Intelligent Systems

Chapter 2: Introduction

Winter Term 2019 / 2020

Prof. Dr.-Ing. habil. Sven Tomforde
Institute of Computer Science / Intelligent Systems group

About this Chapter

Content

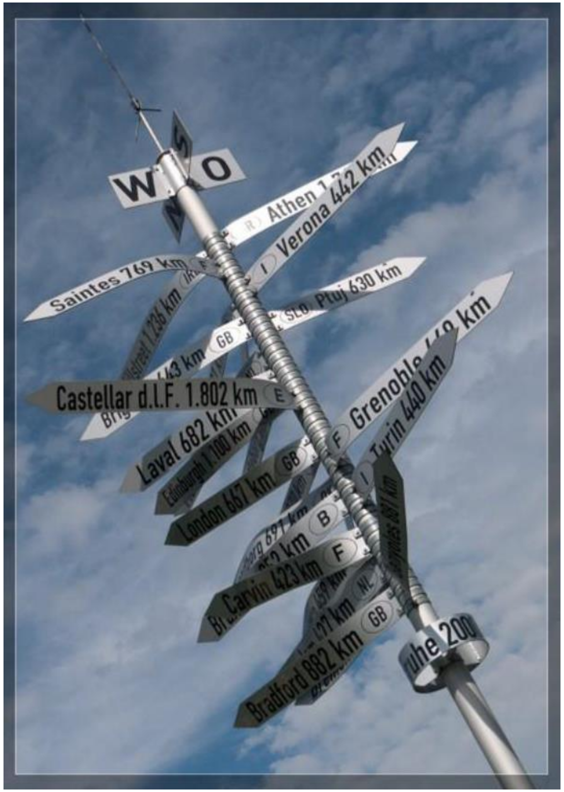
- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings

Goals – Students should be able to …

- … describe the motivation for Intelligent Systems / Organic Computing.
- … give examples for raising complexity in information and communication systems.
- … explain how natural processes can be used as inspiration.
- … summarise which aspects are covered by Organic Computing.

Agenda

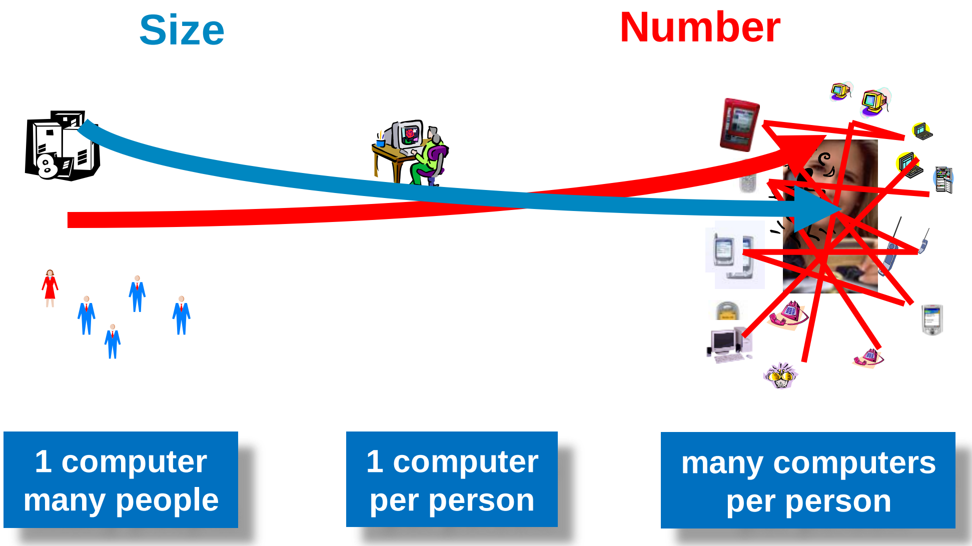
- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings



Motivation

C | A | U
Christian-Albrechts-Universität zu Kiel

- Computing trends
  - Number of devices
    → increases
  - Computational power
    → increases
  - Malfunctions due to mutual influences



- Observations
  - Failures due to high complexity
    → Manageability of interconnected systems decreases
  - Unknown configuration space (dynamic environment)
    → Impossible to predict all possible situations

Traditional system design

Christian-Albrechts-Universität zu Kiel

- Traditional Systems Engineering
  - Relies (mainly) on hierarchical top-down design methods.
  - Everything should be pre-planned and tested at design-time.

- Intelligent Systems
  - Complexity of system tasks is conquered by a distribution of responsibilities.
  - Engineer at design-time is responsible for the basic system design and defining the scope of system‘s freedom.
  - System becomes self-managing by means of automated discovery of appropriate decisions.

