Keynote 1

The Role of models@run.time in Autonomic Systems

Dr. Nelly Bencomo, Aston University, UK

Abstract

Autonomic systems manage their own behaviour in accordance with high-level goals. This talk will present a brief outline of challenges related to Autonomic Computing due to uncertainty in the operational environments, and the role that models@run.time play in meeting them. I will argue that the existing progress in Autonomic Computing should be furtherexploited with the support of runtime models. I will discuss ideas and current research results related to the need to understand the extent to which the high-level goals of the autonomic system are being satisfied to support decision-making based on runtime evidence and, the need to support self-explanation.

Biography

Dr. Nelly Bencomo is a lecturer (assistant professor) in the School of Engineering and Applied Science at Aston University in the UK. She is interested in all aspects of software modelling and specially the application of model-driven techniques, during the development and operation of dynamically adaptive and highly distributed systems. Previously, she was an European Marie Curie Fellow at INRIA Paris (2011-2013). Her Marie Curie project was called Requirements-aware systems and was applied in the context of self-adaptive systems (nickname: Requirements@run.time). With other colleagues, she coined the research topic models@run.time, co-founding the workshop series Models@run.time (which is running since 2006). She has built a portfolio of successful collaborative research projects in areas such as decision-making under uncertainty and distributed, self-adaptive and autonomous systems where she has successfully applied innovative software engineering techniques. Lately, she has focused on the quantification of uncertainty and the use of Bayesian learning to support decision-making for self-adaptation. This work led her to co-found the International Workshop on Artificial Intelligence for Requirements Engineering (AIRE). She seeks to apply software engineering on real-world and multidisciplinary applications and, that has taken her to work in research labs that are not inherently related to software engineering, such as the ARLES team in Inria- France(a lab on software architecture and distributed systems), the reflective middleware group in Lancaster University (UK), a math models Lab in Venezuela and, more recently, the research lab ALICE at Aston (the Aston Lab for Intelligent Collectives Engineering).

She is member of the steering committee of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), and serves or has served on the Program Committee for the Requirement Engineering Conference, MODELS Conference and ICSE-2018 among others. She was the Technical Program Chair for SEAMS in 2014.
Keynote 2
Columbus Smart Cities Initiative

Mr. Mark Patton, Vice President of the Columbus Smart Cities Initiative, USA

Biography
Mark Patton is the Vice President at the Partnership, focusing on the Smart Cities initiative. Prior to joining the Partnership, Mark was the President of FactGem, a data analytics platform for customer intelligence. Mark is a native of the West Coast and moved to Ohio 6 years ago to help establish JobsOhio, the states private economic development organization, where he was the Senior Managing Director, focused on several key industries including: information technology, consumer products, headquarters and logistics. Earlier, Patton held senior roles in both sales and marketing organizations with corporations, including Procter & Gamble, Apple Computer and Eastman Kodak. Prior to moving to Ohio, Mark spent the previous 15 years leading technology start-up companies in Silicon Valley. Mark received his bachelor’s degree from the University of Washington and completed a marketing management program at Stanford University.
Keynote 3

Connected and Automated Vehicles: The Future of Surface Transportation?

Prof. Ragunathan Rajkumar, Carnegie Mellon University, USA

Biography

His research interests lie in all aspects of embedded real-time systems and wireless/sensor networks. In the domain of embedded real-time systems, his interests include but are not limited to operating systems, scheduling theory, resource management, wired/wireless networking protocols, quality of service management, hardware/software architecture, model-based design tools and power management. In the context of wireless/sensor networks, his research interests span hardware, devices, power-efficient networking protocols, run-time environments, large-scale system architectures, visualization and administrative tools.

