Software Engineering for Smart Systems

Jan Bosch (Professor of Software Engineering, Chalmers University of Technology)

Vision for the project

With the increasing size of software systems and the emergence of increasingly autonomous systems, future software engineering practices will exhibit six characteristics:

- All software (including safety critical software) will be continuously deployed (at least once per agile sprint) in virtually all systems while maintaining important properties like safety
- R&D teams will adopt experiment-driven development practices such as A/B testing for feature development and "minimal viable product" (MVP) approach for new product development
- Systems will use streaming analytics inspired approaches to incorporate data-driven behavior for general systems and for personalization purposes
- Systems will autonomously experiment with their own behavior to improve the value delivered to customers and other stakeholders
- Collaboration with other systems in a System of Systems setting will be exploited to enhance the capabilities of the system to sense and collect information about its environment
- Families of similar systems will learn from each other automatically

Although individual researchers and professionals in industry are researching and building elements of this vision, it becomes clear that *there are no development approaches that combine human experimentation, autonomous experimentation and data-driven behavior in an integral approach including architecture and development processes*. As the Swedish software-intensive systems industry continues to shift R&D resources from mechanics and electronics to software, in response to the growing software size and the growing role of software in achieving differentiation and value delivery, we need to be world-class at building systems following the aforementioned principles at scale, with hundreds and thousands of software engineers involved.

The goal of this project is twofold. First, we develop architectural styles and patterns as well as domain-specific architectures for self-adaptive systems that employ automated experimentation to autonomously improve the performance of families of systems. Second, we develop evidence-based engineering approaches that uniformly combine human-driven experimentation and automated experimentation.

Research Challenges

The overall research goal is the generation of development approaches that combine human experimentation, autonomous experimentation and data---driven behavior in an integral approach. In order to realize this, we have recognized four research challenges that we study in this project.

First, we need to conduct research on novel, self-reflective and self-adaptive software architectures that have a set of unique characteristics. A self-reflective architecture is able to measure and track its own performance whereas self-adaptive refers to the ability to adjust its own behavior. To accomplish this, as well as to support experimentation, data collection needs to be an integral part of the architecture. In addition, the architecture needs to support continuous deployment and facilitate automated experimentation.

Second, novel software engineering methods are required that are based on experiment-driven, rather than requirements driven development. These development methods need to support data- driven or evidence-based development, meaning the use of data from systems in the field and the behavior of users of these systems for decision making. In addition, these methods need to support the transition from human experimentation by R&D teams to automated experimentation.

Third, to support the aforementioned methods for software engineering, new techniques for data analytics and use are required. These techniques are required to establish the success of autonomous behavior and determine value of new features. In addition, these techniques need to support automated experimentation.

The final research challenge is concerned with the fact that smart systems will exist in families of dozens, hundreds or thousands of similar instances of each system. As these systems are experimenting with their own behavior, we need to develop techniques that allow the R&D teams responsible for these systems as well as the systems themselves to learn from the family of systems. The learnings need to be merged and redistributed to all systems in order to ensure continuous improvement of all systems.

Industrial Challenges

The industrial challenge is predominantly one of scaling, both the engineering endeavor and the data collection and analysis practice. Already today, Swedish companies can build small autonomous systems using small teams of highly skilled engineers and researchers. These systems and teams however are not employing experimentation and data-driven techniques, but rather employ trial-and-error as their primary mechanism.

However, future industrial engineering of large software-intensive systems, including autonomous systems, will require the involvement of hundreds and potentially thousands of engineers, and data collection from hundreds of thousands or millions of instances of the systems. Currently, there are no engineering approaches that support the future data-driven development practices outlined above and that scale to the level that will be required for large-scale industrial deployment.

This project is concerned with scaling, i.e. developing architectures and development practices that support large-scale development employing the trends and technologies that will become available in the coming decade.

Sub- projects

The project consists of five sub-projects, described in more detail below. Each subproject addresses part of the overall challenge outlined above. Frequent project meetings ensure alignment between the research activities.

Data-driven continuous evolution of smart systems

David Issa Mattos (PhD student, Chalmers), Jan Bosch (advisor, Chalmers University of Technology), Helena Holmström Olsson (co-advisor, Malmö University)

The subproject is concerned with self-adaptive architectures that inherently support automated experimentation and data-driven behavior. In the first phase, the project seeks to develop new architecture styles, patterns and solutions for self-adaptive architectures and systems that address the requirements including integral data collection and automated experimentation. In the second phase, we expand the project to focus on the development methods required to develop systems using self-adaptive architecture. During both phases, industrial validation using the WARAs (Wallenberg Autonomous Research Arenas) is planned to ensure industrial relevance and scalability.