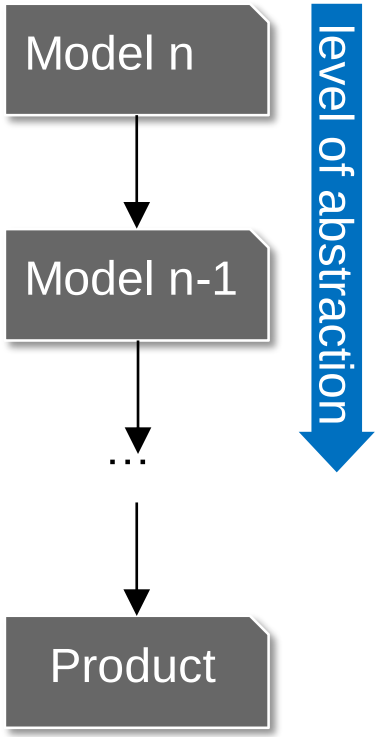
Mastering of complexity in the presence of permanent change!

Traditional system design (2)

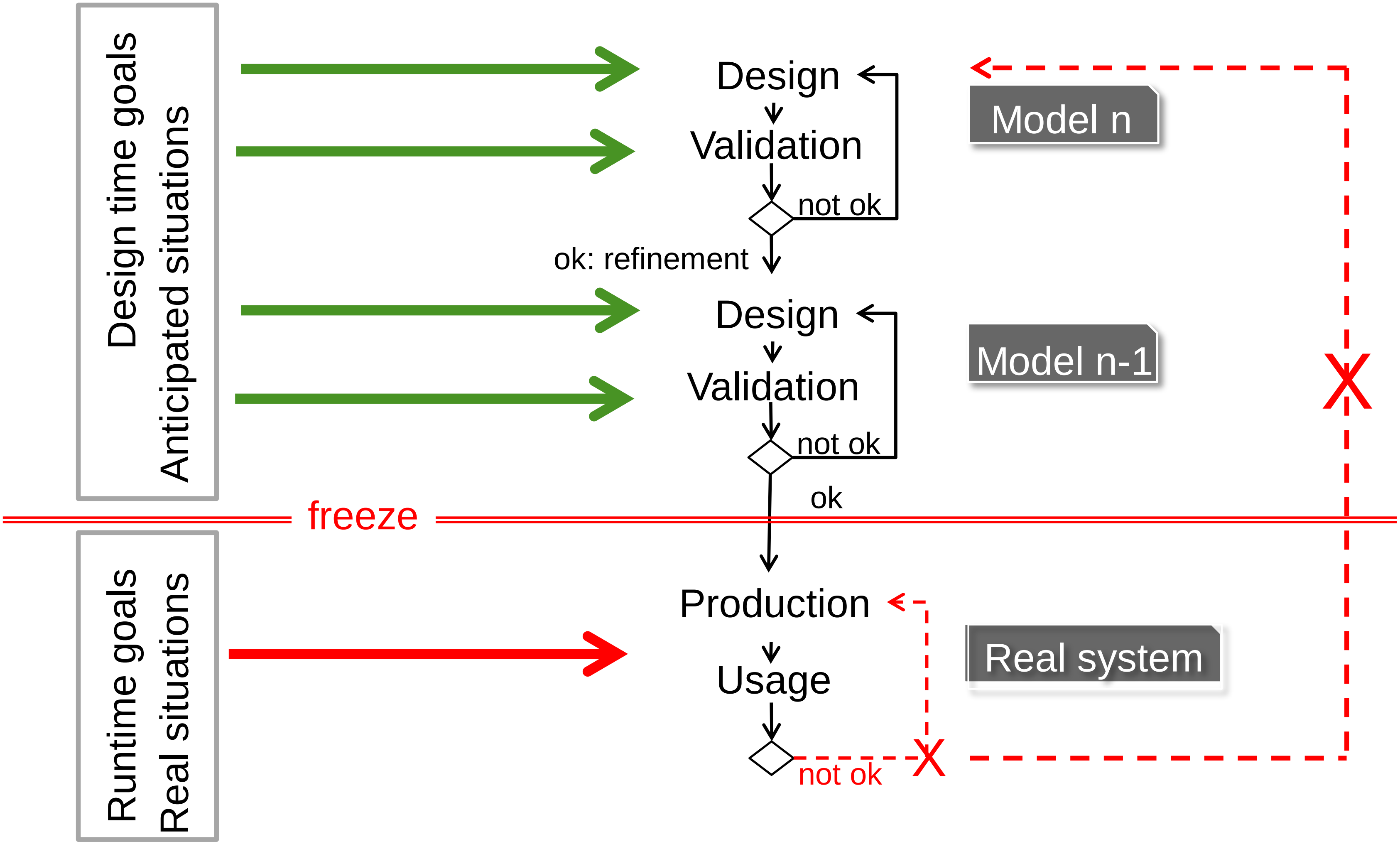
Christian-Albrechts-Universität zu Kiel

- Traditional Systems Engineering
  - Specifies requirements
  - Modelling at abstract level
  - Refine at more detailed level
  - Until the product (system) is finished
  - Test with all foreseen situations

