A SCHEDULING ANALYSIS FRAMEWORK FOR PREDICTING THE WEAKLY HARD REAL-TIME SYSTEMS

HABIBAH BINTI ISMAIL

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

JUNE 2013

To my beloved husband, sons, daughter, mom, late father, siblings, parents-in-law, brothers and sisters-in-law and families

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere appreciation to my supervisor, Associate Professor Dr Dayang Norhayati Abang Jawawi, for her encouragement, guidance, critics, friendship, advices, motivations and constant help during my study. Without her continued support and interest, this thesis would not have been the same as presented here. I have learned a lot from her and I am fortunate to have her as my mentor and supervisor.

I am also indebted to Universiti Teknologi Malaysia (UTM) for providing the facilities, good environment and funds to support my study. All of UTM members also deserve special thanks for their assistance in supplying the relevant efforts. Not forgotten, the Research University Grant (RUG) and Fundamental Research Grant Scheme (FRGS) for funding my study at UTM.

Also, I would like to thank profoundly to my fellow postgraduate's friends for their cooperation, supportive, views and help. My sincere appreciation also extends to all my colleagues, especially in Embedded Real-Time and Software Engineering Lab (EReTSEL), Software Engineering Research Group (SERG) and to all those involved either directly or indirectly. Unfortunately, it is not possible to list all of them in this limited space. Finally, I am grateful to all my family members for their continuous support, love, help and attention.

ABSTRACT

For real-time systems, hard real-time and soft real-time systems are based on "miss restriction" and "miss tolerance", respectively. However, a weakly hard real-time system integrates both these requirements. The problem with these systems is the limitation of the scheduling analysis method which only uses the traditional scheduling approach. Besides that, the current framework has problems with the complexity and predictability of the systems. This study proposed a scheduling analysis framework based on the suitability of scheduling algorithms, weakly hard real-time modelling and the combination of the deterministic and probabilistic schedulability analyses for predicting the weakly hard real-time tasks. Initially, the best fitting specification of a weakly hard real-time system was integrated into the proposed framework and tested in the Modeling and Analysis of Real-Time Embedded systems (MARTE) profile. The profile was enhanced because the current MARTE timing constraint restricted to the hard and soft real time timing requirement, thus some modifications were made to model the weakly hard real-time requirements. For complex systems, rather than only using scheduling algorithms to schedule the tasks, the algorithms were used with Unified Modeling Language (UML) modelling. Sequence diagram complexity factor metrics were used to measure the behavioural complexity. The proposed combination approach was applied on case studies and then evaluated with reference to the existing approaches. The results of the evaluations showed that the proposed framework is more predictable compared to the other frameworks and has addressed the problem posed in this research. In conclusion, the proposed scheduling analysis framework provides a less complex design through the behavioural complexity measurements, as well as increases the predictability of the systems.

ABSTRAK

Bagi sistem masa nyata, sistem masa nyata keras dan lembut masing-masing adalah berdasarkan "sekatan kehilangan" dan "kehilangan bertoleransi". Walau bagaimanapun, sistem masa nyata keras yang lemah menggabungkan kedua-dua keperluan tersebut. Masalah dengan sistem ini adalah keterbatasan kaedah analisis penjadualan yang hanya menggunakan pendekatan penjadualan tradisional. Selain itu, rangka kerja semasa mempunyai masalah dengan kerumitan dan kebolehramalan sistem. Kajian ini mencadangkan satu rangka kerja analisis penjadualan berdasarkan kesesuaian algoritma penjadualan, pemodelan masa nyata keras yang lemah dan gabungan analisis penjadualan berketentuan dan kebarangkalian untuk meramalkan tugas masa nyata. Pada mulanya, spesifikasi terbaik telah disepadukan ke dalam rangka kerja yang dicadangkan dan diuji dalam profil Pemodelan dan Analisis Sistem Terbenam Masa Nyata (MARTE). Profil tersebut telah dipertingkatkan kerana kekangan masa MARTE semasa terhad kepada keperluan masa nyata keras dan lembut, dengan itu beberapa pengubahsuaian telah dibuat untuk memodelkan keperluan masa nyata keras yang lemah. Bagi sistem yang kompleks, selain hanya menggunakan algoritma penjadualan sahaja untuk menjadualkan tugas, algoritma telah digunakan bersama dengan Bahasa Pemodelan Bersepadu (UML) model. Metrik faktor kerumitan gambarajah berjujukan digunakan untuk mengukur kerumitan tingkah laku. Pendekatan gabungan yang dicadangkan telah digunakan pada kajian kes dan kemudian dinilai dengan merujuk kepada pendekatan yang sedia ada. Keputusan penilaian menunjukkan bahawa rangka kerja yang dicadangkan adalah lebih mudah diramalkan berbanding dengan yang lain dan ia telah menangani masalah yang ditimbulkan dalam kajian ini. Kesimpulannya, cadangan rangka kerja analisis penjadualan menyediakan reka bentuk yang kurang kompleks melalui ukuran kerumitan tingkah laku, serta meningkatkan kebolehramalan sistem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMR	-	Autonomous Mobile Robot
BMS	-	Bi-Modal Scheduler
BWP	-	Blue When Possible
CAN	-	Controller Area Network
CBS	-	Constant Bandwidth Server
C(SQ)	-	Sequence Diagram Complexity
DBP	-	Distance-Based Priority
DMA	-	Deadline Monotonic Algorithm
DWCS	-	Dynamic Windows-Constrained Scheduling
EDF	-	Earliest Deadline First
EPL	-	Eclipse Public License
FP	-	Fixed-Priority
GA	-	Genetic Algorithm
GCM	-	Generic Component Model
GQAM	-	Generic Quantitative Analysis Modeling
GRM	-	General Resource Modeling
HLAM	-	High-Level Application Modeling
HRM	-	Hardware Resource Modeling
LCM	-	Least Common Multiple

