Proceedings of the 10th Real-Time Systems Open Problems Seminar (RTSOPS 2019)

INRIA, Paris, France

July 24, 2019

Organized by:

Liliana Cucu-Grosjean, INRIA Paris, France

Robert Davis, University of York, UK

Nathan Fisher, Wayne State University, USA

Technical Program

9:00 Session 1 (Pages 3-5)

Introduction
Rob Davis and Nathan Fisher

Real-Time Communication over Low-Power Wide-Area Network: Challenges and Directions
Abusayeed Saifullah

From Java to Real-Time Java: A Model-Driven Methodology with Automated Toolchain
Wanli Chang, Shuai Zhao, Ran Wei, Andy Wellings, Alan Burns

System-Wide Power Management for Real-Time Systems
Roberto Medina and Liliana Cucu-Grosjean

Can the RUN Scheduling Algorithm go Beyond Periodic Task Models?
George Lima

10:30 Coffee and Cake break
11:00 Session 2 (Pages 6-7)

Towards an Automated, Efficient, and Accurate Schedulability Analysis for Real-Time Cyber-Physical Systems
Mitra Nastri

On Verification and Synthesis of Time-Delay Systems
Naijun Zhan

Validation of Statistical Timing Models of a Periodic Task on a Microcontroller
Anna Friebe, Alessandro V. Papadopoulos, and Thomas Nolte

12:00 Coffee break
12:30 Session 3 (Page 8)

Can we Synthesize Resource Allocation Policies from Examples?
Sathish Gopalakrishnan and Theepan Moorthy

On the verification of autonomous systems and the role of real-time research
Bjorn Andersson and Dionisio de Niz

Open and collaborative classification of RTSS papers
Enrico Bini

13:30 Lunch
15:00 Session 4 (Pages 9-10)

A Multi-Dimensional Adaptive Variable Rate Task Model and Its Potential Role in Reducing Resource Utilization of Embedded Systems
Tam Chantem and Nathan Fisher

Programming Language Support for Time-Sensitive CPS
Aviral Shrivastava

Micro-Architectural Attacks on Cyber-Physical Systems
Heechul Yun

16:00 Coffee break
16:30 Session 5 (Pages 11-12)

On Beyond Time: Managing Cyber-Physical Inter-Dependence and Interference of Real-Time Tasks
Chris Gill

Open Problem Space for Real-Time CPS Software Engineering Research
Qixin Wang

Computing Request/Demand Bound Functions for Task Automata
Nan Guan

Wrap-up
Rob Davis and Nathan Fisher

17:40 Approx. finish.

COLLECTION OF ABSTRACTS

Session 1

Real-Time Communication over Low-Power Wide-Area Network: Challenges and Directions

Abusayeed Saifullah

The evolution of Internet of Things (IoT) is transforming the field of industrial automation including smart manufacturing, remote health care, process control, and smart farming in the form of Industrial IoT, Cyber-physical Systems (CPS) or, more generically, Industry 4.0. Today, industrial IoT and CPS are emerging in large-scale and wide-area applications (e.g., oil-field management, and smart farming) that may spread over hundreds of square kms (e.g., 74x8km^2 East Texas Oil-field), requiring tens of thousands of sensors for automated management. Recently proposed Low-Power Wide-Area Networks (LPWANs) appear as a promising technology to meet this impending demand in industrial automation. LPWAN enables low-power (milliwatts) wireless devices to transmit at low data rates (kbps) over long distances (kms) using narrowband (kHz). It promises to overcome the range limits and scalability challenges in traditional wireless sensor networks, obviating the need of multihop and allowing the devices to directly communicate with the control room (gateway).

The fundamental building blocks of any industrial automation system are feedback control loops that largely rely on real-time communication. In this talk, we shall outline the challenges for enabling real-time communication over LPWANs. The challenges stem from their new characteristics and requirements. For example, LPWANs have been mostly explored for uplink communication (devices to gateway). Due to severe energy-constraint, many LPWANs usually adopt a simple media access control protocol based on ALOHA with no collision avoidance that is naturally unsuited for real-time communication. Some (e.g., LoRa and SigFox) are also required to adopt an extremely low duty-cycling (e.g., 0.1%, 1%, 10%) in some regions (e.g., Europe). As the devices are extremely energy-constrained, their communications have to be at minimal which make real-time control highly challenging. This talk will also outline some possible directions to address the challenges.

