Ph.D Project Description and Study Plan

Ph.D. Student: Claus Rørbæk Thrane (CRT) crt@cs.aau.dk
Supervisor: Prof. Dr. Kim Guldstrand Larsen (KGL) kgl@cs.aau.dk
Thesis Title: Quantitative Formalisms and Analysis *(Provisional title)*

October 9, 2008

This document presents the proposed project, which is to be carried out by CRT for duration of his Ph.D. stipend. The description of the project has has been developed in collaboration by CRT and KGL.
The proposed Ph.D. project and scientific research, is inspired by *The Embedded Systems Design Challenge* (HS06). It concerns research of *quantitative* formalisms extended from automata theory or possibly process algebra, as well as development of relevant quantitative *analysis*. The developed theory is expected to be applicable to modeling and analysis of resource consumption of embedded systems, in which also time is a relevant factor. If possible, the project should – in addition to its theoretical contribution – produce prototype implementations, allowing further evaluation of produced theoretical results, through industrial (or artificial) cases.

1 Introduction

Software is becoming an integral part of everyday life for most of us, whether we know it or not. Moreover, it is a fact that from time to time, we experience software related problems which may be more or less obvious to us as users. Although annoyed by the fact, many have accepted that most software is inherently flawed, and that sometimes the only solution is to power off the failed device. Obviously, this a solution may be unaffordable for large industrial systems and truly unacceptable for mission- and safety critical systems. One of the classic examples of software errors is the crash of Ariane 5 Flight 501 [Dow97], a rocket, which was to deliver 4 satellites to orbit the Earth. The crash resulted in a loss of more than US$370 million. In order to avoid such errors, testing may be used verify that a system operates correctly, by "exercising" the software, in a way makes the presence of errors is very unlikely. Additionally formal verification, based on sound mathematical theory, may be used to prove certain properties of a system (or its design). The latter will be the focus of this Ph.D. project.

1.1 Formal methods and Verification

Formal methods of verification [EMCP00, AILS07], denotes the application of mathematical analyses on formal representations of systems i.e. models. In contrast to informal system specifications (descriptions), widely used in software engineering, a formal representation of system, enjoys the benefit of being unambiguous. This is achieved, since a precise semantic is associated with the respective modeling formalism, disallowing subjective interpretations. Using appropriate formalisms, modeling enables us to apply mathematical techniques to prove certain properties of a system e.g. that *deadlocks*, which may appear as inexplicable system freezes, do not occur due to an error in the system's design.

Although manual application of formal methods, may require substantial effort, it has proven highly beneficial for verification of complicated, often highly parallelized, systems such as embedded systems. Moreover the continuous development of tools such as Spin [Hol97], UPPAAL [LPY97] and VisiualState [Sys08] allows system designers to create (or generate) models which may be verified automatically. It is therefor becoming increasingly feasible that formal verification will find its way to mainstream engineering projects.

1.2 Research Direction

Until recently, the primary effort in the (verification) research community, has been given to the development of theory and tools which would either prove or refute a property of a system, i.e. computing yes/no answers. Although a necessary first step, notions such as *robustness* and *closeness* has seen increasing interest in the community as they provide more information about the correctness of a system. That is, robustness is a term describing verification of systems which may deviate, to some acceptable degree, from the specification, while still satisfying the desired property. Similarly, closeness may be used to describe the fact that systems (models) are not only considered to be correct or not, but it is given some measure of how close, it is to satisfy some property. This allows designers to determine to what extend improvements of the model is required.

The proposed Ph.D. project and scientific research, is inspired by *The Embedded Systems Design Challenge* [HS06], in which Thomas A. Henzinger and Joseph Sifakis calls for a coherent scientific foundation for embedded systems design, where qualitative and quantitative analysis is one of many focus points. The motivation behind this project is to further development of formal methods of verification for mission critical systems with respect to quantitative analysis, including robustness and closeness. It is assumed that the project will extend the work by Henzinger et. al. in [HMP05] on *Quantifying similarities between timed systems* and Luca de Alfaro et al. in [dAFS04] on *linear and branching timed metrics* as well as KGL and my self in [Thr08].

2 Project Proposal

It is my hypothesis that in order to construct a new foundation for systems design, we must revisit early models of systems, such as automata and transition systems, in order to extend these to a general framework incorporating quantities.

