Integrated MARTE-based Model for Designing Component-Based Embedded Real-Time Software

Mohd Z. M. Zaki¹, Dayang N. A. Jawawi² and Mohd Adham Isa³

¹,²,³Universiti Teknologi Malaysia, Johor, Malaysia
¹zulkiflizaki@utm.my, ²dayang@utm.my, ³mohdadham@utm.my

Abstract

Recently, modeling and implementation of Embedded Real Time System (ERTS) are unavoidably becoming more complicated to develop and be reused because of the increasingly complex design. The complexity is due to the functionality increment factor in accordance with users’ needs and demands, resulting in the growing scale of the developed systems. The current development approach based on Object-Oriented (OO) does not match the current requirements of the system. The OO approach has numerous flaws, thus, Component-Based Software Engineering (CBSE) has been appointed to resolve those problems. However, the current CBSE approach also has some drawbacks such as lack of ERTS standardized modeling and specific development methodology. The problems concerning the established ERTS development methodologies, Methods for Component-Based Real-Time Object-Oriented Development Testing (MARMOT) shows the capabilities to tackle ERTS modeling and implementation using CBSE strategy are partially provide the required standard modeling language for development tool adaptation. Thus, an integrated component model is proposed, by integrating MARMOT and a Unified Modeling Language (UML) profile known as Modeling and Analysis for Real-Time and Embedded (MARTE), which can be adapted to the CBSE approach standard modeling. In addition, the proposed model can improve the existing MARMOT software process. The findings showed that the proposed model has reduced ERTS development complexity by enhancing the design mechanisms.

Keywords: Component-Based Software Engineering, Embedded Real-Time Software, Component Model, Software Design, Software Reuse, UML

1. Introduction

Component-Based Software Engineering (CBSE) discipline has evolved and has covered most of the application domains such as medical, commercial, business and others, including Embedded Real-Time System (ERTS). CBSE is an example of software reuse methodologies that has become a popular approach to increase software productivity and improve its quality, thereby decreasing the costs of software development [1]. Because of this, CBSE is a promising approach for ERT system to overcome complexity by enabling reusability in higher granularity and encapsulation. CBSE offers an independent software structure through component interface connection architecture ability [2-4]. The reason for focusing CBSE in ERTS software development is mainly because CBSE would simplify the exchange of the software parts between ERTS developers. Some ERTS experts believe that the use of CBSE can help to improve ERTS into the higher levels.

Basically, ERTS is a combined concept of embedded and real-time aspects to emphasize the relationship between both entities. This is because most embedded systems are real-time in nature [5]. As a result, the embedded system is stimulated by external environment and hardware constraints within a specific timing period. Those exceptional characteristics have established ERTS as distinct system compared to other conventional
systems [3]. ERTS has two main categories, namely soft and hard. Despite the fact that these categories are categories based on different timing-based properties, both are restrained to embedded constraints. Other than timing requirements, embedded constraints like memory limitation also need considerations, especially during the system development [6].

Lately, ERTS has shown an enormous growth of complexity because of the functionality increment factor. This is due to user needs and demands [7-11]. Due to this factor, ERTS development is unavoidably becoming more complicated and sophisticated [12]. The complications involve modeling and implementation approaches such as procedural programming and manual coding process are no longer sufficient to satisfy the complex ERTS, which could cause design complexity, or carry a million or more lines of code at the programming level. These can cause produce software maintenance difficulty due to functionality expansion. Continuation with the conventional development approach can reduce the level of software reusability. Hence, the need for a more acceptable and practical approach is important to assist the development process rather than relying on simplex traditional software development concepts.

Currently, the use of models to reduce software complexity has been widely applied [13]. Model-based approach usage can aid the software development because of the ability to visualize using graphical notations, compared to conventional pseudo codes, algorithms and programming languages [13-14]. The use of graphical notations for software development can offer several advantages such as ease of software evolution, modifiability, construction, and the ability for early error detections, estimations and predictions. However, as the functionality increases, ERTS size becomes larger. Hence, the use of model-based approach is crucial for the success of software development [8]. Eventually, Object-Oriented (OO) methodology has been recommended to facilitate the complex ERTS. Although OO methodology able to provide several initiatives, there still remain disadvantages. In OO methodology, encapsulated objects and the architecture are not clearly separated, thus reducing the granularity of an independent software structure [2-4]. The growing complexity of such systems and software developments require a development method that would enable reuse of once developed parts and provide away for the enabling of those parts. Additionally, the development methods also need to provide more efficient development in a way that produces more reliable systems [15]. This can be resolved by using more adequate methodologies, especially those that can express and solve the complexity of the ERT system. One of the solutions is employing software reuse methodologies such as CBSE [10].