- Traditional design processes
  - Pre-planned and fully tested solutions at design time
  - No revision of design decisions at runtime



Traditional Design Process

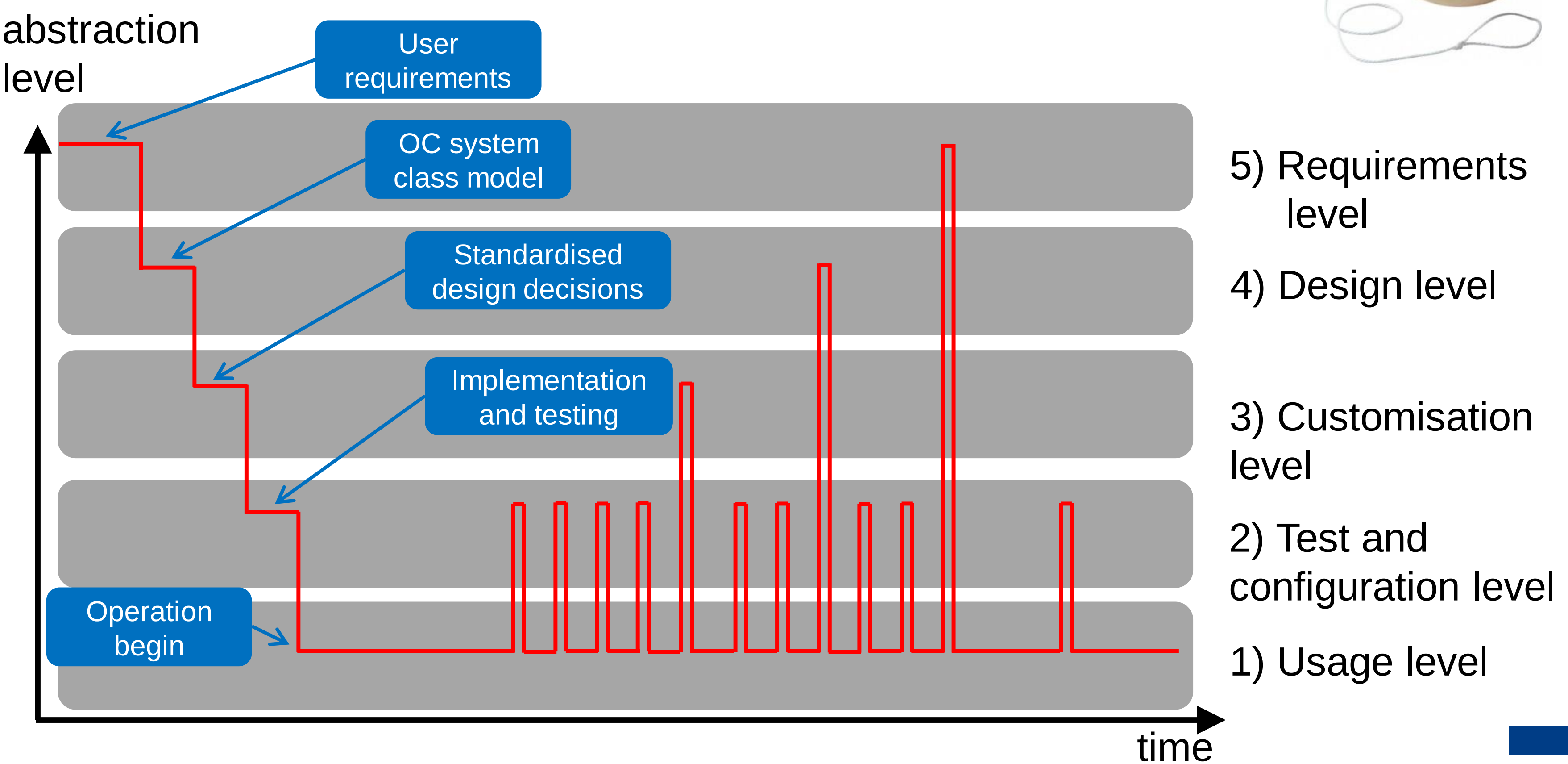


From traditional to intelligent systems

- Intelligent systems consist of autonomous sub-systems:
  - The sub-systems possess sensors and actuators.
  - Sub-systems interact with each other and the environment.
  - No global (or: system-wide) control necessary.
- The resulting interconnected intelligent systems have to:
  - organise themselves,
  - be adaptive and flexible, and
  - learn the optimal behaviour – in order to be able to adapt themselves autonomously to previously unknown situations.
- Most of the decisions are taken based on local knowledge and without user / system-wide influence!

