Service Components and Ensembles: Building Blocks for Autonomous Systems

N. Serbedzija
Fraunhofer FOKUS
nikola.serbedzija@fokus.fraunhofer.de

1. Topics

The major topics are: Engineering Complex Autonomous Systems, Awareness in software, Adaptive components, Service component ensembles, Programming with service components, Reasoning about system properties, Case studies (Swarm robotics, Cloud Computing, E-mobility). Crossing the discipline barriers, the author brings different engineering perspectives illustrating novel concepts on relevant application domains.

2. Brief abstract of the presentation

Modern control systems are highly collective, constructed of numerous independent entities that have individual, but also share common goals. Their elements are both autonomous and cooperative featuring a high level of self awareness and self expressiveness. A complex control system built with such entities must be robust and adaptive offering maximal utilization with minimal energy and resource use.

The tutorial details a novel approach to respond to such challenges. Several software engineering aspects will be presented: (1) service–component ensembles (SCE) as means to dynamically structure independent and distributed system entities; (2) formalization and modeling the fundamental SCE properties as means to rigorously reason about autonomous behavior and aware-rich networking, and (3) adaptive and knowledge-rich software environments and tools for the development of self-aware, self-adaptive and self-expressive autonomic systems.

The approach to construct complex systems using ensembles of service components is generic and applicable in a number of domains. It relies on general-purpose linguistic and tools support, as well as mechanisms to reason about and to prove system properties. The presented theoretical concepts will be illustrated on four different case studies featuring optimization and adaptive control for vehicular assistance, swarm robotics, cloud computing and e-mobility. All four case studies will be explained in details showing a strong pragmatic orientation of the approach.

[see also a popular text @ http://blog.ascens-ist.eu/ or promotion PerAdaTV video @ http://reflect.pst.ifi.lmu.de/]

3. Background knowledge expected of the participants

The tutorial is meant for researchers, practitioners, system designers, teachers and advanced students who want to gain better understanding of new developments and future trends in adaptive control, awareness-rich and adaptive systems.. In addition, it features numerous practical examples. No specific technical expertise is required, since the focus is on principles and design issues rather than on specific details. More advanced techniques are introduced with thorough explanations and four practical application scenarios are used to make those complex themes closer to a wider audience.

4. Objectives

The goal of the tutorial is to presents novel ideas that bring awareness, knowledge, adaptation and emergence into technical systems. It deals with the problems of the next generation systems and indicates further challenges in the area. Using this approach the designers can control and engineer the intended and emergent behaviours with static and dynamic support from formal methods. The approach features:

- Service Component Ensembles (SCE) as reliable, predictable, self-adaptive software entities that balance efficient execution and flexible behaviour via dynamic self-expression.
- Service Component (SC) as self-aware software elements with adaptive behaviour, based on context knowledge.

- Complex adaptive software intensive systems composed by SCs and SCEs in dynamic scenarios.
- Innovative language for service components and service component ensembles (SCEL), integrating self-awareness features and adaptive behaviours as first-class citizens.
- Formal framework integrating classical models of computation and control theory with novel knowledge and game-theoretic models.
- Adaptation patterns and mechanisms for SCs and for SCEs smoothing the distinction between bottom-up and top-down approaches.
- New verification techniques for non-functional properties characterizing resource management and adaptive behaviour, as well as for security.
- An open source "first-of-its-kind" tool-integration platform for the development of complex software systems based on ensembles.

A number of visionary application scenarios are presented, justifying the pragmatic significance of this new technology, followed by a demonstration of an adaptive system.

5. Time allocations for the major course topics

Tutorial Content:	Duration: ½ day
• Motivation	(20 minutes)
• Service Ensembles Concept	(20 minutes)
• SCEL Language to program service components	(20 minutes)
• Adaptation patterns	(20 minutes)
• Tool support – integration tools	(20 minutes)
• Reasoning about system properties	(20 minutes)
• Case studies	(60 minutes)
• Conclusion and discussion	(15 minutes)

6. Qualifications of the instructor(s)

Prof. Nikola Šerbedžija works at Fraunhofer FIRST where he is responsible for new research activities and innovative technology. He was a visiting professor at University of Technology Sydney (1999-2000) and at University of Arts, Berlin (2000 – 2008). His major research areas are: Adaptive Control, Pervasive Adaptation, Ubiquitous Computing, Middleware Architectures, and Internet Programming, mostly applied within embedded and real-time systems, ambient assistance and empathic systems. As a principle designer he led the developments of a number of practical systems in vehicular [leading the REFLEC project http://reflect.pst.ifi.lmu.de/, in- and out-door infrastructures, e-commerce and e-learning domains. His is currently involved in a large EU project dealing with autonomous control [ASCENS project: http://www.ascens-ist.eu/]. He is a member of IFIP2.4 Working Group and advisory board member for Pervasive adaptation [http://www.perada.org/] and Awareness [http://www.aware-project.eu/] EU initiatives. He is the author of more than 100 scientific articles and he held numerous tutorials and key notes.

