Research Journal of Applied Sciences, Engineering and Technology 6(15): 2777-2783, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 28, 2012 Accepted: February 18, 2013 Published: August 20, 2013

Towards Formalizing Time-Constrained System Models in TCOZ

[1, 2]Kong Xiangying, [2]Chen yanhui and [1]Zhuang Yi
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2]Jiangsu Automation Research Institute, Lianyungang, China

Abstract: This study is concerned with using Unified Modeling Language (UML) model and the UML profile for Modeling and Analysis of Real-Time and Embedded (MARTE) systems to depict a Time-Constrained System (TCS). To perform system formal analysis, TCS models are transformed to Timed Communicating Object Z (TCOZ) specifications. The mapping rules for the modeling elements especially time-related issues are given at meta-model level. Static model, dynamic model and MARTE stereotypes in TCS models can be transformed to TCOZ models respectively. These models are composed together to complete specifications according to some rules. Hence the dependability of the TCS can be checked through the reasoning mechanism in TCOZ. A case study shows the applicability and feasibility of our approach.

Keywords: Mapping rules, MARTE, TCOZ, TCS, UML

INTRODUCTION

Time-Constrained System (TCS) is a special real time and embedded system which has strict demand in timeliness and it is generally used in the military, medical and communicational industry. It will lead to a great disaster once the timeliness is dissatisfied. Enhancing the dependability of TCS has becoming a hotspot in the scope of software development.

Unified Modeling Language (UML) is OMG (2009) most-used specification, which is widely used in object-oriented modeling. But UML lacks the ability to describe quantitative nonfunctional properties, such as real-time, dependability and resource. These properties are important in trustworthy real time and embedded systems. Therefore OMG (2011) proposed UML profile for Modeling and Analysis of Real Time and Embedded (MARTE) systems.

UML/MARTE depicts a system in graphics and lacks for secure semantic information. To verify and analyze system models, contributors have done much to transform UML diagrams to formal specification. Mostafa *et al*. (2007) formalized UML meta models using Z specification, but he didn't give the reuse mechanism to support automatic model transformation. Zhang and Jouault (2009) defined the mapping rules between MARTE and FIACRE and performed the heterogeneous model transformation under the AMMA platform. This method separates syntax and semantics transformation, but it needs to construct complete meta model for the formal language. Sergiu-Mihai (2001) gave the detailed transformation rules between TCS model and Z++ and he took the time constraint into account, but most of the real time and embedded systems issues such as concurrency, delay et al. were not covered. Mekki and Ghazel (2010) developed an algorithm for transforming UML to timed automata to verify the TCS model and he just handled some time annotations used for expressing state invariant and transition guards. Wei and Wang (2011), Mirco and Gilmore (2008), Ermeson *et al*. (2008) and Nianhua (2010) proposed their model transformation approaches with restricted applicability.

In this study, we use Timed Communicating Object Z (TCOZ) proposed by Brendan and Dong (2000) as an analysis tool. Our aim is to introduce a TCS model verification approach which is easy to perform, convenient for reuse and non functional property centered.

TCOZ OVERVIEW

TCOZ is a combination of two formal languages, i.e., Object-Z proposed by Smith (2000) and Timed CSP proposed by Reed and Roscoe (1986). Object-Z is fit for modeling data and algorithm with a semantics of single-thread. Timed CSP is an extension of Hoare's Communicating Sequential Processes (CSP) algebra notation and fit for modeling real-time and concurrent behavior. TCOZ perfectly combines the merit of the two languages.

TCOZ adopts the basic structure of Object-Z with an extending concept of channel in the declaration of class and operation. The mainly construction of the

Corresponding Author: Kong Xiangying, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

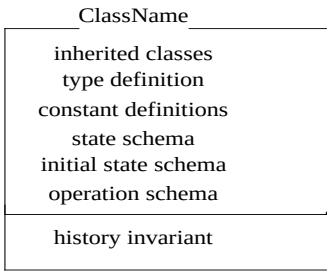

Fig. 1: Class definition in Object-Z

Object-Z is the class definition shown in Fig. 1. The detailed semantics can be seen in Smith (2000).