From Java to Real-Time Java: A Model-Driven Methodology with Automated Toolchain

Wanli Chang, Shuai Zhao, Ran Wei, Andy Wellings, Alan Burns

Real-time systems are receiving increasing attention with the emerging application scenarios that are safety-critical, complex in functionality, high on timing-related performance requirements, and cost-sensitive, such as autonomous vehicles. Development of real-time systems is error-prone and highly dependent on the sophisticated domain expertise, making it a costly process. This talk discusses a new methodology that utilises the principles of model-driven engineering (MDE) and automatically converts standard time-sharing Java applications to real-time Java applications. It opens up a new research direction on development automation of real-time programming languages and inspires many research questions that can be jointly investigated by the embedded systems, programming languages as well as MDE communities.

System-Wide Power Management for Real-time Systems

Roberto Medina and Liliana Cucu-Grosjean

During the last decade, real-time systems have been confronted to trends aiming at integrating an increasing number of services into embedded platforms while keeping size, weight and power-consumption to a minimum. In this talk we are particularly interested in the power-consumption of real-time embedded systems. In fact, efficient energy management is crucial for all battery-powered real-time embedded systems, such as those deployed in drones, autonomous robots, wearable devices, and so on. A widely used technique for reducing power-consumption on real-time embedded systems is Dynamic Voltage and Frequency Scaling (DVFS) on processors. DVFS approaches decrease the voltage and frequency of the processor to reduce energy consumption. By doing so, tasks execution time increases but DVFS approaches derive processor speed values that still guarantee real-time timing constraints. We will show (i) limits on existing DVFS power models and implementations, and (ii) argue that DVFS could also be applied to other components like system's memory furthering power reduction.

Can the RUN Scheduling Algorithm go Beyond Periodic Task Models?

George Lima

RUN optimally schedules sets of implicit-deadline periodic tasks on identical multiprocessors [1]. Optimality is desirable in the sense that optimal algorithms generate a schedule with no missed deadline whenever it is possible to do so. Other good characteristics of RUN are associated with the low overheads in terms of preemption and migration it imposes. Since its publication, the algorithm has been extended for incorporating several application needs such as resource sharing [2], energy-savings [3] or mixed-criticality [4]. Despite these advances, the considered models have been restricted to periodic tasks. The only known RUN extension capable of dealing with sporadic tasks sacrifices optimality [5].

Unlike other multiprocessor scheduling algorithms for real-time systems, RUN achieves optimality by translating a uniprocessor schedule into a schedule for multiprocessors. A key concept in this mapping is the duality principle, which states that scheduling periodic tasks is equivalent to scheduling their slack time, called dual tasks in RUN terminology. In a nutshell, let T be a task with period P requesting C time units ($C < P$). Its dual task T^* is defined as requesting $C^* = P-C$ time units during the same intervals of size P, namely $w(k) = [kP, (k+1)P)$, $k=0,1,2,\ldots$. Assuming that the schedule for T^* is known, then finding the instants that T should execute in w(k) is straightforward by considering that they are complementary to each other: whenever T^* executes T should not execute and vice-versa. This means that the schedule of a set of duals can be used to efficiently derive the schedule for the respective primal tasks. RUN provides a set of transformations, carried out off-line, capable of generating a system schedulable in uniprocessors from a schedulable multiprocessor system. Then, by applying the duality principle on-line, scheduling decisions for the former, generated by EDF, are efficiently translated back to the original system.

Paradoxically, the duality principle lies in the heart of the difficulties in extending RUN to cope with sporadic task loads. As sporadic tasks do not release jobs regularly, the windows of type w(k), as specified above, are not well defined. If T was a sporadic task, then there could be intervals during which both T and T^* do not exist, making the duality-equivalence hard to be applied. Adapting RUN to deal with sporadic task loads without compromising its good characteristics has been shown challenging and is still a non-solved problem.

References.