References:

Recent tutorials/keynotes:

N. Serbedzija: Heaven and Hell: Visions for Pervasive Adaptation**,** FET11 Budapest Congress and World Trade Center, May 2011 (keynote)

N. Serbedzija: Reflective Computing, Véhicules et transports intelligents et communicants, Telecom ParisTech, Paris 30 May 2011 (keynote)

N. Serbedzija, *"What you Like is What you Get" – Engineering a Bridge between Privacy and Personalization*, MobiSec 2010, Catania, May 2010 (*mobisec.org/) and BodyNets2010, Corfu Sep 2010 (www.bodynets.org/welcome.shtml) (tutorial)*

N. Serbedzija: *"E-Privacy: A Missing Axis in Digital Space", E-Society 2010, hwww.esociety-conf.org/*

N. Serbedzija: ""How to protect privacy when we do not know what endangers it" opening talk, YUINFO 2010, *www.e-drustvo.org/yuinfo/*

N. Serbedzija, "Engineering Affective Systems, 2008 PERADA Summer School (www. Perada.org)

Most recent publications in journals and book chapters:

1. N. Serbedzija and S. Fairclough. Reflective Pervasive Systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 7 (1), April 2012 [http://dl.acm.org/citation.cfm?id=2168272]
2. N. Serbedzija. Reflective Computing – Naturally Artificial, N. Serbedzija. Reflective Computing – Naturally Artificial, This Pervasive Day. (Ed. Jeremy Pitt), ISBN: 978-1-84816-748-3, Imperial College Press, June 2012. [http://www.icpress.co.uk/compsci/p790.html]
3. N. Serbedzija G. Beyer. Reflective Assistance Pervasive Adaptation in Real Life Computing , *FET'11European Future Technology Conference*, Elsevier Proceedings Computer Science, (2011)
4. N. Serbedzija, G-M. Bertolloti, Adaptive and Personalized Body Networking, Journal *of Autonomous and Adaptive Communications Systems*, Vol.6, No.3, 2013.
5. G. Kock, M. Ribaric and N. Serbedzija. Modeling User-Centric Pervasive Adaptive Systems - the REFLECT Ontology. In: Intelligent Systems for Knowledge Management, Vol. 252. Nguyen, Ngoc Thanh; and Szczerbicki, Edward (Eds.) Series: "Studies in Computational Intelligence", Springer 2009, ISBN: 978-3-642-04169-3.

Components are partially autonomous entities that can participate in ensembles (become their members). Components proactively sense the SSAâ€™s environment and provide their knowledge to other components to allow them to take smart and well-founded decisions.Â There are many other smart systems where ensembles are inherently involved; for instance, it would be natural to apply them in the examples provided in [3, 4], such as on-street parking meters, employing swarms of sensors in a vehicle, and a number of applications in the area of road-side computing and intelligent transportation.Â Obviously, in such settings, building the necessary programming abstractions and machinery for ensemble specification and formation from scratch is practically infeasible. ASCENS: Engineering Autonomic. Service-Component Ensembles. Martin Wirsing1, Matthias HÂ¨olzl1, Mirco Tribastone1, and Franco Zambonelli2.Â build systems with self-aware, intelligent components that mimic natural fea-. tures like adaptation, self-organization, and autonomous as well as collective be-. havior. However, traditional software engineering, both agile and heavyweight While individual algorithmic building blocks such as perception and planning have been explored before in the literature [15], [9], [10], [13], we focus on the data-ow across the building blocks and their inherent task-level parallelisms (TLP).Â The unique scenarios and use-cases let us build commer-cial autonomous vehicles at a reasonable cost.Â 2. The latency consists of four major components: the time for the computing system to generate control commands from the sensor inputs (Tcomp), the time to transmit the control commands to the vehicleâ€™s actuator through the Controller Area Network (CAN) bus (Tdata), the time it takes for the mechanical components of the vehicle to start reacting (Tmech).