Data collection and analysis for continuous experimentation

Rasmus Ros (PhD student, Lund University), Per Runeson (advisor, Lund University), Elizabeth Bjarnason (co-advisor, Lund University)

This subproject aims at exploring and advancing methods for operational data collection and analysis in the above described context of continuous deployment and experimentation. In current software engineering practices there are significant gaps between the users and the engineers, leading to suboptimal solutions. In order to bridge this gap, methods for continuous monitoring of system usage and experimentation with new and modified features has evolved. Current approaches are limited to reactive evaluation of minor variations of the software services. We aim to go beyond basic quantitative measurements of response, and explore options of qualitative analysis of user behavior, to allow proactive recommendations to be presented to the software engineers. In a first step, we survey in detail existing proposals, secondly we focus on the data collection and analysis from the architectures, mentioned above, and thirdly, integrate the recommendation system into the organic development environment, discussed below.

Collective run-time adaptation of systems of systems
Piergiuseppe Mallozzi (PhD student, Chalmers), Patrizio Pellicione (advisor, Gothenburg University), Christian Berger (co-advisor, Gothenburg University)

This subproject will focus on the automotive domain (cars that cooperate with other cars, cyclists, pedestrians, road, lamppost, etc.). State-of-the-art methods assume to have a complete understanding of the vehicle and all the possible interactions with the environment as a pre-requisite to guarantee safety. Unfortunately, this is not often the case due to huge variety of environments. This huge variability and uncertainty hamper the specification at design time of all the possible configurations. Furthermore, today's safety standards (such as ISO 26262 for the automotive domain) do not address issues where collaborative vehicles can share safety information. Safety might easily become a bottleneck preventing the transition of new and promising ideas into business successes.

The openness and adaptability of smart vehicles require moving part of the certification at run time. We will investigate methods and techniques to assure that a vehicle is going to keep certain safety requirements despite adaptations triggered by the different environments in which the vehicle will operate. Part of the safety analysis will be performed at design time, however the final assurance of safety properties will be shifted at run-time once all the information needed to perform the analysis will be available. Our strategy will be investigating incremental verification approaches that will permit to deduce the correctness of a system of systems by the results of verifications of individual constituent systems.

Organic evolution of development organizations
Rebekka Wohlrab (industrial PhD student, Systemite AB, Chalmers), Patrizio Pellicione (advisor, Gothenburg University), Riccardo Scandariato (co-advisor, Gothenburg University), Ali Shahrokni (Systemite AB)

The ambition of this subproject is to conceive intelligent, integrated, and system-level development data management methods and tools to support organizations on their mission towards developing smart systems. The new advanced, distributed mechatronics systems manifesting as self-driving cars or Internet-of-things have pushed people and organizations to a different level of thinking; a level where understanding and relating to the context and the bigger picture, a system architecture, is at least as important as providing high quality in the details. This small shift in perspective demands a new way of organizing and even new methods of handling the changes in new organizations.

Data management methods and tools support development organizations in their "organic evolution" that involves and keep synchronized during evolution three distinct but interdependent structures: (i) the architecture of the system under design, development, or refinement, (ii) the structure of the organization, including also partners, subcontractors, and others, and (iii) the production infrastructure used to develop and deploy the system. This synchronization will reduce problems caused by both technical debt, i.e. when the architecture is lagging, and social debt, i.e. when the structure of the organization is missing.

Autonomous Energy Efficient Software for 5G Radio Access Networks

Diarmuid Corcoran(Industrial PhD student KTH, Ericsson AB),Christian Schulte (advisor KTH), Rolf Stadler (co- advisor, KTH), Andreas Ermedahl (Industrial advisor, Ericsson AB)

This sub-project project will investigate the impact of new, distributed, software techniques on the energy efficiency of 5G radio access compute infrastructure. The goal is to create intelligent software techniques that will contribute to affordable, sustainable and energy efficient 5G radio access networks in a virtualized and cloud context. A fundamental principal will be the mechanisms and techniques to enable autonomous energy efficient architectures and algorithms.

Understanding energy in radio access infrastructure is a multi-faceted challenge consisting of antenna near components (SW and HW), transport network components (SW and HW) and compute infrastructure components (SW and HW). The proposal will study aspects of the compute infrastructure and partially the transport infrastructure. In particular it will investigate how control and user plane (payload) software functions can be adaptively provisioned across the, cloud based, compute and network infrastructure in order to minimize system energy consumption. This is a highly complex issue where system properties, such as, failure, robustness, scalability, latency and hardware capabilities need to be considered. To manage this complex situation and build intelligent software solutions we propose investigating the potential of and, where appropriate, enhancing current state-of-the-art tools, such as, real-time analytics, machine learning and constraint based optimization techniques. These techniques should be adapted and integrated into the functional architecture of the 5G network in novel ways.