[1] P. Regnier, G. Lima, E. Massa, G. Levin, and S. Brandt, “RUN: Optimal Multiprocessor Real-Time Scheduling via Reduction to Uniprocessor”. 32th IEEE Real-Time Systems Symposium, Vienna, pp. 104–115, 2011.

[2] L. Bonato, E. Mezzeti and T. Vardanega, “Supporting Global Resource Sharing in RUN-Scheduled Multiprocessor Systems,” International Conference on Real-Time Networks and Systems. Versaille, France, pp. 109–118, 2014.

[3] H. Chishiro, M. Takasu, R. Ueda and N. Yamasaki, "Optimal Multiprocessor Real-Time Scheduling Based on RUN with Voltage and Frequency Scaling". IEEE 18th International Symposium on Real-Time Distributed Computing, Auckland, pp. 284-287, 2015.

[4] R. Gratia, T. Robert and L. Pautet, "Scheduling of mixed-criticality systems with RUN". In the Proc. of theIEEE 20th Conference on Emerging Technologies & Factory Automation, Luxembourg, pp. 1-8, 2015.

Session 2

Towards an Automated, Efficient, and Accurate Schedulability Analysis for Real-Time Cyber-Physical Systems

Mitra Nasri

With the increasing complexity of software and hardware components used in real-time cyber-physical systems (CPS), the gap between the state of the art on schedulability analysis and what is actually needed by industry is rapidly growing. In our recent research, we have focused on closing this gap by designing an efficient and accurate schedulability-analysis framework to derive the worst-case response time for various scheduling policies (both for uniprocessor and multiprocessor systems) and workload models. The analysis explores the space of possible execution scenarios (and schedules) in the presence of uncertainties on the processes' execution and arrival times. In this talk, I will introduce the framework, show how it solves some of the existing schedulability problems, and discuss some of the open research directions.

On Verification and Synthesis of Time-Delay Systems

Naijun Zhan

Conventional real-time embedded systems have over the past two decades evolved into an open, interconnected form, now known as cyber-physical systems. The advent of systems of cooperative cyber-physical systems draws attention to a central problem of networked and distributed control systems: the ubiquity delay in feedback loops between logically or spatially distributed components, which is not adequately reflected in traditional models of hybrid-state dynamics based on ordinary differential equations and immediate transitions. The occurrence of feedback delays may significantly alter a system's dynamic response. Unmodelled delays in a control loop consequently have the potential to invalidate any stability or safety certificate obtained on a related delay-free model, which is the current practice in hybrid-system analysis.

Validation of Statistical Timing Models of a Periodic Task on a Microcontroller

Anna Friebe, Alessandro V. Papadopoulos, Thomas Nolte

Probabilistic timing analyses have been proposed for use in soft real-time systems, to remedy the problem that deterministic estimates of the task's Worst-Case Execution Time and Worst-Case Response-Time are overly pessimistic. Often assumptions are made that a task's probability distributions of response times and execution times are independent of other tasks. It is clear that in real systems, this assumption is often violated. In this paper, we analyze the timing behaviour of a simple periodic task on a microcontroller, a Raspberry Pi model 3 running Arch Linux ARM. In particular, we observe and analyze the distributions of wake-up latencies and execution times for the sequential jobs released by the periodic task. We observe that the timing behaviour of jobs is affected by release events during the job's execution time, and of other processes running in between subsequent jobs of the periodic task.

Using a data consistency approach we investigate whether it is reasonable to model the timing distribution of jobs affected by release events and intermediate processes as translations of the empirical timing distribution of non-affected jobs. According to the analysis, a translated distribution model of non-affected jobs is improbable for the execution time distribution of jobs affected by intermediate processes. Regarding the latency distribution with intermediate processes and the execution time distribution affected by release events during execution, no certain conclusions can be drawn.

Session 3

Can we Synthesize Resource Allocation Policies from Examples?

Sathish Gopalakrishnan and Theepan Moorthy

Scheduling policies for real-time systems are often developed using intuition about the scheduling problem at hand. Such policies are then analyzed to understand the guarantees they can provide. For scheduling problems where the requirements are statistical, it is harder to identify such policies up front. Would it be possible to use examples of scheduling decisions for sample workloads to then synthesize good policies? Given the recent work in example-driven program synthesis, I speculate that this approach may indeed be feasible and effective for some scheduling/resource allocation problems.