LST-CAN	-	Latest Send Time-CAN
MARTE	-	Modeling and Analysis of Real-Time Embedded systems
NIMSAD	-	Normative Information Model-based Systems Analysis and Design
OMG	-	Object Management Group
PAM	-	Performance Analysis Modeling
PDF	-	Probability Density Function
PFs	-	Probability Functions
PTDA	-	Probabilistic Time Demand Analysis
RHSM	-	Robotic Highway Safety Markers
RLP	-	Red as Late as Possible
RMA	-	Rate Monotonic Algorithm
RM-RTO	-	Rate Monotonic RTO
RTO	-	Red Tasks Only
SAM	-	Schedulability Analysis Modeling
SDCF	-	Sequence Diagram Complexity Factors
SMUF	-	Sequence Method Utilization Factor
SMW	-	Sequence Message Weight
SOW	-	Sequence Object Weight
SPT	-	Schedulability, Performance and Time
SRM	-	Software Resource Modeling
SRMS	-	Statistical Rate Monotonic Scheduling
STDA	-	Stochastic Timed Demand Analysis
SysML	-	Systems Modeling Language
TDA	-	Time Dilation Algorithm

TDA	-	Time Demand Analysis
UML	-	Unified Modeling Language
WCET	-	Worst Case Execution Time
WCRT	-	Worst Case Response Times
WHRTCBS	-	Weakly Hard Real-Time Constant Bandwidth Server

LIST OF SYMBOLS

D_i	-	Deadline
T_i	-	Period
C_i	-	Worst case execution time
t	-	Time instant
U	-	Utilization factor
N	-	The number of tasks
τ_i	-	Task
μ_i	-	Worst case execution pattern
R_i	-	Worst case response times
k	-	Upper bound
H	-	The period of time is called the hyperperiod
h_i		The number of invocations of a task in the hyperperiod at level *i*
A_i		Task invoked in the hyperperiod
a_i		Task invoked in the hyperperiod at level *i*
λ	-	Weakly hard constraints
hp	-	The set of higher priority tasks
$\mathrm{G_i}$	-	Equal to the worst case execution time
(┼)	-	Explicitly mentioned

(┬)	-	Implicitly mentioned
(—)	-	Not mentioned at all

CHAPTER 1

INTRODUCTION

1.1 Overview

Real-time systems are computer systems in which the correctness of the system depends not only on the logical results, but also on the time factors at which the results are produced. Formally, a classification of real-time tasks or systems is based on the importance of missing a deadline.

Traditional real-time systems are classified into two categories, namely, hard real-time systems and soft real-time systems (Shin et al., 1994). In applications of real-time for hard real-time systems, no missed deadline is tolerated; in other words, the deadline must be met successfully, otherwise there is a damaging effect on the system. For soft real-time systems, the missed deadline is tolerated as long as it is minimised and occurs occasionally; however, the term “occasional” is not precise. Nevertheless, it is still acceptable even though the task is delayed because missing a deadline usually happens in a non-predictable way.

The new generation for a real-time system is the weakly hard real-time system which provides a mechanism that can tolerate some deadlines using specifications in a clear, predictable and bounded way where the deadlines can be missed; thus, the following advantages have been identified when defining weakly hard tasks in a real-time system (Bernat et al., 2001):

- Alleviating the pessimism in the parameters of the system and worst-case scenarios as occurs with all hard real-time tasks.
- Providing a mechanism for fair degradation of the quality of the service tasks.
- Obtaining a fair mechanism for deciding which task needs to be skipped during transient overload.