TCOZ regards the operation scheme as the terminable CSP process which changes the state and regards the class as the non terminating CSP process.

The process STOP is the deadlocked process which does nothing; SKIP, on the other hand, is a special process that does nothing but terminates successfully; WAIT t represents the process terminates after time t during which it does nothing; a→P represents the execution sequence of event a and process P; The sequential composition P; Q is a process which first behaves as P but continues by behaving as Q immediately after P terminates successfully; The external choice operator P□Q allows a choice of behavior according to what events are requested by its environment. P|||Q represents the asynchronous parallel of P and Q.? and ! represent the input and output communication between the process. a@t records the time at which the event occurs and allows the subsequent behavior. An important concept in CSP is the notion of channel. A channel is a collection of events of the form c.n; prefix c is the channel name and collection of subfixes the allowed values of the channel. μP• [expression]→P represents the recursion process. Active objects have their own thread of control and have the MAIN identifier in the Timed CSP part. Passive objects are controlled by other objects and without an MAIN process.

MAPPING RULES

This section describes how to transform UML model annotated with MARTE to TCOZ.

Static mapping: Static view shows the characteristics of the objects and relationships between them.

- **MARTE data type mapping:** Most of data types used in MARTE are defined in MARTE model library. The primitive types Integer, Boolean, Unlimited Natural, Real are corresponding to Z, B, N, All Numbers in TCOZ respectively. Types in common use and real-time related types mapping are given in Table 1.
- **Real time modeling element mapping:** Class stereotyped by RtUnit can be transformed to active object and PpUnit to passive object. Operations stereotyped by RtAction, RtBehavior, RtFeature and RtService can be transformed to operation scheme. Attribute can be transformed to TCOZ

Table 1: Types Mapping

MARTE date type	TCOZ date type	Remark
String	seq char	There is no string type in TCOZ, it can be seen as the sequence of char
T[ ]	seq T	There is no array type, it can be seen as the sequence of the elements
Enumeration	Free type	TCOZ has its own enumeration mechanism
User-define	Declaration	User-define type must be declaration in the front of the specication
NFP_Duration	ℤ	The three types are real-time related type,
NFP_Date time	ℤ	There are no direct mapping, semantically
NFP_Energy	ℤ	They are mapped to ℤ

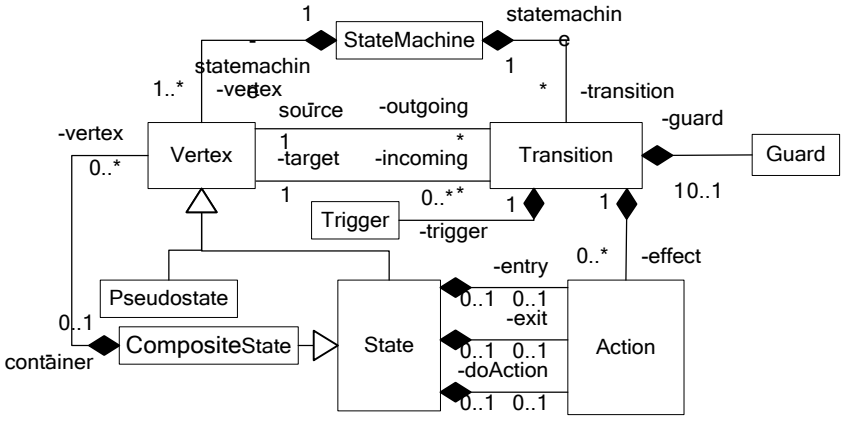

Fig. 2: UML State chart meta model

attribute which is defined in the state scheme. Association can be seen as a special attribute. Generalization can be depicted directly in the inherited classes.

