Invited Talk

Towards Self-Adaptive Software-Intensive Legacy Systems

Hausi A. Müller
Department of Computer Science
University of Victoria, Canada
hausi@cs.uvic.ca
http://webhome.cs.uvic.ca/~hausi/

Abstract

Software Legacy systems typically refer to large, complex systems that have evolved to a state where they significantly resist further modification and evolution. Over the past decade, the life of many of these cash cows of the software industry has been extended by migrating them to modern platforms. Recently, in part due to the proliferation of the Web, those same systems seem to be morphing again into highly configurable systems of systems. According to a recent SEI study, software-intensive systems of systems are likely to evolve into Ultra-Large-Scale (ULS) Systems. In this talk we will argue that software-intensive legacy systems need to morph into self-adaptive and self-managing systems to be able to cope with the challenges of continuous evolution.

The simultaneous explosion of information and integration of technology and the continuous evolution from software intensive systems to systems of systems to ULS systems requires new and innovative approaches for building, running and managing software systems. A consequence of this continuous evolution is that software systems must become more versatile, flexible, resilient, dependable, robust, continuously available, energy-efficient, recoverable, customizable, self-healing, configurable, or self-optimizing by adapting to changing contexts and environments. One of the most promising approaches to achieving such properties is to equip software systems with self-adaptation mechanisms.

The challenge therefore for the software reverse engineering and reengineering communities is to investigate methods and techniques to instrument software-intensive systems with monitoring and self-adaptive mechanisms.
Bio

Dr. Hausi Müller is a Professor in the Department of Computer Science at the University of Victoria, British Columbia, Canada. He was the founding Director of BSENG, a CEAB accredited Bachelor of Software Engineering degree program in the Faculty of Engineering. He is a Visiting Scientist at CAS, the Center for Advanced Studies at the IBM Toronto Laboratory and SEI, the Carnegie Mellon Software Engineering Institute. For over a decade he has been a principal investigator and Chair of the Technical Steering Committee of CSER, a Canadian Consortium for Software Engineering Research. In 2006 he received the IBM CAS Faculty Fellow of the Year Award, the CSER Outstanding Leadership Award, and Stevens Citation for his many contributions to the software reverse engineering community.

Together with his research group and in collaboration with IBM Canada and CA Canada he investigates methods, models, architectures, and techniques for autonomic and adaptive systems and for application monitoring and diagnostics. He also concentrates on building ACSE (Adoption-Centric Software Engineering) tools and on migrating legacy software to network-centric and enterprise application platforms. Dr. Müller's research interests include software engineering, software evolution, autonomic computing, monitoring and diagnostics, service-oriented architectures, adoption-centric software engineering, software architecture, software reverse engineering, software reengineering, program understanding, visualization, and software engineering tool evaluation.