The research which I propose for this project may be categorized, and motivated, as follows:

2.1 Quantitative Properties of Systems

The term "quantitative properties" should be interpreted here in a very broad sense, it is intended as a reference to any property of an object which we may "measure". Intuitively, one may think of such things as *power consumption* and *time requirement* or any resource consumption in general, of an object such as a watch, car or space craft. Moreover, quantitative properties may refer to any relationship; influence or dependency of individual quantities on each other, here one might think of milage of a vehicle with respect to speed and fuel consumption.

In order to verify quantitative properties for software or hardware systems, we must carefully consider the formalisms which we use to express our models. Existing formalisms, such as *Timed Automata* (TA) presented in [ACD90], Weighted Automata, from [DR07] and Probabilistic Transition Systems [HK97] already allow modeling of quantitative properties such as temporal requirements, weights/cost and probabilities respectively and all exhibit nice[1] properties for algorithmic analysis. Moreover all of these, has successfully been applied for modeling large complex systems. But none of the above support verification of properties related to combining quantitative information e.g. both time and cost.

The search for such extensions of formalisms, with combined quantitative information, has seen increased focus over recent years. TA extended with cost, presented in [LBB$^+$01], referred to as *Priced Timed Automata* (PTA or WTA) enables the analysis of resource consumption, for finitely and infinitely running, timing critical systems. However, such extensions impose problems on algorithmic analysis. In the case of PTA it has been shown that the reachability problem becomes undecidable for systems with 3 or more clocks but decidable for one clock [BLM07]. In this project I propose further research of such formalisms, with the goal of creating a general framework for modeling and the analysis of combined quantitative properties.

2.2 Analysis: Robustness, Closeness and Continuity of composition

In order to determine how closely a property is satisfied by the system under analysis, the project must include research into relevant metrics, equalities, equations and their properties, including computability and complexity. Additionally, the design of quantifying logics and the use of discounting techniques should be investigated along with any possible relationships of logics and behavioral relations. Models are often constructed by composing existing subsystems, or defined in terms of its components - thus the project should investigate the continuity of quantitative properties w.r.t. the classical compositional

[1] Although the problem of model-checking of e.g. Timed Automata has been shown to be **PSPACE**-complete, it is nevertheless decidable.

operators (e.g. parallel composition). From a practical point of view, this may lighten the remaining proof obligation, whenever correctness has been shown for (some of) the system's components. Moreover, in the case of large systems, a complete model may be inconvenient to construct due to the combined complexity of components.

2.3 Expected Results - Significance and Applications

As intuitively described above, it is currently expected that this project will produce, or contribute to, a coherent way of modeling systems with multiple quantitative properties of interest. Proposed formalisms should exhibit good properties for efficient analysis of quantitative properties in the sense described above. In the this case, produced results may be applied directly to analysis of real-world problems, possibly through the development of prototype tools.

2.4 Method

The methods to be used are the standard ones in theoretical computer science: Development of formalisms for modeling, showing usefulness of a formalism by arguing for its applicability on one hand, and proving useful results such as decidability of certain important properties and complexity results on the other hand, showing evidence of achievable results by experiment, and possibly provide a proof-of-concept implementation. Formalisms used in the project will mostly be automata-based, though some attention will be spent on developing associate logics useful for model-checking.

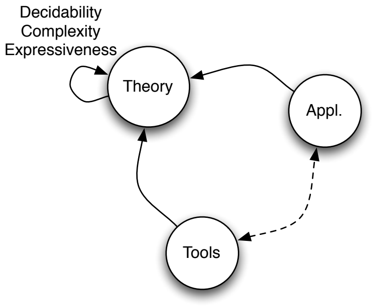

Figure 1: Abstract workflow

Moreover, it is expected that a number of relevant cases will be explored in order to further validate any theoretical claims. Fig 1 illustrates that this fact, as well as emphasises the theoretical focus of the project, where the use of cases and tool experiments is used to further the development of theoretical results.

3 Thesis Outline

The Ph.D. thesis is going to be organized as a plurality of of scientific papers.

3.1 Summary of Research topics (Objectives)

The following lists a number of candidate research directions, as well as possible paper titles. Titles are listed by the expected progression of developed theory. Proposed papers are expected to of standard length (approx. 12 pages) and will likely be submitted to one of the events listed in Section 6.2.1.

- The formulation of quantitative formalisms, including a survey of existing models.
- Behavioral and Quantitative Relations (metrics), and their Computability.
- Continuity of Quantitative Relations for WLTS.
- Quantifying Infinite Behavior of WLTS and WTA.
- Applying Discounting for finite and infinite state Quantitative Systems.