Dynamic mapping: Statechart diagram models the life cycle of an object to specify the dynamic behavior changing with time. The Statechart model element is shown in Fig. 2. State is a condition during the life of an object or an interaction during which it satisfies some condition. The condition may performs some action or wait for some events to occur. For general state, it is declared in the state scheme and can be transformed into the pre-condition or post-condition in the operation scheme corresponding to transition. Composite state includes sub states which are parallel or sequential and it can be divided into simple state. Initial state in composite state can be ignored semantically. Pseudostate has the form of state but different actions with the usual state vertex. We just take the initial state into account. Semantically initial state is translated to the initial state scheme. Action can be entry/exit/do action for a state and it can be transformed to operation scheme according to its semantic information.

Transition represents the change of an object from sourceState to effectState and specifies the reaction trace of an object to event. The transition is triggered by an event. Then it may perform specific actions provided that transition guard is satisfied. The general format of a transition is shown as follows:

Event-name [guard-condition] /action expression: Event is the specification of a kind of change of state that may happen in the modeled system and it is divided into signal event, call event, change event and timed event. Semantically all of them can be seen as the satisfaction of some condition. We define a state attribute for each trigger in the TCOZ state scheme. We map them as precondition in the operation scheme corresponding to the transition, Except for timed event. MARTE extends the UML timed event meta class. It will be elaborated below.

Guard is a Boolean expression that transition will go if it is true. So it can be translated to the pre-condition in the predicate part. Action performed by the transition can be transformed into the operation scheme. The timed sequence of the actions can be presented by using the operator (;). In the concept of UML, the execution time of an action can be ignored compared with the external environmental event.
Transition can be defined as follows in TCOZ:

Transition≙[guard∧trigger_condition]•actionSequence

In which actionSequence is the time sequence of the operation scheme corresponding to actions.

A complex transition comes from or goes to a series of parallel sub states. Join represents one source state and several effect states, while Fork represents several source states and one effect states. Synchronization can be regarded as a combination of join and fork.

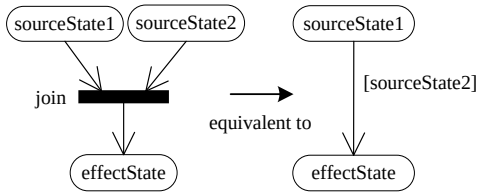

Fig. 3: Join and its equivalent transition

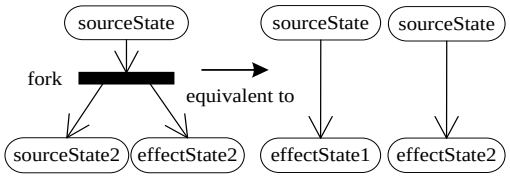

Fig. 4: fork and its equivalent transition

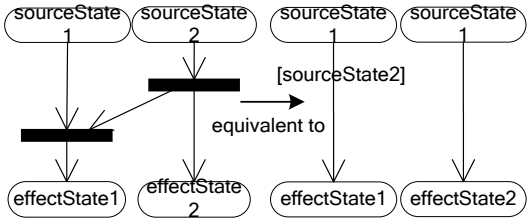

Fig. 5: Synchronization and its equivalent transition

Join is equivalent to a simple transition where sourceState2 is regarded as an extra guard condition of the transition as shown in Fig. 3.

Fork is equivalent to two independent simple transitions as shown in Fig. 4.

Synchronization is equivalent to two independent simple transitions out of which sourceState2 is regarded as the guard of another transition as shown in Fig. 5.

MARTE stereotype mapping: We are concerned about the time and resource non-functional property. ResourceUsage represents the run-time mechanism that effectively requires the usage of the resource. Its structure in MARTE is given in Fig. 6. UsageTypeAmount uses the Tagged value “execTime” to record the elapsed time and “energy” to record the use amount of the energy.