On the Verification of Autonomous Systems and the Role of Real-Time Research

Bjorn Andersson and Dionisio de Niz

Autonomous systems are all rage now and it is worth asking what could be the role of real-time systems research in this area. We discuss some requirements and ideas for solutions. Specifically, we observe that there is a cultural difference between embedded-safety-critical-culture-with-C-programmning and Silicon-valley-move-fast-and-break-things-culture-with-python-and-javascript. On the one hand, we need small operating systems kernels that can be formally verified. On the other hand, we need lots of functionality, e.g. GPU drivers and Python run-time for certain types of machine learning. Also, creating a plan that assumes the worst-case behavior of the physical environment makes autonomous systems very slow; thus, planning for the average case and reacting to bad situations may be better than planning for the worst.

Open and Collaborative Classification of RTSS Papers

Enrico Bini

Oftentimes we find ourselves describing to colleagues what the real-time systems community investigates about. Not an easy task since timing is affected by all layers of abstractions: from the architecture up to languages. Also, the aging of researchers tends to limit the scientific memory to (say) 10 or 20 years. This may lead to the well-known phenomenon of periodic "reinvention of the wheel".

In this presentation, it will be illustrated an effort made to classify all real-time papers. Papers ever published at RTSS are used as hopefully representative sample of the research community interests. Believing that claiming the authorship of such an effort will be an obstacle to contributions, this work is made publicly available on github. Contributions are more than welcome.

Session 4

A Multi-Dimensional Adaptive Variable Rate Task Model and Its Potential Role in Reducing Resource Utilization of Embedded Systems

Tam Chantem and Nathan Fisher

We consider adaptive variable rate (AVR) tasks which may change their execution time or inter-arrival rates based on the physical state of the system. For instance, an engine-control task may execute a different set of functions based on the angular velocity of the crankshaft for automotive engine control. Depending upon the RPM, the engine-control task might require different execution times. Furthermore, since this task is invoked upon every revolution of the crankshaft, the inter-arrival rate between invocations of the task is also dependent upon the RPM. Current research in AVR tasks have sought to obtain schedulability conditions under simple single-dimensional dependencies, e.g., guarantee the timeliness of an AVR task on an ECU when it depends only upon the crankshaft velocity, and fails to capture the fundamental properties of many AVR tasks: the physical context that determines a task's timing parameters is often complex and multidimensional; furthermore, these physical dimensions are often interdependent. For instance, the optimal engine timing for fuel efficiency and emission reduction is often a complex function of crankshaft angular velocity and engine temperature. Therefore, the execution time of an engine-control task might be dependent upon both the angular velocity of the crankshaft and current engine temperature. Furthermore, the value of the physical parameters might be interdependent -- e.g., engine temperature is affected by the crankshaft -- and are likely even predictable in the sense that relationship between the parameters follow some well-established physical laws.

An open problem is to develop a multi-dimensional AVR task model that exploits the multidimensional physical dynamics of the system to make predictions and extract dependencies in the system's computational demand. We believe that doing so will improve the utilization of the computational resources (reducing the size, weight, and power requirements), integration of multiple subsystems on a shared platform, and the resiliency of systems comprising AVR tasks. To solve this open problem, new theoretical models and analytical tools are required to understand and leverage the dependencies between the underlying physical dynamics of the system and the computational timing parameters of the control system. New methods to quantify the computational demand of these dependencies on the system under consideration must also be developed.

Programming Language Support for Time-Sensitive CPS

Aviral Shrivastava

Distributed Cyber-physical Systems (CPS) have timing requirements that must be met. Writing programs that satisfy the desired timing constraints is challenging because modern programming languages do not provide a way to express the timing requirements of an application. Making an application meet some timing requirements is quite an elaborate task, in which essentially the programmer is completely responsible for developing an implementation that will meet the timing requirements. It is the programmer's responsibility to meet the execution time specification of a routine. It is the programmer's responsibility to setup the interrupts, and service the interrupts. It is the programmer's responsibility to setup the timer counts. It is the programmer's responsibility to synchronize the time between the components of a distributed CPS. This becomes especially hard in IoT systems, where the components can be from different manufacturers, have different hardware components, and different software libraries. In this talk, I will set the stage for our NSF TickTalk project -- which aims to define a high-level API to specify the timing requirements of a distributed CPS, so that the developer only has to specify the timing requirements that must be met, and then the compiler and the libraries take on the task of implementing the timing requirements.