A hard real-time system is very restrictive because all the tasks must meet the deadlines or, in other words, no deadlines are allowed to be missed. Meanwhile, a soft real-time system is too relaxed because no guarantee can be given to the deadline, whether it is met or missed. As hard real-time and soft real-time systems are based on “miss restriction” and “miss tolerance”, respectively, the weakly hard real-time system can integrate both of these requirements in which the distribution of its met and missed deadlines during a window of time is precisely bounded. For weakly hard real-time tasks, the missed or lost deadline happens occasionally and it can be considered, however it is still necessary and crucial to finish the tasks within a given deadline, otherwise failure occurs for the tasks. In a weakly hard real-time system, the number of deadlines that may be missed can be specified; in other words, it may specify exactly how many deadlines may be missed in the worst case. This makes a weakly hard real-time system stronger than a soft real-time system.

Multimedia systems are a typical example of systems with weakly hard real-time requirements because in such a system is it unnecessary to meet the entire tasks and the deadlines as long as the misses (or deadlines) are spaced distantly/evenly. Hard tasks usually co-exist with soft tasks; thus, it means that most hard real-time tasks are not that hard actually. The occasional miss or loss of some deadline can be tolerated but missing deadlines have to be stated precisely; that is, the way it happens is predictable and accurately known. For example, in autonomous mobile robot (AMR) systems, the tasks are divided into two group (or types) according to hard and soft real-time tasks (Jawawi et al., 2006). Thus, missing the deadlines of certain tasks is acceptable for AMR software. As a consequence, AMR can be defined as weakly hard real-time tasks.

A framework for the schedulability analysis of real-time tasks can determine whether a specific task set derived from a software model can satisfy certain timing constraints and can be successfully scheduled. That framework enables scheduling analysis to predict the behaviour of critical tasks by meeting the deadline and at the same time predicting the bounded way in which missing of some deadlines is acceptable by comparison to less critical tasks (Bernat, 1998). A more realistic framework is required for the scheduling analysis of weakly hard real-time tasks because the constraints of missing deadlines do not exist in hard real-time task analysis frameworks and are not stated precisely in soft real-time tasks analysis frameworks (Bernat et al., 2001).

In order to determine whether a real-time system can run within the timing constraints put upon it, a number of different algorithms have been designed to analyse a system and determine whether it is schedulable or not. The timing constrained requirements are the direct input for the scheduling analysis algorithms. Scheduling analysis is a mathematically sound way of predicting the timing behaviour of a set of real-time systems (Klein, 1993). There are many scheduling policies that can be used in real-time system development. The most well-known and widely-used scheduling algorithms for real-time tasks are the rate monotonic (RM) algorithm and deadline monotonic (DM) algorithm for fixed priority scheduling and the “earliest deadline first” (EDF) algorithm for dynamic priority scheduling (Liu and Layland, 1973).

Modelling timing constraints and scheduling behaviour through the adaptation of modelling language is recognised as an alternative way to predict the timing behaviour and performance of set concurrent tasks in order to react to the changing environment (Jensen, 2009). This is due to the increasing complexity of contemporary ubiquitous real-time systems which require an adequate modelling language. The well-known and most widely-used modelling language for software modelling systems is the Unified Modelling Language (UML). The new extension for the UML profile, called Modelling and Analysis of Real-Time Embedded (MARTE) system, has been standardised by the Object Management Group (OMG) to be the future standard for UML modelling of real-time and embedded systems although a number of other modelling standards exist already (MARTE OMG, 2007). This new profile is intended to replace the existing UML Profile for Schedulability, Performance and Time (SPT) because MARTE provides some new key features such as support for non-functional property modelling and adds rich time and resource models to the UML.

1.2 Background of the Problem

The accuracy of real-time software depends not only on the logical results, but also on the time at which the outputs are generated. This is due to the fact that these software systems interact with the physical world or environment via sensors and actuators and this environment changes with time. Thus, real-time software is difficult to develop because, besides the functionality, the timing of each software task is an important factor that needs to be considered.

Due to the need for timing analysis, scheduling theories have been developed to provide and offer mathematically fundamental tools to predict the timing behaviour of set concurrent tasks. However, these theories have not fulfilled most of the application requirements since the scheduling algorithms have been successfully executed only on feasible systems (systems in which all the deadlines have to be met) and on infeasible systems (systems which could tolerate missed deadlines), as the performance for such algorithms may be executed poorly and may be unacceptable (Anderson and Baruah, 2004). Zhu (2009) presented weakly hard real-time combination constraints and proposed a new Constant Bandwidth Server (CBS) algorithm, which uses weakly hard real-time systems to reduce the variance in all tasks (called WHRTCBS). However, the WHRTCBS cannot apply to the periodic tasks because the tasks of the WHRTCBS are aperiodic tasks and their deadlines are random.

