Hauptseminar SS2017:

„From *Algorithmic* Computing to *Autonomic* Computing“

Prof. Dr. Frank J. Furrer

Context

In ***algorithmic computing***, the program follows a predefined set of rules – the algorithm. The analyst/designer of the program analyzes the intended tasks of the program, defines the rules for its expected behaviour and programs the implementation. The creators of algorithmic software must therefore foresee, identify and implement all possible cases for its behaviour in the future application!

However, what if the problem is *not fully defined*? Or the *environment is uncertain*? What if situations are *too complex to be predicted*? Or the environment is *changing dynamically*? In many such cases algorithmic computing fails.

In such situations, the software needs an additional degree of freedom: ***Autonomy***! Autonomy allows software to adapt to partially defined problems, to uncertain or dynamically changing environments and to situations that are too complex to be predicted. As more and more applications – such as autonomous cars and planes, adaptive power grid management, survivable networks, and many more – fall into this category, a gradual *switch from algorithmic computing to autonomic* computing takes place.

Autonomic computing has become an important software engineering discipline with a rich literature, an active research community, and a growing number of applications.

History

Autonomic computing ([1], [2], [3]) has been introduced by IBM in 2001: The reason was the growth in complexity of *modern computer installations*, which was becoming unmanageable. The basic idea was to develop management software with *self-* properties*. The originally proposed set of properties were *self-configuration*, *self-healing*, *self-optimization* and *self-protection*. Autonomic computing systems therefore are able to re-organize, optimize, defend and adapt themselves with no real-time human intervention ([6]). Autonomic computing relies on many branches of science – especially computer science, artificial intelligence, control theory, machine learning, multi-agent systems and more.

Autonomic Computing Architecture

The key element proposed by IBM was the *autonomic computing architecture*: The MAPE-K architecture (= **M**onitor – **A**nalyse – **P**lan – **E**xecute – **K**nowledge). This architecture is basically a modern control loop, reading state values from the physical computer installation, processing it through the MAPE-K loop and acting on the physical computer installation ([3], [6], [7]). The interesting part is the “K” – for knowledge: The MAPE-K loop utilizes technologies from the field of *artificial intelligence* ([8], [9]) in order to reach various degrees of autonomy.

Cyber-Physical Systems

Soon after the introduction of the MAPE-K architecture, it became widely used in the construction of ***cyber-physical systems*** ([6], [7]). Cyber-physical systems (CPS) are cooperating networks of *physical* and *computational* components. In a CPS – such as a modern car, plane or train – the physical components are *controlled by software*. The “intelligence” is therefore implemented in programs with varying degrees of autonomy, often using technologies from the field of artificial intelligence.

Seminar Work

This seminar will work on the central theme: **Which are the history, the technologies and the applications of Autonomic Computing?**

Each participant choses one of the 3 fields:

F1: Research and describe the *historical development* from algorithmic computing to today’s autonomic computing;

F2: Investigate which *autonomic computing technologies* fill the MAPE-K architecture today and in the future;

F3: Identify and document a number of *promising applications* of cyber-physical systems (CPS) based on autonomic computing.

The Hauptseminar has *3 seminar days* (see separate work program, dates below):

- An introduction day: Autonomic Computing will be introduced in a lecture by Professor Dr. Frank J. Furrer, and the parts of the Hauptseminar (Paper, presentation) will be defined,
- Individual, guided research in the selected area and authoring of a scientific paper. Feedback from peer reviewers,
- A first seminar day: The participants will present their results and receive feedback from the audience,
- Improvement of the paper and the presentation, based on the peer feedback,
- A second seminar day: The participants will present their improved results and receive feedback from the audience,
- Delivery of the final paper.

The participants will learn: (a) to do focused research in a specific area (“Autonomic Computing”), (b) to author a scientific paper, (c) to improve their LATex expertise, (d) to experience the peer-review process and (e) to hold convincing presentations, and (f) to benefit from a considerable broading of their perspective in the field of technology, software, and applications.

As a final outcome of the seminar, a *proceedings* volume – including all the papers produced by the participants – will be assembled and made available in electronic form to anybody interested.

Seminar language is English. Three seminar days will be held and 3 ECTS credits are awarded for the successful participation. Audience is limited to 7 participants. Please register in advance.

Seminar Schedule:

Kick-Off Meeting (Introduction): Wednesday, **April 19, 2017** / 11:10 – 12:40 in APB/INF 2101

Seminar Day 1: Wednesday, **June 7, 2017** / 09:20 – 10:50 & 11:10 – 12:40 in APB/INF 2101

Seminar Day 2: Wednesday, **July 12, 2017** / 09:20 – 10:50 & 11:10 – 12:40 in APB/INF 2101

References

1) Mandatory Reading:

The fundamental knowledge:

[1] Philippe Lalanda, Julie A. McCann, Ada Diaconescu: **Autonomic Computing – *Principles, Design and Implementation.*** Springer-Verlag, London UK, 2014. ISBN 978-1-4471-5006-0.

The seminal work:

[2] IBM Research Paper, 2001: **Autonomic Computing – *IBM's Perspective on the State of Information Technology***. Downloadable from: http://people.scs.carleton.ca/~soma/biosec/readings/autonomic_computing.pdf [last accessed: 2.2.2016]

Introduction to the Architecture:

[3] IBM White Paper: ***An architectural blueprint for autonomic computing***. 3rd edition, June 2005. Downloadable from: http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf [last accessed: 2.2.2016]

Cyber-Physical Systems:

[4] NIST Engineering Laboratory: **Cyber-Physical Systems**. Downloadable from: https://www.nist.gov/el/cyber-physical-systems [last accessed: 23212017]

2) References:

[5] Mark Klein, Raj Rajkumar, Dionisio De Niz: Cyber-Physical Systems (SEI Series in Software Engineering) Addison Wesley, USA, 2016. ISBN 978-0-321-92696-8

[6] Manish Parashar, Salim Hariri: **Autonomic Computing – *An Overview***. In: J.-P. Banatre et al. (Eds.), UPP 2004, LNCS 3566, pp. 247–259. Springer-Verlag Berlin Heidelberg, 2005. Downloadable from: http://143.106.148.168:9080/Cursos/IA844/02-11/autonomic.pdf [last accessed: 23.1.2017]

[7] Payal Mittal, Abhay Bansal, Abhishek Singhal: **A Study on Architecture of Autonomic Computing Self Managed Systems.** International Journal of Computer Applications (0975 – 8887), Volume 92, No.6, April 2014. Downloadable from: http://research.ijcaonline.org/volume92/number6/pxc3894890.pdf [last accessed: 23.1.2017]

[8] **Artificial Intelligence – *Overview***. Downloadable from: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm [last accessed: 23.1.2017]

[9] Stuart J. Russell, Peter Norvig: **Artificial Intelligence – *A Modern Approach***. Prentice Hall International, USA, 3rd revised edition, 2016. ISBN 978-1-292-15396-4

V1.0/24.01.2017