- Algorithms for model-checking of Quantitative logics.
- A Prototype Tool for Analysis of Quantitative Properties.

The papers will be bound together by an introduction and conclusion, that highlights the similarities and differences between different techniques and theory.

4 Provisional Time Schedule

The project is intended to follow the standard three-year duration of a Ph.D scholarship, divided into six modules each comprising a standard academic (Fall or Spring) semester. Each semester is expected to produce 1 - 2 publications, cf. the progression outlined in Section 3.1. All teaching obligations will commence, in accordance with Table 2 below. As courses (See Table 1) has not yet been announced, they cannot be scheduled yet.

F08 Effort should be spent on investigation of the related field of *Weighted Automata* and relationship between WLTS and PTA. Moreover any remaining unresolved issue in the existing WLTS work should be addressed, and a paper should be submitted to a QAPL'09 as well as presented at NWPT'08. Finally the course on *complexity theory* at AU is taken.

S09 Continue initiated work research on quantitative formalisms and metrics. Research on computability and complexity of metrics on quantitative models should be completed and published. It is expected that all joint courses is to be taken, as well as the remaining study course.

F09 I expect that the primary focus to be on logics and model-checking, possibly including a prototype implementation of devised algorithms.

S10 While possibly visiting a relevant research group at another university, focus should be given to composition of systems (and their proofs) and the extension of existing work to infinite state systems. Produced results should be published.

F10 Finish and submit any unfinished papers, on logics and composition of proofs. At this point, a sufficient amount of papers should be completed for the thesis.

S11 The completion of my Ph.D. thesis and its defense.

5 Collaboration

5.1 Supervision

The project is supervised by Prof. Dr. Kim Guldstrand Larsen. Expected cooperation include collaborating on papers and discussions of the scientific issues in the process. Meeting frequency is expected to be weekly if possible, throughout most of the project period.

5.2 Internal Collaboration

In addition to any collaboration with KGL, both Jiří Srba and Ulrich Fahrenberg work within relevant areas and have expressed interest in doing joint work.

5.3 External Collaboration

Two different opportunities are currently present for an international stay and external collaboration. Through KGL, it would be possible to visit The School of Computer and Communication Sciences at EPFL, which would allow me to collaborate with Thomas Henzinger. Alternatively, it would be possible to visit University of Pennsylvania and Oleg Sokolsky, who also works on verification and quantitative systems such as weighted automata.

6 Practical Considerations

6.1 Courses adding up to 30 ECTS

At the current time, it is not possible to predict exactly which project-related courses will be available throughout the duration of the project. An appropriate selection of courses for a Ph.D scholarship includes a combination of joint and project-related courses, as well as advanced theoretical study courses. Table 1 sums up a preliminary overview of expected credit giving activities as well as their type, i.e. (S)tudy course, (J)oint Course and (P)roject related course or activity.

Table 1: Course credits and association

ECTS	**Type**	**Activity**	**Organizer**
6.00	S	Computational Complexity Theory	K. Arnsfelt Hansen (AU)
3.00	S	Static Analysis	MT-Lab / F. Nielsen (ITU)
2.50	J	Management of Research and Development	Prof. Frank Gertsen
3.75	J	Writing and Reviewing Scientific Papers	Prof. Jakob Stoustrup
2.00	P	Reviews of (10) scientific papers	KGL (and DES members)
3.00	P	Model-Checking	MT-Lab / KGL
2.00	P	Linear Programming or Combinatorial Search	MT-Lab
4.00	P	Marktoberdorf Summer School	Univ. Munich & NATO
4.00	P	Workshops and Conferences	N/A
30.25			**Total** (currently)

By the current plan this gives 15.25 ECTS in joint and study courses as well as, 15 ECTS in projects courses and activities. There are currently no plans for participating in study groups.

6.2 Teaching obligations and Dissemination

Scientific results achieved throughout the Ph.D. project will be disseminated to the community primarily by two means: By publishing papers in journals and conference proceedings, and by giving presentations at conferences and workshops as well as the local research unit. Additionally, knowledge will be communicated through lecturing and student supervision. Table 2 summarises how hours for fulfillment of teaching obligations are expected to be distributed.