Although CBSE offers many advantages, it still cannot be fully adapted with ERTS. The main reason is the CBSE lack of ability to cope with the ERTS constraint [5, 16]. Implementing CBSE in ERTS is a challenging task compared to performance in normal systems and usually has hardly been done in research contexts and environments [4, 17]. At present, most of the developers have a tendency to adopt specific component technology to support CBSE in the software development. However, most of the component technology adaptation process depends on the support of specific tools [18]. This is because most of the component technologies are not constructed using any standard language or notation. The use of standardized notation can eliminate necessity for specific tools, in which case it can unlock the implementation towards general tools.

This paper aims to adapt UML profile for ERTS with CBSE development methodology for ERTS. This adaptation is done in order to support component modeling with higher degree of reusability and detailed specification of ERTS design. To realize this work, we have proposed an integrated component model and offer supports for ERTS modeling and implementation. We also discuss how the integrated component model can support ERTS modeling through undertaking of a case study. Furthermore, we validate our approach into one ERTS case study and have substantiated it with a comparative evaluation method.
The rest of the paper is organized as follows: Section 2 summarizes and compares related work of the existing approaches of CBSE methodologies for ERTS and UML Profile for ERTS. Section 3 explains the integrated component model and its implementation using selected case study. Section 4 evaluates our approach where the result is validated with a comparative evaluation and discussed in the section. Finally, we draw the conclusions in Section 5.

2. Related Works

This section will briefly introduce CBSE methodologies for ERTS and UML Profiles for ERTS. The main focus is on the mapping of UML profile for ERTS into CBSE methodology for ERTS. Through the mapping, both profile and methodology are integrated which can profitable for both parties. The benefits are conducted in different manner, whereby the profile is used to support detailed specification of ERTS and the methodology is used for support the component model basic structure.

2.1. CBSE Methodology for ERTS

There are several CBSE methodologies which have been developed for general purposes, for example, Rational Unified Process (RUP) [19-25]. These methodologies have been widely used in business-based and enterprise-based systems. Those general CBSE methodologies mostly correspond with the latest and standardized modeling language and are equipped with specific modeling tools. But, most of them are not suitable to be adapted with ERTS development due to shortcomings in addressing ERTS special requirements.

For that reason, several new CBSE methodologies have emerged specifically for ERTS development. Those methodologies such as ACCORD [26][27], MARMOT [28-29], COMDES [30-31] and ELCRA [32] have been introduced to allow CBSE development mechanisms to be used in ERTS development without neglecting its special constraints.

a. ACCORD is proposed to aid real-time application development. ACCORD users mainly focus on non-expert software developers such as engineers. This methodology is intended for providing abstract methods and tools. This is to ensure that non-expert real-time software developers can use the methodology for specifying and prototyping real-time systems. The development cycle on ACCORD methodology includes three phases, namely, Analysis, Prototype and Tester. Software components are produced at Analysis and Prototype phases. Analysis phase includes Preliminary Analysis Model (PAM) and Detailed Analysis Model (DAM). In the Analysis phase, the components are identified at DAM phase. ACCORD can only support for software elements. ACCORD methodology is also placed to provide an implementation tool, which can adapt concepts of UML profiling and modeling rules. UML 2.0 is used in mapping the action semantics syntax.

b. Method for Component-Based Real-Time Object-Oriented Development and Testing or MARMOT is a system development methodology for an ERTS. MARMOT aims to provide the ingredient to master the multi-disciplinary effort of developing ERTS. MARMOT methodology provides templates, models and guidelines for the products describing a system, and how these artifacts are built. MARMOT methodology includes four major activities, namely: Composition, Decomposition, Validation and Embodiment. Components are identified at the Design and Implementation phases. MARMOT methodology considers both software and hardware in the development activities. This methodology supported by UML 2.0 for modeling language mechanisms.

c. COMDES or Component-Based Design of Software for Distribution of Embedded Systems is a component-based software development framework, intended for efficient development of distributed embedded control systems with hard real-time requirements.
COMDES framework includes methodology, Model of Computation (MoC), modeling techniques, implementation and code synthesis. COMDES methodology attempts to establish an engineering methodology encompassing high-level modeling and analysis issues as well as low-level implementation and deployment techniques. Components are identified via MoC and a hierarchical structure. COMDES methodology is interested only in the software elements. COMDES has supported tools by workbench tools, which have several features, namely: graphical editor, constraint language, and formal verification and code generator.