ResourceUsage has the state change of time and energy. It can be transformed into operation scheme semantically in which the type Z is transformed from MARTE type according to the rules defined above. Scheme of ResourceUsage is given in Fig. 7.
In TCOZ there is the type T defined:

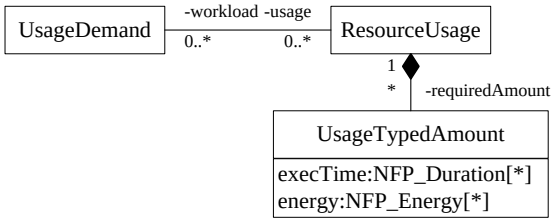

Fig. 6: Stereotype resource usage

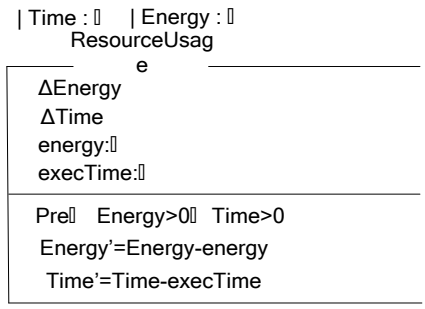

Fig. 7: Resource usage scheme

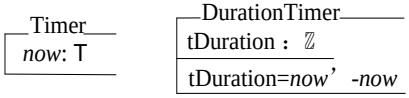

Fig. 8: Scheme of timer and duration time

$$T = R \odot T$$

where,
R = The real numbers and
T = The SI symbol for dimensions of time

TCOZ introduces a global real-time clock represented by a distinguished state attribute now. In order to represent the time-related information, we define the operation scheme Timer and DurationTimer to specify current time and duration time respectively. Scheme of Timer and Duration Timer are given in Fig. 8.

A transition stereotyped by ResourceUsage can be defined as the junction of the transition, ResourceUsage and DurationTimer in TCOZ:

TransitionResourceUsage≙Transition∧ResourceUsage∧DurationTimer

MARTE time model has defined a set of time-related concepts and semantics which can be simplified as shown in Fig. 9.

In MARTE a clock is considered as a means to access to time, TimedElement in a model which refers to a clock can be approximately redefined using the type T in TCOZ. In TCOZ all timing information is represented as real valued measurements in seconds.

TimedInstantObservation denotes an instant in time when the observed object arrive at a state, A TimedDurationObservation denotes some interval of time between one state and another:

TimedInstantObservation≙[t:T]•transition∧Timer∧[t = *now*] when it observe the sourceState

TimedInstantObservation≙[t:T]•transition∧Timer∧[t = *now'*] when it observe the effectState

TimedDurationObservation≙[t1:T;t2:T]•transition∧Timer∧[t1= now]∧[t2 = now']

A TimedConstraint is a constraint imposed on the occurrence of an event (TimedInstantConstraint), or on the duration of some execution, or even on the temporal distance between two events (TimedDuration Constraint). The constraints are specified by predicates (InstantPredicate and DurationPredicate) and both of them refer to the TimedObservation. TimedConstraint can be seen as the precondition for the transition:

TimedConstraint≙[OBi:TimedObservationi;...]•TimedExpression(OBi.tj)