For complex systems, besides using the scheduling algorithms only to schedule tasks and determine whether a task is schedulable or not, the algorithms can be used together with UML because UML is a commonly accepted modelling language for complex systems (Jensen, 2009). The MARTE profile, a standard was defined to improve the specification of timing requirements and to prepare models for timing analysis (Woodside, 2007). The problem with the current MARTE profile is that its timing constraint has been restricted with hard and soft real-time systems. For soft real-time systems, the timing requirement, called the “miss ratio” is already defined in the MARTE profile. However, the window of time over the maximum ratio is not well specified in the MARTE profile. Therefore, to specify the maximum allowable deadlines that may be missed more precisely, weakly hard real-time requirements need to be added to the MARTE profile.

The modelling profile must cope with the complexity of the system, including the structure and behaviour aspects. As a result, it is essential to evaluate which model copes with the complex structure and behaviour as well as its non-functional requirements. The model’s features must allow designers to map the problem domain semantic directly onto the model. This is beneficial for users in order to develop maintainable and less complex real-time systems (Pereira, 2000). The behaviour of the system is known as a set of external and internal sequences of events, actions and transitions (Harel and Gery, 1997). It also can be said that the behaviour of a system is the response to the external events and the execution of actions that are taken at any time (Rational Software Corporation, 2003). Hence, it is important to measure the behavioural complexity of design in weakly hard real-time systems in order to reduce the system’s complexity.

The most important feature of real-time systems is their predictability (Goossens et al., 2001). The objective in real-time systems is to meet the timing requirements of the tasks and the property required to do so is predictability. In real-time systems it is required to guarantee that the temporal constraints will be met during execution. Also, predictability requires that information about the system is known. The schedule theory of the weakly hard real-time systems aims to solve the situation that most real-time applications can tolerate certain deadlines to be missed but the challenge is to ensure the missing of deadlines occurs under a precise distribution over a finite time window (Zhu, 2009). Meeting all the deadlines is impossible; thus, Bernat (1998) provided a conceptual framework for specifying real-time systems that can tolerate occasional losses of deadlines in which the distribution of the met and lost deadlines is precisely bounded.

Some researchers have considered the control system as a case study; for example, Bernat and Cayssials (2001) used a robot control system case study to apply a scheduling framework called the bi-modal scheduler. However, the DMA they used for the schedulability analysis is not optimal for weakly hard systems. Another similar work was done by Broster et al. (2002) in which they used weakly hard constraints on a controller area network. However, the fault model used in the schedulability analysis has limitations. Since a large number of studies on weakly hard real-time systems have used mobile robot systems in their schedulability analysis, it makes sense that the mobile robot control system case study is the best case study for this type of study. This is supported further by the cases analysis performed by Bernat et al. (2001) in which the robot control system was a system mixture of hard and soft tasks, thus it can be described as a weakly hard task system. It is generally required to specify the upper bounds on the number and pattern of deadlines missed during a period of time.

Some promising efforts about the specification of weakly hard real-time systems have been reported. For example, Hamdaoui and Ramanathan (1995) presented the notion of (*m, k*)-firm deadlines to specify tasks (or messages) which are desired meets at least *m* deadlines in any window of *k* consecutive invocations in the context of scheduling messages. They declare that, for all the tasks, there is no differential between the parameters *m* and *k*. They presented a scheduling algorithm called the distance-based priority (DBP) assignment, where tasks which are closer to missing their (*m*, *k*)-firm constraint have higher priority. This approach is a best-effort scheduling algorithm where no guarantee can be obtained and it offers a straightforward priority assignment policy. It only works to minimise the number of tasks that could be missed but no guarantee is given on the number of deadlines a task can miss. However, those ideas only use the *m* and *k* deadline model while the richer information on the whole pattern (zeroes and ones) contained is neglected.

Koren and Shasha (1995) introduced the skip-over scheduling algorithm in which the algorithm skips some task invocations according to the notion of the skip factor, *s*. If a task has a skip factor of *s* it will have one invocation skipped out of *s* consecutive invocations. That means the distance between two consecutive skips must be at least *s* periods (it can be specified as a (*s* – 1, *s*)-constraint). When $s =$ infinity (∞), no skips are permitted. However, the disadvantage of this skip constraint is that a selected number of task invocations are discarded (or skipped) even though the tasks could meet their deadline or there may be available computation resources to finish on time.