d. The main purpose of Early Life-Cycle Reuse Approach methodology, known as ELCRA is to aid a systematic software reuse, particularly for an ERTS development. This methodology mostly focuses on the development of Autonomous Mobile Robot (AMR). In ELCRA methodology, ERTS specification is converted into implementation. This process uses recognized and optimal methods, which are common to the developers. ELCRA methodology is composed of three processes, namely: pattern-oriented analysis, component-based design and component-oriented programming framework. There are also two component repositories, namely: analysis-pattern component repository and design components repository. Component is identified at the early life cycle reuse stage by component-based analysis pattern adaptation. This adaptation can be used to defined real-time behavior of real-time components. ELCRA has focused attention to only software elements.

As a summary, MARMOT can provide capabilities to satisfy the ERT modeling criteria, using the MARTE profile with CBSE. Based on the review, it is concluded that the MARMOT method is useful in supporting multidisciplinary knowledge that involves software and hardware. In addition, MARMOT is a CBSE development methodology that fulfills all the criteria for model-related support, which can then be used to assist in the components for design phase and implementation. Furthermore, MARMOT uses UML 2.0 as modeling language. This can be a great advantage in order to ensure the use of MARMOT with MARTE.

As referred from MARTE profile, the component development mostly focuses on the MARTE design phase. So, from the evaluation, MARMOT has shown the capability to support component development during this phase. The most important criteria, reusability, can also be fulfilled with the use of MARMOT with MARTE, since MARMOT has provided the reuse support in the design phase. Although ELCRA also provides some mechanisms, which are almost similar to MARMOT, ELCRA does not focus on hardware components. Similarly, there is no support component at implementation and no support on the code template.

2.2. UML Profile for ERTS

Modeling ERTS system with suitable modeling language to reduce software development complexity is a crucial task. UML has become a strong candidate for designing and modeling ERT systems. This is mainly due to the abstraction ability provided by this language [33]. Introduced by OMG, UML is a graphical language for visualizing, specifying, constructing and documenting the artifacts of software and system development [34]. Lately, UML has served as a standard language of software blueprints and has been used widely for software and system developments.

Although UML has shown its ability to be used in modeling software and systems widely, it has still developed challenges and been challenged by some important problems. One of these problems is the ability to cope with ERTS unique requirements such as real-time constraints, embedded resource restrictions and component modeling [35, 10]. This is because the UML specification only specifies syntax and semantics of its notation, but does not determine how to apply its elements within a development process [35]. Nevertheless, UML has been built with the ability of extension mechanisms. This ability
can allow customization of language, enabling definition profiles for specific domains [8]. This extension is known as profile.

Recently, there are many profiles, which have been developed to extend UML for modeling ERTS. Examples of the profiles are UML for Real Time (UML-RT) [33], UML for Scheduling, Performance and Time (UML-SPT) [36] and UML for Modeling and Analysis for Real-Time and Embedded (MARTE) [37].

a. UML for Real Time or UML-RT has been developed by Rational Software as a means to capture the concepts of real-time profile, which have been defined in the Real-time Object Oriented Modeling (ROOM) Language [33]. UML-RT has become a popular modeling language not just designed for interpretation of a design model with allocation of information for quantitative analysis [38][39]. Modeling real-time systems in UML-RT consist of two types of model constructions, namely: structural model and behavioral model. Many industrial developers have selected UML-RT as a targeted modeling language due to the ability of UML-RT to provide component-oriented notation close to the standard of UML 2.0 [39]. UML-RT provided formality in order to handle active objects called capsules. These capsules can have hierarchical decomposition consisting of an internal model structure. This internal structure is composed of capsules, which enables complex real-time and embedded systems to be modeled in a structure model using UML-RT.

b. UML for Schedulability, Performance and Time, known as UML-SPT is a UML profile for real-time system modeling. OMG embraced UML-SPT in 2002 [36] to increase the interest in using OO modeling paradigms and UML to real-time systems. UML-SPT is a modeling language that provides a specific framework to tackle timing and extra-functionality properties of real-time system development. Besides, UML-SPT can also support ordinary UML models in predicting quantitative analysis, which includes schedulability analysis [39]. In fact, UML-SPT is also equipped with stereotypes and labeled values with the intention of interpreting the UML models.