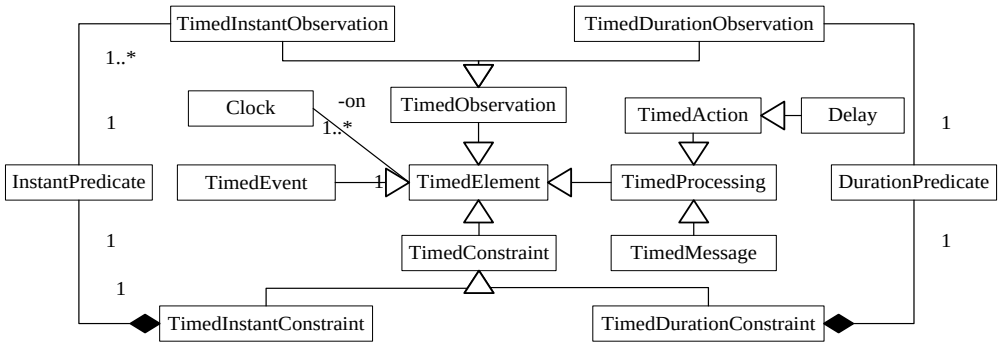

Fig. 9: MARTE time structure

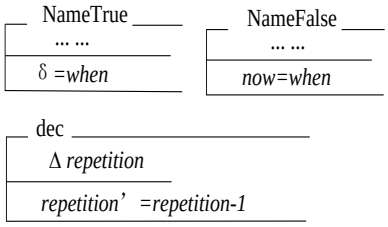

Fig. 10: Scheme of NameTrue, NameFalse and dec

A TimedEvent is an event the occurrences of which are bound to clocks. The when property specifies when the first occurrence occurs. The every property specifies the duration between successive occurrences. The number of occurrences can be limited by the repetition property. When is Relative is true, the event occurs after entering a state with a duration of when else event occurs at an absolute time when. NameTrue and Name False are the schemes corresponding to the event when is Relative is true and false respectively, which schemes are given in Fig. 10. TCOZ introduces a distinguished identifier δ to represent the duration of the state calculations performed by the operation.

The TimedEvent is transformed into the event_Name defined below. actionSequence is the actions triggered by the event:

event_Name ≙μC•[*isRelative*∧*repetition*]• NameTrue;actionSequence;dec→*WAIT* every ;C [□*isRelative*∧*repetition*]• *NameFalse;dec*→*WAIT* every; C

The TimedProcessing stereotype is a generic concept for modeling activities whose instants and durations are explicitly bound to clocks. When it is applied to a state machine, all the TimedElement inside are transformed according to the rules defined above. A delay represents a null operation lasting for a given duration, a delay of t after process P is expressed below.