A primary focus of his research is to build practical and functioning systems which can be analyzed/proved to be correct (in terms of timeliness, jitter, power efficiency, quality of service metrics, etc.). He was one of the principal contributors to Rate-Monotonic Analysis (RMA). RMA is supported by an impressive list of standards including POSIX Real-Time Extensions (IEEE 1003.1), the Real-Time Specification for Java, Real-Time UML (UML 2.0), Real-Time CORBA (CORBA 2.0), Ada 95, Ada 83, and automotive standards like OSEK and CANbus. He was the principal architect of tools like TimeWiz (from TimeSys), which supported schedulability analysis. He was the primary founder of TimeSys along with two other co-founders. (The company founded in 1996 now focuses on embedded Linux with an added emphasis on real-time features – thanks to his group’s work on real-time extensions such as Linux/RK).

Their work on resource kernels abstracts the notions of RMA, EDF and other real-time scheduling policies while enforcing the assumptions these theories make. It also extended resource management from processors to network bandwidth, disk bandwidth and physical memory resources. The resource set abstraction of resource kernels turns out to be an ideal construct for managing and minimizing power consumption without violating the responsiveness characteristics of applications. They have lately started to extend this notion to distributed systems (using Distributed RK), and to multicore processors.

He is a strong proponent of model-based design and development. Their tool, SysWeaver, embodies their approach and has been applied in multiple application domains including avionics, software radios, distributed automotive systems and sensor networks.

They then went a layer above and studied in depth tradeoffs across multiple applications each with multiple Quality of Service (QoS) dimensions using the same set of (finite) resources. This resulted in the QoS-based Model Allocation Framework (Q-RAM).

In recent years, his group has also spent a considerable amount of effort to build predictable wireless sensor networks. The FireFly sensor networks and their large-scale deployment of FireFly across the Carnegie Mellon campus (called Sensor Andrew) are concrete outcomes of this work.

Finally, thanks to their focus on real-time embedded and networked systems, they work with General Motors on Vehicular (V2V) Networks and on robust real-time platforms for autonomous driving. Yes, he
serves as the Co-Director from Carnegie Mellon of the General Motors-Carnegie Mellon Information Technology Collaborative Research Laboratory and the General Motors-Carnegie Mellon Autonomous Driving Autonomous Driving Collaborative Research Laboratory.
Keynote 4

Building Warehouse-scale Computers

Dr. John Wilkes, Principal Software Engineer, Technical Infrastructure

Google, USA

Abstract

Imagine some product team inside Google wants 100,000 CPU cores + RAM + flash + accelerators + disk in a couple of months. We need to decide where to put them, when; whether to deploy new machines, or re-purpose/reconfigure old ones; ensure we have enough power, cooling, networking, physical racks, data centers and (over longer a time-frame) wind power; cope with variances in delivery times from supply logistics hiccups; do multi-year cost-optimal placement+decisions in the face of literally thousands of different machine configurations; keep track of parts; schedule repairs, upgrades, and installations; and generally make all this happen behind the scenes at minimum cost.

And then after breakfast, we get dynamically allocate resources (on the small-minutes timescale) to the product groups that need them most urgently, accurately reflecting the cost (opex/capex) of all the machines and infrastructure we just deployed, and monitoring and controlling the datacenter power and cooling systems to achieve minimum overheads – even as we replace all of these on the fly.

This talk will highlight some of the exciting problems we’re working on inside Google to ensure we can supply the needs of an organization that is experiencing (literally) exponential growth in computing capacity.

Biography

John Wilkes has been at Google since 2008, where he is working on automation for building warehouse scale computers. Before this, he worked on cluster management for Google’s compute infrastructure (Borg, Omega, Kubernetes). He is interested in far too many aspects of distributed systems, but a recurring theme has been technologies that allow systems to manage themselves.

He received a PhD in computer science from the University of Cambridge, joined HP Labs in 1982, and was elected an HP Fellow and an ACM Fellow in 2002 for his work on storage system design. Along the way, he’s been program committee chair for SOSP, FAST, EuroSys and HotCloud, and has served on the steering committees for EuroSys, FAST, SoCC and HotCloud. He’s listed as an inventor on 40+ US patents, and has an adjunct faculty appointment at Carnegie-Mellon University. In his spare time he continues, stubbornly, trying to learn how to blow glass.