c. OMG called for a newly introduced UML profile known as MARTE or UML for Modeling and Analysis for Real-Time and Embedded [37]. MARTE is the first standard UML designed for modeling ERTS after more than three years of work. This UML profile addresses the ability to model and analyze real-time systems and embedded domains. The MARTE specification is composed of three main packages. The first package views the foundation for real-time and embedded concepts domain, the second package defines the model-based design and the third package defines the model-based analysis. MARTE has provided some new key features, such as support for non-functional property modeling and adds rich time and resource models to UML. Moreover, this profile has also defined of concepts for multi-platform modeling and allocation. Additionally, MARTE has been used for quantitative analysis such as scheduling and non-functional quality analysis [40, 41].

Although UML-RT can allow the construction of a complex event-driven or distributed system, UML-RT is still not suitable for supporting and modeling of timing constraints [38, 39]. This is because it mainly focuses on the concepts of active components, which are the capsule. UML-RT can address real-time issues, as it represents as an extension of UML, but not directly in modeling language. UML-SPT is one of the ERTS profiles created to encounter ERTS modeling problems. However, there are several limitations that have been revealed due to current implementations of the profile to model ERTS real scenario, including lack of behavior annotation and lack of support for CBSE approach. Intending to replaces UML-SPT, OMG called for a newer profile known as MARTE. MARTE has several enhancements in addressing ERT system and software development, especially for supporting non-functional property, time and resource annotations, and
enabling CBSE approach. MARTE has been upgraded to adapt CBSE in software development with the introduction of component modeling.

Although all new features seem to be very convenient for developing an ERTS, MARTE still has several shortages of features, particularly for the design and implementation process. It has been discovered that currently MARTE has no support on a code generation template [8]. Acting as a profile for a modeling language, MARTE has no specific systematic development methodology. However, the use of UML to model components can reduce component reusability, as UML is a modeling language, comprising with many diagrams, which are related to each other. Therefore, there is a requirement for mechanisms needing to be adapted with UML, especially MARTE that can address component modeling, having higher reusability without eliminating other related or bounded diagrams.

There are other types of modeling language profiles for ERTS available, but these profiles are still not able to entirely fulfill requirements to perform CBSE activities adequately [35]. Nevertheless, the modeling of ERTS using this modeling language is closely related to the selection of suitable CBSE methodology. Selection of modeling languages can support the modeling process [6]. Among CBSE development methodologies for ERTS, several weaknesses can be noted. Examples of these are not corresponding with recent modeling language and non-compliance of component modeling with standard modeling language. Based on these weaknesses, most of the methodologies require a more acceptable modeling language, which can emphasize ERTS features and CBSE attributes precisely.

3. The Integration Methodology

Integration between MARTE profile and MARMOT methodology is possible because both of them, although not in the same paradigm, still depend on UML as the core modeling language. This modeling language is the parent of MARTE profile, and has been used as the main modeling language in MARMOT methodology. To assist with the integration process, an integration methodology, as illustrated in Figure 1, has been introduced.

This integration methodology, composed with steps and elements, is introduced to aid the integration process so it can be executed in a proper manner due to the visibility of the process. The visibility of the process assists in checking the correctness of the entire integration flow. There are five main steps and two main elements, which have been recognized. These steps are, namely: Gather, Identify, Map, Integrate and Validate, and the related elements are Profile and Methodology. During the integration, most of the process focused on the relationship of MARTE profile and MARMOT methodology elements and artifacts. This is because their relationship obviously depends on the elements and artifacts in both profile and methodology.
The first step is to understand MARTE profile. This is important to investigate integration of MARTE profile elements with MARMOT methodology elements according to the newly proposed methodology. During this step, investigation and identification on MARTE elements (mostly relating to a model-based and CBSE approach) are conducted. Hence, the second step understands MARMOT methodology and the elements and artifacts involved along the process. During this step, identification on MARMOT elements, especially on structure and behavior modeling, will be conducted and investigated.

The third step is mapping of MARTE profile elements and MARMOT methodology elements. The mapping is carried out with the consideration of the relationship between both profile and methodology. Mapping the related features of both elements and features of MARTE and MARMOT performs this process. The integration process continues with the fourth step, which is integrating both MARTE and MARMOT. The integration process will be assessed based on the elements, which have been identified during the first and second steps, and the result obtained from the mapping process in the third steps. Finally, the fifth step applies the integrated development approach into a case study. In this step, the applicability of the integrated development approach will be verified using the case study by modeling the structure and behavior diagram. This can be verified by the newly enhancement which has been adopted during the integration.