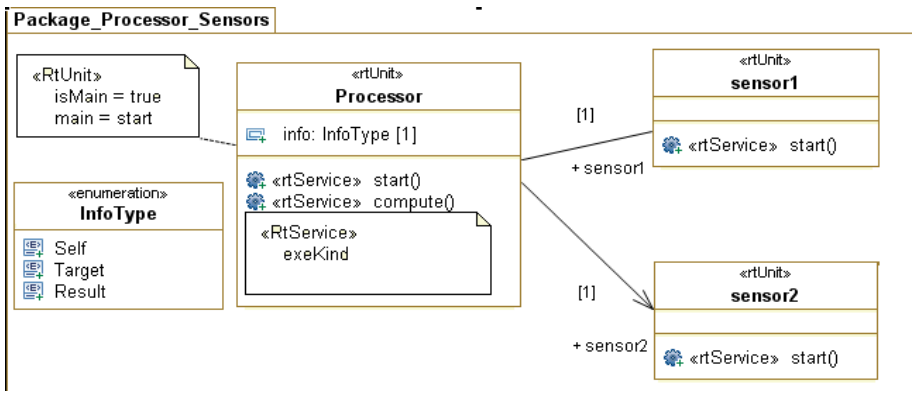

Fig. 11: Static structure of the system

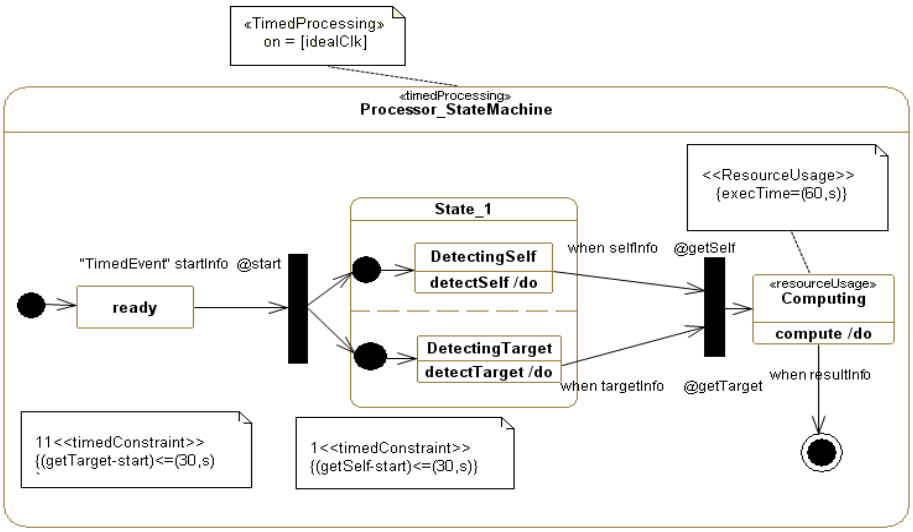

Fig. 12: Dynamic model of the system

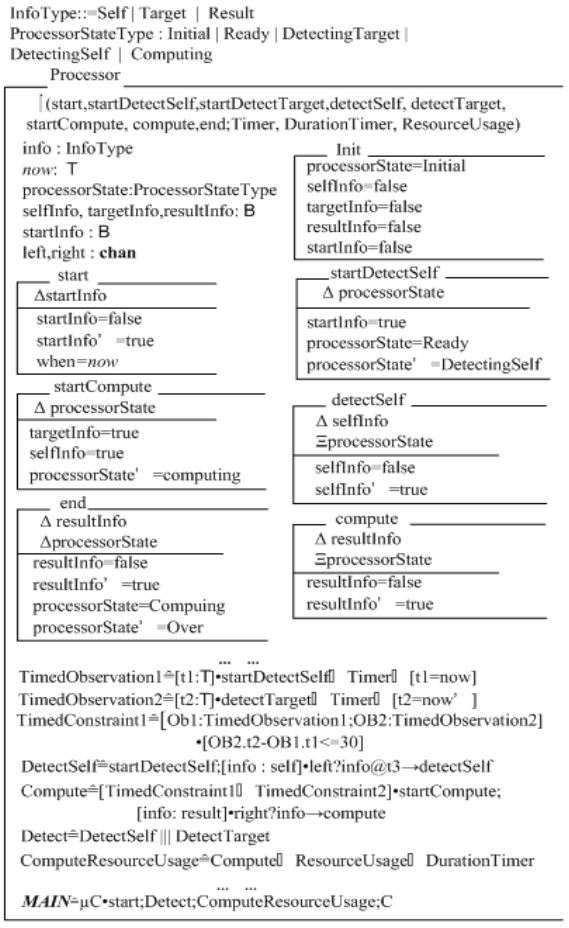

Fig. 13: TCOZ specification of the system

CASE STUDY

UML modeling annotated with MARTE: Exploration robot is a typical TCS and it has one central processor and two sensors. One sensor is employed to detect the target information such as the direction and obstacles. The other one is employed to detect information itself such as forward direction and speed. The central processor computes the route through the information from the sensors. Recently CEA List Team (2008) developed papyrus to support the MARTE modeling. The logical and dynamic model annotated with MARTE in papyrus are given in Fig. 11 and 12.

The processor and two sensors are stereotyped by RtUnit, which represents an autonomous execution resource. The isMain property is true represents that the processor has the same life cycle with the system. The main property shows the main entry of the system. The operations are stereotyped by RtService which depicts the concurrent policy, synchronization kind etc.

In this study, we are just concerned about the state machine corresponding to the processor. The state machine refers to an ideal clock through stereotype TimedProcessing. The trigger start is stereotyped by TimedEvent which has a period of 100 sec. TimedInstantObservation associates to the occurence of timed event start, change event getTarget and getSelf to record the occurring time. The change event getResult triggers the transition when the compute process is over. The state computing is stereotyped by ResourceUsage to depict the time elapsed in the inner transition. The stereotype TimedConstraint depicts the constraint of the TimedObservation.