Micro-Architectural Attacks on Cyber-Physical Systems

Heechul Yun

Microarchitectural attacks are a class of software attacks that targets hardware. Modern high-performance computing hardware employs a variety of sophisticated microarchitectural components---multiple levels of caches, prefetchers, out-of-order speculative out-of-order execution engine, etc.---to improve performance. Microarchitectural attacks target weaknesses in these microarchitectural components and many kinds of successful attacks---which leak secret, alter data, and delay execution times of the victim---have been demonstrated in recent years. As safety-critical cyber-physical systems (CPS) are increasingly relying on high-performance hardware, microarchitectural attacks are becoming an important and serious threat to their safety and security. In this talk, I will present examples of microarchitectural attacks in the context of CPS and discuss the challenges and potential solution approaches to defend against these attacks.

Session 5

On Beyond Time: Managing Cyber-Physical Inter-Dependence and Interference of Real-Time Tasks

Christopher Gill

Modern operating systems and concurrency platforms provide policies and mechanisms that can be used to enforce timing constraints on application tasks, even when tasks have inter-dependent sub-tasks that may interfere with each other or with other tasks unless managed precisely according to rigorously defined scheduling techniques. However, the abstractions that many such platforms provide, and the complexity and cost of using them successfully even for timing assurance, are increasingly ill-suited for cutting edge cyber-physical systems with significant degrees of interdependence, dynamic behavior, and stringent constraints all (1) involving other semantics as well as timing, and (2) spanning both cyber- and physical domains.

This talk will survey a couple of recent examples from the real-time systems and cyber-physical systems literature that serve to illustrate these challenges, and will discuss a few current and envisioned research directions that appear promising towards addressing them. Together such efforts have significant potential to expand the real-time notion of non-interference to support timing guarantees, to address additional semantics for control and security in next-generation cyber-physical systems.

Open Problem Space for Real-Time CPS Software Engineering Research

Qixin Wang

Real-Time Cyber-Physical Systems (CPS) are the results of the inevitable merge of computers with other domains of physical-world practices. A key challenge to CPS is software engineering: how to coordinate experts of disparate backgrounds to build dependable, maintainable, and efficient complex CPS. This challenge is exacerbated as each cyber/physical domain advances, resulting in combinatorial growth of the cross-domain interaction/interference complexity.

The CPS software engineering challenge implies a huge problem space involving multiple dimensions. In this talk, we aim to introduce this problem space to a broad audience, to reveal its abundant academic and practical values, and to call for joint effort to address the challenge of CPS software engineering.

Specifically, in this talk, we shall explore the CPS software engineering problem space with several concrete examples. Through these examples, we shall demonstrate how cross-domain thinking can help address/constrain various CPS software engineering problems, particularly those involving real-time issues.

Computing Request/Demand Bound Functions for Task Automata

Nan Guan

Timed Automata and the extension Task Automata are expressive models to describe complex real-time task systems, but their analysis problems are computationally intractable. On the other hand, in real-time scheduling theory people usually represent real-time task systems by simple models (e.g., the periodic and sporadic task models), for which the highly efficient abstractions (e.g., the Request/Demand Bound Functions) and analysis techniques (e.g., busy-period analysis) have been developed to check system schedulability. We are interested in borrowing the efficient abstractions and analysis techniques in real-time scheduling theory to improve the analysis efficiency for complex real-time task systems modelled by Timed/Task Automata. As the first step, we ask the following question: can we efficiently compute the Request/Demand Bound Functions for an individual Task Automaton? In this talk, I will briefly introduce the semantics of Timed/Task Automata, formalize the questioned problem, and provide some background of existing abstractions used in their state-space exploration.