4. The Integration Result

The integration result is shown step by step according to the integration methodology discussed in the previous section.

4.1. Information Gathering and Elements Identifying

An information gathering process is done by collecting information regarding MARTE profile and MARMOT methodology. According to the findings and information from the previous step, the similar feature in both profile and methodology is the component model. This is due to the availability of component models in both profile and methodology. Model integration needs to occur at the same level or paradigm of model. Therefore, this availability of component models is important to ensure that the integration can be carried out. MARTE profile has introduced MARTE General Component Model (MARTE GCM) and MARMOT has promoted its own MARMOT Component Model (MARMOT CM).

These elements are actually those of component model structures, which are already being applied in MARMOT. The mapping subsequently continues to find similarities.
between these elements with MARTE GCM elements. Using the similarities, the meta-model integration of both component models can be done. Furthermore, the investigation on MARMOT CM can assist in improving the current component model that has been implemented by MARMOT to a new enhancement. MARTE GCM will provide the enhancement elements.

a. Mapping of Component Models Elements

This is necessary to ensure that the integration process is performed correctly, based on the correct integration point. The integration point is the similar element that exists in both profile and methodology. Therefore, during the integration process, most of the process mostly relied on the relationship between MARTE GCM and MARMOT CM. Thereby; this integration process can focus on the relationships of CBSE methodology that can support the new UML profile for ERTS. In addition, investigation on MARTE GCM can also help to determine the relationship of a formal CBSE component model with MARTE GCM. The result of relationships mapping MARTE GCM to MARMOT CM elements is shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Component Models Elements Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Structure</td>
</tr>
<tr>
<td>Behavior</td>
</tr>
<tr>
<td>Interface</td>
</tr>
</tbody>
</table>

These elements are actually those of component model structures, which are already being applied in MARMOT. The mapping subsequently continues to find similarities between these elements with MARTE GCM elements. Using the similarities, the meta-model integration of both component models can be done. Furthermore, the investigation on MARMOT CM can assist in improving the current component model that has been implemented by MARMOT to a new enhancement. MARTE GCM will provide the enhancement elements.

The results from the mapping are subsequently used to derive the meta-model. The meta-model is not a new development. This meta-model is an enhanced meta-model for a component model. The current component model is enhanced with additional; elements from other component model elements. Therefore, using elements identified through the
mapping, the integrated component model is developed. Similar elements, identified from the mapping, are used as integration points. The integration point is the point at which the integration occurs. The integration point is selected based on the most similar element that can be used as a basis for the integration.

**b. Integrating the Component Models**

The integration of two component models is represented in Figure 2. The integration is based on the component models’ meta-model integration. By using MARMOT component structure as the integration point, integration on both of the component models is enabled. This occurs because by providing methodology to MARTE, the integration will be based mostly on MARMOT method. It will then be enhanced using new elements from MARTE that have been introduced to support ERTS development, mainly in the designing and modeling phase.

![Figure 2. MARTE GCM and MARMOT CM Integration](image)

During the integration process, there are no elements of meta-models, either from MARTE GCM or MARMOT CM as they have been removed from the original meta-model structures. Instead, the integration of the meta-models has given more consideration to the similarity of the elements, which have been recognized based on the mapping process. Therefore, the elements in both MARTE GCCM and MARMOT CM will retain the same relationships. Based on the integration, most of the enhancements involve the capabilities of expressing the constraint requirements of ERTS directly into modeling. MARTE GCM has provides new elements to MARMOT CM, namely: Connector, AssemblyPart, ConnectorEnd, InteractionPort, InvocationPort, RtAction, FlowPort, FlowProperty, ClientServerPort, ClientServerFeature and RealTimeFeature. This includes the introduction of the new port mechanisms in structure and temporal properties mechanisms in behavior. The application of the integrated component model will be used during the development phases. Based on the integration meta-model, the current component model that has been adapted in MARMOT methodology is enhanced. The enhancement will be supported by new elements that have been proposed in MARTE to support design and modeling for ERTS development.
c. Applying the Integrated Component Model

An implementation on a case study of an ERTS named Robotic Wheelchair System (RWS) case study [42] was used to demonstrate the application of the integration. This application has been defined in the context of an academic challenge, the goal of which is to provide a case study for benchmarking different kinds of design approaches to ERTS. This case study was chosen because it is fairly representative of the problems encountered by ERTS developers when designing such systems. These problems mainly relate to ERTS constraints and CBSE development approach. The case study is also suitable for showing adaptations (either software or hardware component), as RWS is an ERTS test bed, which has been integrated with various components. This is an advantage for using RWS because it can emphasize the use of ERTS modeling and CBSE development methodology in the development. The integrated component model is applied and showed in Fig. 3 using an example of Infra Red (IR) sensor component.