Model transformation: Through parsing the XMI files created by papyrus and applying the mapping rules in the above section, the LaTeX format TCOZ specification can be got. Here we give the brief specification in Box format in Fig. 13 for reading.

The detailed transformation process is omitted for space limitation of this study.

Although we have proposed the approach to transform TCS model to TCOZ specification, there are still some limitations need to be pointed out. Firstly our work just covers a small part of MARTE. Secondly TCOZ is currently lacking in complete tool chains which limits the industrial application of this approach.

CONCLUSION

Combining of real time software modeling and formal method can enhance the dependability of system. In this study, we use UML model annotated with MARTE to represent a TCS. We provided the mapping rules between TCS model and TCOZ at the meta-model level. The main contributions of this study can be summarized as follows:

- Mapping rules for UML static model especially some real-time related MARTE types to approximately express them in TCOZ.
- Transformation from statechart diagram to TCOZ by defining the mapping rules at meta model level.
- Redefinition of MARTE stereotypes in TCOZ especially time and energy non functional property issues by defining scheme of Timer and DurationTimer.

The future research directions are given as follows:

- Sequence diagram formalism to analysis the communication of the system.
- Other MARTE non-functional property such as performance analysis modeling etc.

REFERENCES

Brendan, M. and J.S. Dong, 2000. Timed communicating object Z. IEEE T. Software Eng., 26(2): 150-177.

CEA List Team, 2008. Papyrus1.12 (Open Source Tool for Graphical UML2 Modeling). Retrieved from: http:// www.papyrusuml.org.

Ermeson, A., M. Paulo, C. Gustavo and N. Bruno, 2008. Mapping UML interaction overview diagram to time petri net for analysis and verification of embedded real-time systems with energy constraints. CIMCA, 2008: 615-620.

Mekki, A. and M. Ghazel, 2010. Time-constrained systems validation using MDA model transition: A railway case study. 8th International Conference of Modeling and Simulation.

Mirco, T. and S. Gilmore, 2008. Automatic extraction of PEPA performance models from UML activity diagrams annotated with the MARTE profile. Proceedings of the 7th International Workshop on Software and Performance, pp: 67-78.

Mostafa, M.A., M.A. Ismail and H. EL-Bolok, 2007. Toward a formalization of umL 2.0 metamodel using Z specifications. 8th ACIS International Conference on Software Engineering, Artificial, pp: 694-701.

Nianhua, Y., 2010. Modeling UML sequence diagrams using extended Petri nets. Proceedings of the International Conference on Information Science and Applications, pp: 596-603.

OMG, 2009. Unified Modeling Language Specification, Version 2.2. Retrieved from: http:// www. omg.org/spec/UML/2.2/ Infrastructure/PDF

OMG, 2011. UML Profile for MARTE: Modeling and Analysis of Real-time Embedded Systems. Retrieved from: http:// www. omg.org/ spec/ MARTE/ 1.1.

Reed, G. and A. Roscoe, 1986. A timed model for communication sequential processes. Proceeding of 13th International Colloquium on Automata Languages and Programming Rennes: Springer-Verlag, LNCS226: 314-323.

Sergiu-Mihai, D., 2001. Combining semi-formal and formal notations in software specification: An approach to modeling time-constrained systems. Ph.D. Thesis, at Dalhousie University.

Smith, G., 2000. The Object-Z Specification Language: Advances in Formal Methods. Kluwer Academic Publishers, USA.

Wei, J. and H. Wang, 2011. Modeling MARTE sequence diagram with timing pi-calculus. 14th IEEE International Symposium on Object/Component/ Service - Oriented Real - Time Distributed Computing, pp: 61-66.

Zhang, T. and F. Jouault, 2009. MDE-Based mode transformation: From MARTE model to FIACRE model. J. Software, 20(2): 214-233.