embedded system properties and who will be the end user of the system. Each step in the proposed hybrid model will be mapped and explained as related to the flood detection system. The phases and steps that will be mentioned in the subsequent sections can be referred to from Figure 2.

Mapping of System Definition Phase

In the System Definition phase, there are four steps, namely Needs Assessment, Requirements Definition, Identifying Extra-functional Properties, and Source Selection.

Needs Assessment: As embedded system usually has a specific task, it must bw ensured that the developed system satisfies the initial needs. In this flood detection system, it is understood that the project needs to develop a tool to detect natural floods and give continuous alert information to the public.

Requirements Definition: After the needs are clearly identified, it is now crucial for the researchers to gather all the related information in detail. In this case of flood detection system, the researchers can meet with the representatives from Jabatan Pengairan dan Saliran. During the meeting, the researchers must ask all the related questions to the system. For example:

Where will the system be placed?

Who will monitor the system?

What kind of warning do they want?

What is the medium to broadcast information?

All of these questions are crucial so that the requirements of the system are clearly identified and to ensure that the system being developed will adhere to all the requirements needed by the client.

Identifying Extra-functional Properties: This step is crucial and must be put in the System Definition phase because embedded systems differ from one another. There must be a tradeoff on what extra-functional properties want to be made a priority because not all of them can be kept. The system will be developed according to the priority agreed in this step.

Source Selection: Sources for hardware must be selected carefully and must be compared to the embedded system’s needs.

MAPPING OF SYSTEM DEVELOPMENT PHASE

In the System Development phase, there are six steps, namely Refining Conceptual Architecture, Partitioning, Allocating, Scheduling, Development of Components, and Integration.

Refining conceptual architecture: After the needs are clearly identified from the previous phase, before the development begins, all the elements involved must be linked together to see the flow of the intended system. As for the flood detection system, the flow of data can be seen as presented below:

River Water Level → Sensor/Microcontroller → GSM Modem → Flood control system (receive information by cellphone) → Notify public by uploading information to a specific website

Partitioning: Before the coding starts, the researchers must identify the functions and partition them to be implemented into smaller, interacting pieces of components. For the flood detection system, the functions needed are as follows:

Functions for each water level sensor and sending signals to PIC

Function for PIC sending information to GSM

How the functions interact with each other

Functions for website notification

Allocating: Allocate the partitions to the microcontroller. Especially for embedded system, the memory size of the microcontroller is limited. Therefore, the partitions must be allocated according to the PIC memory size. In programming a PIC, it is unlike modular programming for software development where memory is rarely the issue. However for PIC, it involves stacks; context-switching that requires high speed processing. Therefore, a suffice amount of memory needs to be allocated for each running function.

Scheduling: Schedule the times at which function are executed. This is important when several functional partitions share one hardware (microcontroller) unit. Each of the water level sensors will be on standby mode if they do not sense any raise. If there is water raise, each water level sensor will run its function in order to send signals to the PIC when needed.

Development of Components: After the functions have been partitioned, allocated, and scheduled, now, it is time to develop the system by components. It will start by programming the PIC microcontroller and GSM modem, and developing an interface.

Integration: Integrating all functions into one whole system for interactions. Since the flood detection system involves another hardware, which is GSM modem, all of these components can be integrated, as well as with its system interface.

MAPPING OF SYSTEM DEPLOYMENT PHASE

In the System Deployment phase, there are three steps, namely Implementation, Test and Evaluation, and Final System Acceptance.

Implementation: Implement the system for a trial purpose. During this step, the system is brought to the real situation. The water level will be raised to test the system's response.

Test and evaluation: Testing for defects of hardware; be it the water level sensors, microcontroller or GSM modem, as well as any part of the software that is not functioning

Final System Acceptance: End users (if any) are satisfied with the system. In this case, the flood detection system is brought to the real situation and the system responds well to the rise in water level.

With the phases and steps in the new hybrid model for embedded systems development methodology carefully followed, the system can be built in a more systematic and structured manner.

DISCUSSION

The proposed hybrid model was developed by reviewing available methodologies in software engineering. The methodologies reviewed are Waterfall, V-model, Prototyping, and Spiral. Only selected common software development methodologies that suit embedded system requirements are reviewed because there should be a balance in developing the proposed hybrid model. If too much attention is given to all available software development methodologies, the research direction would neglect the hardware part. As for the hardware part, this paper reviewed systems engineering methodology because it is used for sophisticated embedded systems and high-end projects. A systems engineering methodology is developed using RAS (Recognize-Analyze-Synthesis). And this is the method followed when developing the hybrid model.

After the development methodologies have been selected, an analysis needs to be performed to see how the two chosen methodologies can be interleaved with each other. As for V-model, it is chosen because it emphasizes planning for verification and validation of the product in the early stages of product development. The development is planned in such a way that each deliverable must be testable. Embedded system is a system that requires high reliability. Thus, the high number of tests included makes V-model an excellent choice. However, not all testing are kept because the proposed model is meant for a medium-scale embedded system. Some testing are combined and some are removed and replaced with another kind of testing that suits a medium-scale embedded system. As for systems engineering, the steps are suitable for system engineering, which normally involves sophisticated and high-end projects.

The proposed hybrid model is inspired from the difficulty in developing an embedded system that does not have a solid development methodology to which system developers can adhere to. The research managed to propose a novel model in embedded system development methodology that hybrids software engineering and systems engineering and is verified by using flood detection system as a case study. With the phases and steps in the new hybrid model for embedded systems development methodology carefully followed, the system was built in a more systematic and structured manner addressing the every needs of an embedded system's requirements.

ACKNOWLEDGMENT

The authors wish to thank the Ministry of Education Malaysia for funding this study under Long Term Research Grant Scheme (LRGS/b-u/2012/UUM/Teknologi Komunikasi dan Informasi).

REFERENCES

Chung, L., Nixon, B.A., Yu, E., & Mylopoulos, J. (1999). "Non-functional requirements in software engineering". *The Kluwer International Series in Software Engineering*. Kluwer Academic Publishers Group, Dordrecht, Netherlands.

Crnkovic, I. (2005). "Component-based approach for embedded system design", in *International Conference of Software Engineering*, (ICSE'05), St. Louis, Missouri, USA, May.

Jerraya, A. A., & Wolf, W. (2005). *"Hardware/Software Interface Codesign for Embedded Systems,* TIMA Laboratory, Princeton University.

Ramesh, U. B. K. (2012). Master's thesis: "*A taxonomy for extra-functional properties of embedded system*", Malardalen University, Sweden.

Suliman, A., & Nazri, N. (2014a). Overview of software engineering and systems engineering development methodology for embedded system, *Knowledge Management International Conference 2014*, KMiCE2014, pp. 643–648.

Suliman A., & Nazri, N. (2014b). A new hybrid model of software engineering and systems engineering for embedded system development methodology, *6th International Conference in Multimedia and Information Technology*, ICIMU2014, 351–355.

Wolf, W. (2000). “Computers as Components – Principles of embedded computing system design”. *Morgan Kaufmann*.