Figure 3. Example of RWS IR Sensor Component

As illustrated in Figure 3, data based communication can be enabled using MARTE ports. The type of this value may be given directly as the port type. Alternatively, if the data needs to be sent and received through the same port, MARTE has introduced flow specifications, which contain only properties. Instead, with the new enhancement using the integration meta-models, the modeling component in MARMOT has become more detailed and specific with a new notation introduced by MARTE, specifically to support CBSE development and design in ERTS. This detailing can help developers in terms of increasing understanding of ERTS design. Figure 4 shows the new elements existing in the newly integrated component model.
Figure 4. Elements of the Proposed Integrated Component Model

The component models are implemented based on the proposed integrated component model. The external component structure shows the composition of component model, including the component itself and the usage of ports, signals and interfaces. Figure 5 shows the example of IR Sensor component. The component is demonstrated in iconic form. In iconic form, additional notation elements such as texts and comments of components, ports and interfaces properties are excluded. Later, the component model has been detailed in the form of component and its related elements, as shown in Figure 6.

Figure 5. IRSensor Component (Iconic Form)

Figure 5 illustrates external structure of IR Sensor component, named as IRSensor. The IRSensor component is showed in iconic form. The iconic form is a form, which simplifies the component model by excluding additional information such as component specifications, relationships, ports and interfaces descriptions and internal properties. All these elements are clarified inside the internal structure of the component model. IRSensor component is a passive component. It has three ports, an inPort and two outPort. The inPort, IRSensorDriver port, is annotated with ◦ notation and the outPort, condition ports are annotated with □ notation. Those ports properties are specified using notes notation and are elaborated in the internal component structural diagram.

Later, the component is refined into detailed specifications, as shown in Figure 6. The detail specifications are illustrated using a class diagram. In the class diagram, the component is decomposed into smaller parts for the realization process. The realization
process shows the related relationship between classes inside a component, which is important to ensure the self-governing and independent ability.

IRSensor component is a <<ppUnit>> stereotype. A <<ppUnit>> stereotype is used to indicate a passive component. A passive component is defined based on the component properties that do not require task synchronization. IRSensor component required two internal processes. Those processes are initialization and execution. The processes are implemented into the component using functions, which are INITIRSensor function and EXECIRSensor function. The functions are implemented based on Component-Oriented Programming (COP) Framework [42]. In COP Framework, a passive component required to have two internal processes, initialization process and execution process. The processes are meant to ensure the component can functional accordingly as an individual unit.

The IRSensor component has two types of ports, namely: <<clientServerPort>> and <<flowPort>>. Both of these ports serve different purposes. A <<clientServerPort>> is an interface-driven communication port. This port communicates with other components or library through function-type interface specification. The interface is specified using <<clientServerFeature>> element. The <<clientServerFeature>> element specifies the detailed specification of the port such as type of specification, type of interface and relationships. In this case, the <<clientServerPort>> port is used to connect the component with its driver, which is located in Hardware Abstraction Layer (HAL) library. The component is connected to the driver by calling respective functions, which is attached to the microcontroller’s library. In other words, this kind of port can avoid unwanted reconfiguration, as the microcontroller’s library is in peer-to-peer connection with the hardware. If the configuration is reset to different settings, so the hardware possibly is not connected to the library. On other hand, <<flowPort>> port is a data-driven communication port. This port is used to send data from a component to another component directly. The data is sent using data-type interface. This port allows the data being shared by the components, but connected to each other. This mechanism is important because components could reconfigure themselves accordingly, based on the tasks and periods.

\textit{d. Validating Integrated Component Model}

Validation of the integrated MARTE GCM and MARMOT CM is possible. This is because both of them, although not in the same paradigm, still depend on UML as the core modeling language of both profile and methodology. This modeling language is the parent of MARTE profile, while has been used as the main modeling language in
MARMOT methodology. Based on this reason, the process can rely on the meta-model to validate the integration.