Koren and Shasha (1995) also introduced the (*m*, *k*)-constraint which is equivalent to the $\binom{n}{m}$ constraint introduced by Bernat et al. (2001). This approach is the closest technique to the one used in our study. They introduced the notion of the $\binom{n}{m}$ constraint that is useful for weakly hard real-time systems which are expanded from the concept of $\binom{m}{k}$ constraints. Most importantly, they clearly specified the number of deadlines a task could miss and the pattern of how these deadlines can be missed with the introduction of four temporal constraints, also known as weakly hard constraints, and the two patterns (zeros and ones) that represent a missed deadline with a 0 and a met deadline with a 1, also called the μ-patterns.

As a result, a motivation arises from the review of these three works to undertake an evaluation by comparing these three weakly hard real-time specifications in order to find which specification is better in predicting the behaviour of a task if a deadline is missed. This comparison can be based on several criteria by experimenting with the case study.

1.3 Formation of Research Questions

Most existing frameworks for the analysis and scheduling of real-time tasks are focused on hard and soft real-time tasks. A more suitable framework is required for the scheduling analysis of weakly hard real-time tasks by using the weakly hard constraints. The framework must enable the prediction of the behaviour of a task in the case where deadlines are missed, including the number of deadlines missed and how many times the tasks missed the deadlines.

As discussed in relation to the problem background, there are three well-known specifications of weakly hard real-time systems. As each specification has its own approach, it is essential to evaluate which specification is better able to predict the behaviour and performance of a task. To do so, an initial evaluation must compare the weakly hard real-time specifications that are commonly used in the academic field.

As real-time systems become more complex, alternative methods are required to reduce the complexity and to predict the timing behaviour of weakly hard real-time tasks besides using the typical (or traditional) scheduling approach. Therefore, in this research, a UML-MARTE profile will be used as the modelling language to model weakly hard real-time systems. Nevertheless, the problem with the current UML-MARTE profile is that it cannot support the timing requirements of weakly hard real-time systems because its profile is restricted to hard and soft real-time requirements (MARTE OMG, 2007). Thus, the existing UML-MARTE standard needs to be modified in order to implement weakly hard real-time tasks in the UML-MARTE profile.

The issue arising herein is how to increase the predictability of weakly hard real-time tasks in terms of the deadlines missed. This is because, even though some deadlines can be missed, the tasks still need to be guaranteed to be predictable by meeting the timing requirements, such as, specifying in clear such a met and missed deadlines of the tasks. The problem with predictability in the current framework is that its scheduling work is limited to the use of deterministic schedulability analysis only; thus, to move away from this limitation, and domination; the deterministic schedulability analysis can be used with probabilistic schedulability analysis in order to provide more predictable weakly hard real-time tasks.

A research question needs to be answered in order to solve the research problem. Derived from the research problem, the following research question is addressed in this study:

How can a scheduling analysis framework with less complexity and more predictability for weakly hard real-time task performance and behaviour be developed?

To answer the main research question, the following sub-questions need to be addressed:

1) How can the temporal constraints for weakly hard real-time systems be defined?

    a) What are temporal constraints and why do we need temporal constraints?
    b) What are the current specifications of weakly hard real-time systems?

c) What criteria should be considered in evaluating the specification?

2) How can systems for weakly hard real-time tasks be scheduled?

   a) Which scheduling algorithm is suitable to use with weakly hard constraints?

3) How does the MARTE profile support weakly hard real-time requirements?

   a) Why is it necessary to model weakly hard real-time requirements using the MARTE profile?
   b) What are the problems with the current MARTE profile?
   c) How can the MARTE support for weakly hard real-time requirements be proved?

4) How can a probabilistic schedulability analysis be added to a deterministic schedulability analysis?

   a) Why is it necessary to propose the combination of deterministic and probabilistic schedulability analyses?
   b) Does the proposed framework perform better than the existing framework?
   c) What criteria should be considered in evaluating the scheduling analysis framework?

1.4 Objectives of the Study

The objectives of the study are as follows:

1) To enhance the UML-MARTE profile as the modelling language in order to model weakly hard real-time requirements.
2) To propose a scheduling analysis framework based on the weakly hard real-time modelling and the deterministic and probabilistic schedulability analyses.
3) To evaluate the complexity of the profiles in the framework and to evaluate the proposed scheduling analysis framework by comparison with the existing framework.

1.5 Scope of the Research

The scope of this research is defined by the following parameters:

- It provides a framework for specifying real-time tasks that could allow several deadlines to be missed occasionally.
- It uses a schedulability analysis to predict task performance.
- It focuses on mobile robot system requirements.
- It uses UML profiles for visualisation of the design model.
- It focuses on the basic real-time modelling and schedulability analysis such as how to model the information required.
- It particularly deals with MARTE modelling capabilities to enable predictive quantitative analysis, namely, schedulability for weakly hard real-time tasks.