Table 2: Teaching

Hours	**Activities**	**Responsibility**	**Sem.**
70	*Into to Java*	Lecture	1
75	*Intro to Distributed Systems*	Assistant	
75	*Models and Tools for Parallelism*	Assistant	2
70	*Into to Java*	Lecture	3
216	*Group supervision - x 2*	Supervisor	
0	Abroad	–	4
70	*Into to Java*	Lecture	5
216	*Group supervision - x 2*	Supervisor	
75	*Models and Tools for Parallelism*	Assistant	6
867	from 435 (teaching) + 423 (supervision)		**Total**

Hours for group supervision, are based on typical numbers, i.e. groups of 6 students, amounting to 108 hours pr group.

6.2.1 Workshop and Conferences

While it is not possible at this point to predict precisely which conferences will have relevant workshops in the three-year period, the project falls within the general area of formal methods and verification, subjects traditionally covered at the following events:

- AV - The workshop Automata and Verification.
- NWPT - Nordic Workshop on Programming Theory.
- QAPL - Workshop on Quantitative Aspects of Programming Languages.
- CONCUR - International Conference on Concurrency Theory.
- FOSSACS - Foundations Of Software Science And Computation Structures.
- TACAS - Tools And Algorithms For The Construction And Analysis Of Systems.
- FORMATS - Formal Modelling and Analysis of Timed Systems.
- QEST - Quantitative Evaluation of SysTems.
- CAV - Computer Aided Verification.
- ICALP - International Colloquium on Automata, Languages and Programming.
- EXPRESS - Expressiveness in Concurrency.
- INFINITY - International Workshop on Verification of Infinite-State Systems.
- SOFSEM - Current Trends in Theory and Practice of Computer Science.

6.3 Financing

The project will be financed jointly, such that 2 years work is financed by *Quasimodo* (EU Seventh Framework Programme) and 1 years joint financing by *Dept. of Computer Science* and *The Faculties of Engineering, Science and Medicine.*

6.4 Copyrights and Patents

All rights to intellectual property, are governed by standard regulations at AAU. Hopefully, any software produced in the project will be released under a permissive free software license, such as the BSD or MIT licenses. This allows the general computer science research community to study and adapt the software freely, and allows it to be adapted for other academic and/or industrial uses. It is not intended that any software developed during the project will be patented; firstly, the legality of software patents is currently controversial in the European Union, and secondly, patented software may present legal hindrances for other researchers in using the technologies developed.

References

[ACD90] Alur, Courcoubetis, and Dill. Model-checking for real-time systems. In *LICS: IEEE Symposium on Logic in Computer Science*, 1990.

[AILS07] Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri Srba. *Reactive Systems: Modelling, Specification and Verification.* Cambridge University Press, 2007.

[BLM07] Patricia Bouyer, Kim Guldstrand Larsen, and Nicolas Markey. Model-checking one-clock priced timed automata. In *FoSSaCS*, pages 108–122, 2007.

[dAFS04] L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching metrics for quantitative transition systems, 2004.

[Dow97] Mark Dowson. The ariane 5 software failure. *SIGSOFT Softw. Eng. Notes*, 22(2):84, 1997.

[DR07] Manfred Droste and George Rahonis. Weighted automata and weighted logics with discounting. In *CIAA*, pages 73–84, 2007.

[EMCP00] O. Grumberg E. M. Clarke and D. A. Peled. *Model Checking.* MIT Press, 2000.

[HK97] Michael Huth and Marta Z. Kwiatkowska. Quantitative analysis and model checking. In *Logic in Computer Science*, pages 111–122, 1997.

[HMP05] Thomas A. Henzinger, Rupak Majumdar, and Vinayak S. Prabhu. Quantifying similarities between timed systems. In Paul Pettersson and Wang Yi, editors, *FORMATS*, volume 3829 of *Lecture Notes in Computer Science*, pages 226–241. Springer, 2005.

[Hol97] Gerard J. Holzmann. The spin model-checker. *In Proceeding FORTE 1999*, 28:481–497, 1997.

[HS06] Thomas A. Henzinger and Joseph Sifakis. The embedded systems design challenge. In *14th International Symposium on Formal Methods (FM)*, Lecture Notes in Computer Science, pages 1–15. Springer, September 2006.

[LBB$^+$01] Kim Larsen, Gerd Behrmann, Ed Brinksma, Ansgar Fehnker, Thomas Hune, Paul Pettersson, and Judi Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. *Lecture Notes in Computer Science*, 2102:493+, 2001.

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997.

[Sys08] IAR Systems. Visiualstate, Sep 2008. `http://www.iar.com/website1/1.0.1.0/371/1/`.

[Thr08] Claus Thrane. On weighted labelled transition systems - quantative relations and logic. Technical report, Distributed and Embedded Systems unit AAU, 2008.