To aid in the validation process, the validation technique uses the Meta-Model Matching Experiment [43] to measure the quality of matching results. This work focuses on the validation of integration meta-model with respect to its quality matching and relevancies, as well as numbers of elements after the integration. Hence, in order to measure these features, a matching meta-model measurement technique is being reused.

This technique is originated in the field of the information retrieval. But to avoid faults and miscalculations, the meaning of each term used before have been redefined using the original specification [44, 45]. This is to ensure that each term used in the measurement satisfies the related elements in the integrated component model. This is because the originality of the technique is being used in the information retrieval domain; therefore there is a need to redefine to ensure the applicability of the measurement to be used in other domains as well.

The validation has uses a measurement technique to compare the numbers of MARMOT CM elements, $A_{MARMOT}$ and MARTE GCM elements, $B_{MARTE}$ respectively. In this technique, Precision and Recall are used as the primary measurement parameters. Precision has a mutual relationship with Recall, in which one thing affects or depends upon another. This technique subsequently uses a primary measurement parameter known as $AB_{MEASURE}$. The measures are based on the notion of true positive ($tp$), true negative ($tn$), false positive ($fp$), and false negative ($fn$). The definition of $tp$ is a number of elements that overlap between MARMOT CM and MARTE GCM elements, ($A_{MARMOT} \cap B_{MARTE}$), while $fp$ is a number of elements of MARMOT CM and MARTE GCM elements, which are not overlapped with each other. This is known as false matches ($fp = A_{MARMOT} \cap B_{MARTE}$) where $B_{MARTE} = (\mid tn \mid + \mid fp \mid)$. In additions, $fn$ defines the number of MARTE GCM elements that do not overlap with MARMOT CM elements. This is known as missed matches ($fp = B_{MARTE} \cap A_{MARMOT}$) where $A_{MARMOT} = (\mid fn \mid + \mid tn \mid)$. In this situation, true negative ($tn$) represents the elements, which are not represented in both MARMOT CM and MARTE GCM.

\[
\text{Precision} = \frac{\mid tn \mid}{\mid A_{MARMOT} \mid} = \frac{\mid tp \mid}{\mid tp \mid + \mid fp \mid}
\]

From the above definitions, equations of Precision and Recall are constructed. Precision is used to evaluate the relevant matching of MARTE GCM elements into MARMOT CM elements. Referring to Precision equation, if the Precision value is higher, then it can be conclude that the matches are found. On the other hand, if the number of $fp$ is equal to zero, then all the matches are considered correct. Recall is used to evaluate the frequency of relevant matches of $B_{MARTE}$. Based on the Recall equation below, if the recall value is higher, then it is states that nearly most relevant matches have been found.

\[
\text{Recall} = \frac{\mid tp \mid}{\mid B_{MARTE} \mid} = \frac{\mid tp \mid}{\mid tp \mid + \mid fn \mid}
\]

To measure $AB_{MEASURE}$, below equation is used. The equation for calculating $AB_{MEASURE}$ uses both values from Precision and Recall to encounter any problems of misestimating of measurement. Therefore, there is still a need for an equal weight average of Precision and Recall for formal measurement.
Therefore, in order to evaluate the relevancies number of the integrated component model, a basic mathematical logic based on a set theory has been used. The basic mathematical theory is derived based on relevant rules and guidance [46-47]. The mathematical logic equations are modeled as follows:

\[
AB_{\text{MEASURE}} = 2 * \frac{|tp|}{(|fn| + |tp|) + (|tp| + |fp|)}^{\frac{1}{2}}
\]

\[
= 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

The detailed of the validation has been done in [48]. From the validation calculation, the proposed integrated component model comprises a balanced average result value. Based on the mathematical logic equations, the numbers of integrated model elements are equivalent to the number of MARMOT CM union with MARTE GCM and the summation of the element types. We can conclude that our proposed integrated component model has been mapped correctly.

5. Comparative Evaluation

This section investigates the improvement of the proposed approach as compared with the existing works. The investigation is based on a comparative evaluation with the related works as mentioned in Section 2. To fairly compare, we choose the almost similar methodologies with our approach as shown in Table 2. The selected methodologies are equally compared with the selected comparison criteria regarding UML-based component model related to the ERTS requirements.