1.6 Significance of the Study

This research aims to contribute towards improved real-time scheduling by providing a scheduling analysis framework for predicting the weakly hard real-time task behaviour and performance. In order to cope with the increasing complexity in real-time systems, a modelling language is used, and as a way to address the problem in the current profile, modifications of the UML-MARTE profile are done in order to support the timing requirements and predictions of weakly hard real-time systems. Moreover, in order to increase the predictability of weakly hard tasks in terms of the number of deadlines missed, we propose the combination of deterministic and probabilistic schedulability analyses.

1.7 Organisation of the Thesis

The thesis is structured in seven chapters. This chapter provided an overview of weakly hard real-time systems. The background of the problem and the motivations for the research were explained. The research objectives and scope were also identified. Chapter 2 describes the basic theory of weakly hard real-time systems. The literature on real-time systems is reviewed in order to understand the work related to the objectives of the present research. Chapter 3 sets out the research methodology and describes the research flow.

Chapter 4 explains the general comparisons and case study comparisons carried out to investigate the best-fit weakly hard real-time specification. Chapter 5 presents the proposed scheduling analysis framework in detail, and discusses how it can solve the two main issues of reducing system complexity and increasing the predictability of the systems. Also, it explains the strategy of the modifications processes of UML-SPT and UML-MARTE profiles. Chapter 6 provides a basic schedulability analysis of the proposed framework. This includes analysis of the combination approach and behavioural complexity measurements, and then the proposed scheduling analysis framework is evaluated by reference to the closest extant research. In Chapter 7 we conclude our work and make suggestions for promising directions in future research.

REFERENCES

Agrawal, S., Shankar, R. and Ranvijay (2011). A Three Phase Scheduling for System Energy Minimization of Weakly Hard Real-Time Systems. *Global Journal of Computer Science and Technology*. vol. 11(10). version 1.0. May.

Anderson, J. H. and Baruah, S. K. (2004). Energy-Efficient Synthesis of Periodic Task Systems upon Identical Multiprocessor Platforms. *Proceedings 24th International Conference on Distributed Computing Systems* (*ICDCS'04*). 428-435.

Arlitt, M. and Williamson, C. (1996). Web Server Workload Characterization: The Search for Invariants. *In Proceedings of the ACM SIGMETRICS '96 Conference*. Philadelphia, PA. May.

Atlas, A. K. and Bestavros, A. (1998). Statistical Rate Monotonic Scheduling. *In Proceedings of the 19th IEEE Real-Time Systems Symposium*. December. Madrid, Spain.

Audsley, N.C., Burns, A., Richardson, M.F., and Wellings A.J. (1991). Incorporating Unbounded Algorithms into Predictable Real-Time Systems. *Computer Systems Science and Engineering*. vol. 8(3). 80–89.

Audsley, N.C., Burns, A., Richardson, M.F., and Wellings A.J. (1991). Hard Real-Time Scheduling: The Deadline-Monotonic Approach. *In Proceedings 8th IEEE Workshop on Real-Time Operating Systems and Software*. May.

Audsley, N. C. and Burns, A. (1990). *Deadline-Monotonic Scheduling*. Technical Report YCS146. Department of Computer Science, University of York, UK.

Bernat, G., Burns, A. and Llamosi, A. (2001). Weakly Hard Real-Time Systems. *IEEE Transactions on Computers*. 50(4), April, 308-321.

Bernat, G. and Cayssials, R. (2001). Guaranteed On-line Weakly Hard Real-Time Systems. *22nd IEEE Real-Time Systems Symposium (RTSS'01)*. December. London, England: 25-35.

Bernat, G. (1998). *Specification and Analysis of Weakly Hard Real-Time Systems*. PhD Thesis. January. Department de les Ciències Matemàtiques i Informàtica. Universitat de les Illes Balears. Spain.

Boegholm, T., Kragh-Hansen, H., Olsen, P., Thomsen, B. and Larsen, K.G. (2008). Model-Based Schedulability Analysis of Safety-Critical Hard Real-Time Java Programs. *Proceedings of the 6th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES'08)*. September. Santa Clara, California, USA: 106-114.

Booch, G., Rumbaugh, J. and Jacobson, I. (2005). *Unified Modeling Language User Guide, The (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional.

Broster, I. *et al*. (2003). FIRST Architecture Framework: Key Scheduling Technologies for the Future. *Deliverable D-AF.1-v1*. January.

Broster, I., Bernat, G., and Burns, A. (2002). Weakly Hard Real-Time Constraints on Controller Area Network. *14th Euromicro Conference on Real-Time Systems (ECRTS'02)*. June. Vienna, Austria: 134.

Burns, A., Bernat, G. and Broster, I. (2003). *A Probabilistic Framework for Schedulability Analysis*. Lecture Notes in Computer Science. vol. 2855. 1-15.

Burns, A. and Bernat, G. (2001). *Jorvik: A Framework for Effective Scheduling*. Real-Time Systems Research Group. Department of Computer Science, University of York, UK.

Cruz, R. L. (1991). A Calculus for Network Delay, Part 1: Network Elements in Isolation. *IEEE Trans on Information Theory*. vol. 37(1). January. 114-131.

Diaz, J. L. and Lopez, M. L. (2001). *Probabilistic Analysis of the Response Time in a Real-Time Systems*. Technical Report. University of Oviedo.

Douglass, B. P. (2004). *Real Time UML: Advances in the UML for Real-Time Systems*. CA, USA: Addison Wesley Longman Publishing Co. Inc. Redwood City.

Fischer, K., Hordys, G. and Vogel-Heuser, B. (2004). Evaluation of an UML Software Engineering Tool by Means of a Distributed Real Time Application in Process Automation. *Lecture Notes in Informatics, Modellierung*. 135-148.

Gardner, M. K. (1999). *Probabilistic Analysis and Scheduling of Critical Soft Real-Time Systems*. PhD Thesis. University of Illinois, Urbana-Champaign.

George, R. and Kanayama, Y. (1996). A Rate Monotonic Scheduler for the Real-Time Control of Autonomous Robots. *Proceedings of IEEE International Conference on Robotics and Automation*. 2804-2809.

Goossens, J. and Macq, C. (2001). Limitation of the Hyper-Period in Real-Time Periodic Task Set Generation. *In Proceedings of the RTS Embedded System* (*RTS'01*).

Hamdaoui, M. and Ramanathan, P. (1995). A Dynamic Priority Assignment Technique for Streams with (m,k)-firm Deadlines. *IEEE Transactions on Computers*. vol. 44(12). December. 1443-1451.

Harel, D. and Gery, E. (1997). Executable Object Modeling with Statecharts. *Computer* 30(7). 31-42.

Harter, P. K. (1984). *Response Times in Level Structured Systems*. Department of Computer Science, University of Colorado, USA. CU-US-269-94.

Hasan, A. A. M. and Deris, S. (2006). Real-Time Scheduling for Autonomous Mobile Robots using Genetic Algorithm and Constraint Based Reasoning. Master's Thesis. Universiti Teknologi Malaysia.

I-Logix (2004). Rhapsody Tutorial in C++. Release 5.2. *Technical Report of I-Logix*.

Immonen, A. and Niemelä, E. (2008). Survey of Reliability and Availability Prediction methods from the viewpoint of software architecture. *Software and Systems Modeling*. vol. 7(1). 49-65.

Jawawi, D.N.A., Deris, S. and Mamat, R. (2006). Enhancement of PECOS Embedded Real-Time Component Model for Autonomous Mobile Robot Application. *IEEE International Conference on Computer Systems and Applications.* 882-889.

Jensen, K. E. A. (2009). *Schedulability Analysis of Embedded Applications Modelled using MARTE*. Master's Thesis. Technical University of Denmark, Kongens Lyngby, Denmark.

Joseph, M. and Pandya, P. (1986). Finding response times in a real-time system. *BCS Computer Journal*. vol. 29(5). October. 390–395.

Kanjilal, A., Sengupta, S. and Bhattacharya, S. (2009). Analysis of Complexity of Requirements: A Metrics Based Approach. *In Proceedings of the 2nd India Software Engineering Conference* (*ISEC'09*). 131-132.

Kheong, L. S. and Jayaratna, N. (2009). Framework for Structuring Learning in Problem-Based Learning. Retrieved Feb 11, 2011, from http://pbl.tp.edu.sg/Understanding%20PBL/Articles/lyejayarartna.pdf.

Kim, K., Bello, L. L., Min, S. L. and Mirabella, O. (2002). On Relaxing Task Isolation in Overrun Handling to Provide Probabilistic Guarantees to Soft Real-Time Tasks with Varying Execution Times. *In Proceedings of the 14th Euromicro Conference on Real-Time Systems*. June.

King, P. J. B. (1990). *Computer Communication Systems Performance Modelling*. Prentice Hall.

Klein, M.H., Lehoczky, J.P. and Rajkumar, R. (1994). Rate-Monotonic Analysis for Real-Time Industrial Computing. *Computer*. 25-32.

Klein, M. H. (1993). *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time System*. Boston: Kluwer Academic Publisher.