The comparison criteria are divided into two aspects, namely Modelling and Implementation. Modelling aspect is reflecting to the capability of a component model on the methodology for supporting notation (A1), detailed resource (A2) or software and hardware modelling within the grasp of ERTS modelling, and using standardizes (A3) modelling language. On the other hand, implementation aspect is concerning the influence of the CBSE concept of reusability (B1), applicability (B2) and adaptability (B3) onto the results of ERTS development.

The comparative evaluation in Table 2 summarizes the improvement of our approach against the selected methodologies. For modelling aspect, a yes and no are used to mark the results as they fulfil the criteria. A yes or no mark in parenthesis means that a methodology is partially fulfils the criteria. As for implementation aspect, a low, medium and high are used to rank the approaches. A low mark means a methodology is either have or not have the ability to fulfil the criteria. A medium mark means a methodology has the required ability but in a limited condition. A high mark means a methodology has either the highest or fully fulfils the criteria.
Based on Table 2, for modelling aspect, ERTS notation are fully supported by the proposed approach and ELCRA, but are neglected by ACCORD and COMDES. However, works in MARMOT is partially support but not completely and explicitly annotate based on ERTS requirements. On the other hand, only MARMOT and our proposed approach support on modelling component for both software and hardware, known as detailed resource modelling but not included in ACCORD, COMDES and ELCRA, where only focus on software components. The result shows that MARMOT and our approach are using standardize modelling language for the component modelling, but partially included in ELCRA, where the component model is using non-UML modelling. However, ACCORD and COMDES are not using standardize modelling language but have been customized towards their needs.

As for implementation aspect, only ELCRA and our proposed approach have high reusability because both are design-level component, which has higher reusability values. In contrast, COMDES and MARMOT have medium reusability capability, but ACCORD reusability capability is low. Likewise, ELCRA, MARMOT and our proposed approach show high applicability because easy to understand and implement. However, this cannot be done using either ACCORD or COMDES, as the methodologies are complex for implementation. Lastly, our proposed approach the ability to adapt with many tools as our approach is compromises with standard modelling language and CBSE concept. ACCORD has medium adaptability value because the availability of its own UML-based tool. But, as for COMDES, ELCRA and MARMOT, the adaptability should be improved by proposing tool to support the implementation process.

From the comparative evaluation result in Table 2, we can conclude that our approach, the integrated component model highly supports both modelling and implementation aspects of ERTS design.

6. Conclusions

In this paper, we discussed the adaptation of UML profile for ERTS into CBSE methodologies for ERTS. Complexity is a challenge that has been identified in ERTS system development. In this paper, complexity is refined in both modelling and implementation. There has been lack of standardized ERTS modelling in current CBSE development methodology and lack of CBSE software development methodology. It is possible to reduce this complexity by promoting the use of CBSE for ERTS development, with the adaptation of ERTS modelling language. Therefore, an integrated model of profile and methodology has been proposed in order to solve the problem by improving the component modelling in both modelling and implementation.

Guided by this motivation, we have proposed an integrated component model to help reduce model complexity. The aim of the developed integrated component model is to outline strategies for adapting ERTS profile and CBSE methodology in ERTS design. This integrated component model has been validated using Meta-Model Matching Experiment technique to check the correctness. From the validation, the result shows that our proposed integrated component model is integrated in proper manner and acceptable
value. In addition, we also have done a comparative evaluation with other approaches as well. The comparative evaluation has been performed using the selected criteria is the aspect of modelling and implementation. The result shows that our proposed integrated component model highly supports both modelling and implementation aspects of ERTS design.

Acknowledgements

Special thanks to Embedded and Real-Time Software Engineering Lab (EReTSEL) and Software Engineering Research Group (SERG), K-Economy Research Alliance (RAKE) for the knowledge supports, sharing and motivations, Malaysia Ministry of Education for the financial and funds support, and Faculty of Computing, Universiti Teknologi Malaysia for the facilities and infrastructures.

References


Authors

Mohd Zulkifli Mohd Zaki received the B.Sc and MSc in Computer Science from University Teknologi Malaysia in 2008 and 2012 respectively. His current research interests include software quality analysis, software modeling and component-based software engineering.

Dayang Norhayati Abang Jawawi received the Ph.D in Computer Science from University Teknologi Malaysia in 2006. She is now a senior lecturer at Software Engineering Department at University Teknologi Malaysia. Her research interest includes real-time system, component-based system, software quality and software modeling.

Mohd Adham Isa received the M.Sc and Ph.D in Computer Science from University Teknologi Malaysia in 2009 and 2013 respectively. His current research interests are software quality analysis, software modeling and software measurement.