Koren, G. and Shasha, D. (1995). Skip-over: Algorithms and Complexity for Overloaded Systems that Allow Skips. *Proceedings of the 16th IEEE Real-Time Systems Symposium*. December. Pisa, Italy: 110-117.

Lehoczky, J. P. (1996). Real-Time Queuing Theory. *In Proceedings 17th IEEE Real-Time Systems Symposium*. December.

Lehoczky, J.P. (1990). Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. *Proceedings of 11th Real-Time Systems Symposium*. December. 201-209.

Leung, J.Y.-T. and Whitehead, J. (1982). On the Complexity of Fixed Priority Scheduling of Periodic, Real-Time Tasks. *Performance Evaluation*. vol. 4(2). December. 237–250.

Liu, C.L. and Layland, J.W. (1973). Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. *Journal of the ACM*. vol. 20 (1). 46-61.

Lu, W.-C., Hsieh, J.-W., Shih, W.-K. and Kuo, T.-W. (2006). A Faster Exact Schedulability Analysis for Fixed-Priority Scheduling. *Journal of Systems and Software*. vol.79. 1744-1753.

Mok, A. and Chen, D. (1997). A Multiframe Model for Real-Time Tasks. *IEEE Trans. Software Engineering*. vol. 23(10). October. 635-645.

Milutinovic, V. (1995). *A Research Methodology in the Field of Computer Engineering for VLSI*. vol. 2. 811-816.

Niu, L. and Xu, J. (2012). Improving Schedulability and Energy Performance for Weakly Hard Real-Time Systems. *IEEE 31st International Performance Computing and Communications Conference (IPCCC'12)*. 41-50.

OMG (2007). UML Specification version 2.1.2 OMG document formal/2007-11-04. *Technical report of Object Management Group*.

OMG (2005). UML Profile for Schedulability, Performance, and Time Specification. . *Technical report of Object Management Group*.

OMG MARTE Specification. (2007). A UML Profile for MARTE: Beta 1. OMG Adopted Specification ptc/07-08-04. August.

OMG (1997). The Unified Modeling Language for Object-Oriented Development, Documentation Set Version 1.0.

Pereira, C. E. (2000). Are Object-Oriented Concepts Useful to Real-Time Systems Development?. *Real-Time Syst.* 18(1). 89-94.

Qadi, A.A., Huang, J., Goddard, S. and Farritor, S. (2005). A Performance and Schedulability Analysis of an Autonomous Mobile Robot. *17th Euromicro Conference on Real-Time Systems (ECRTS'05)*. 239-248.

Rational Software Corporation (2003). C++ Reference. Rational Rose Real-Time. Version: 2003.06.00, Part no: 800-026109-000. *Technical report of Rational Software Corporation*. ftp://ftp.software.ibm.com/software/rational/docs/v2003/unix_solutions/pdf/RoseRT/rosert_cpp_ref_guide.pdf.

Shi, J., Goddard, S., Lal, A., Dumpert, J. and Farritor, S. (2005). Global Control of Robotic Highway Safety Markers: A Real-Time Solution. *Real-Time Systems*. vol. 29(2-3). 183-204.

Shin, K.G. and Ramanathan, P. (1994). Real-Time Computing: A New Discipline of Computer Science and Engineering. *Proceedings of the IEEE*. vol. 82(1).

Silly-Chetto, M. and Marchand, A. (2007). Dynamic Scheduling of Skippable Periodic Tasks: Issues and Proposals. *14th annual IEEE International Conference and Workshops on the Engineering of Computer-Based (ECBS'07)*. March. Tucson, Arizona, USA: 171-177.

Singh, Y., Popli, M. and Shukla, S. S. P. (2012). Energy Reduction in Weakly Hard Real-Time Systems. *1st International Conference on Recent Advances in Information Technology (RAIT'12)*. 15-17 March. 909-915.

Takagi, H. (1990). *Queuing Analysis of Polling system: An Update in Stochastic Analysis of Computer and Communication System*. Elsevier Science North Holland, Amsterdam: 267-318.

Tia, T. S., Deng, Z., Shankar, M., Storch, M., Sun, J., Wu, L. C. and Liu, J. S. (1995). Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times. *In Proceedings of Real-Time Technology and Applications Symposium*. 164-173.

Woodside, M. (2007). From Annotated Software Designs (UML SPT/MARTE) to Model Formalisms. *Proceedings of the 7th International Conference on Formal Methods for Performance Evaluation* (*SFM'07*). Springer-Verlag. Berlin, Heidelberg: 429-467.

Zhu, X. (2009). Research on Weakly Hard Real-Time Constraints and Their Boolean Combination to Support Adaptive QoS. *World Academy of Science, Engineering and Technology*.