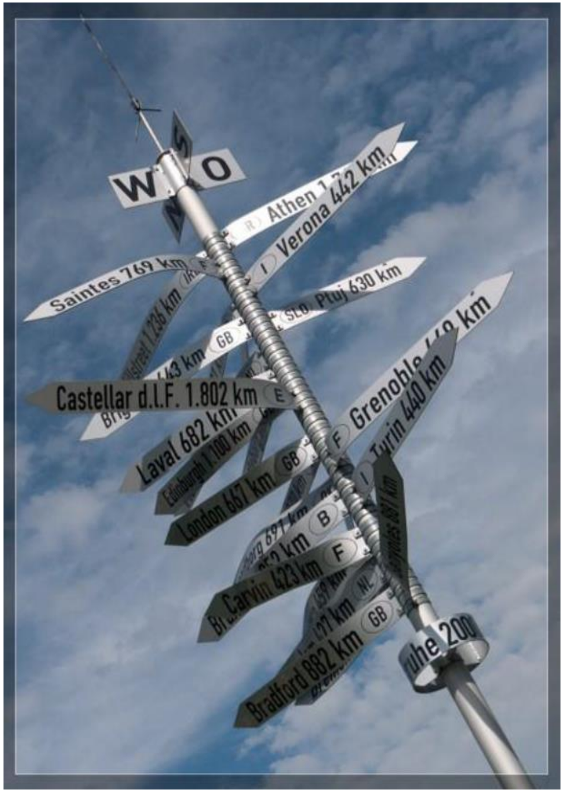
Intelligent System Process

Goal: Automated Jo-Jo design



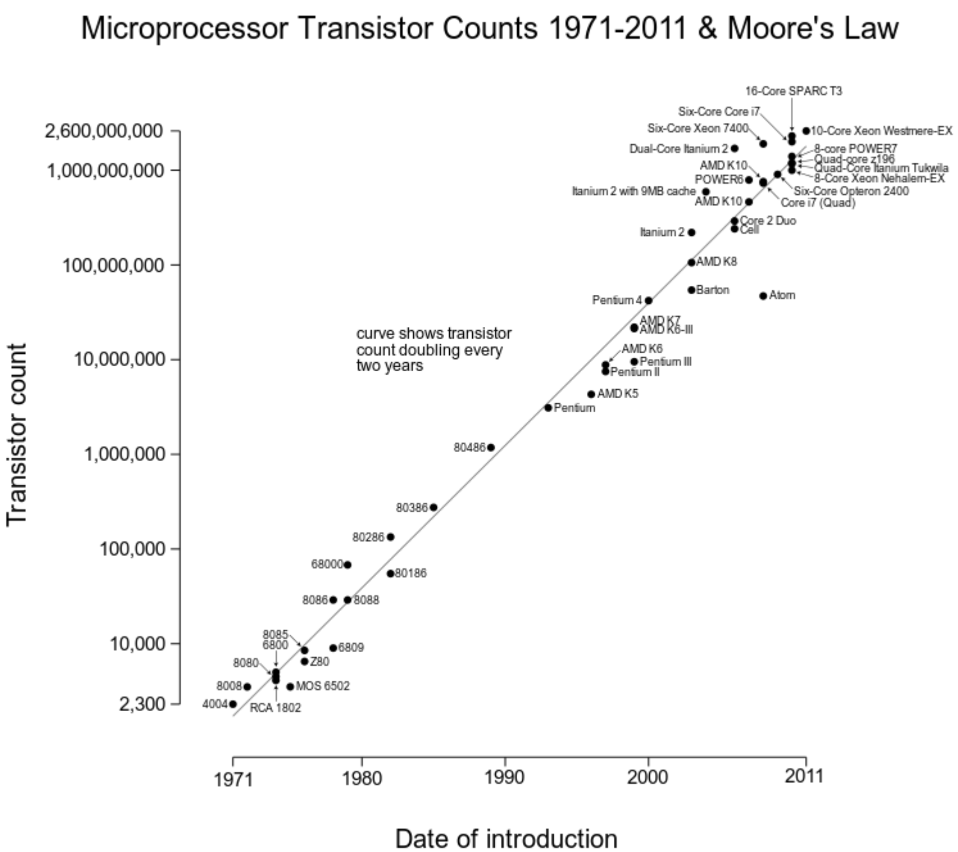
Agenda

- Motivation
- **Examples for growing complexity**
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings

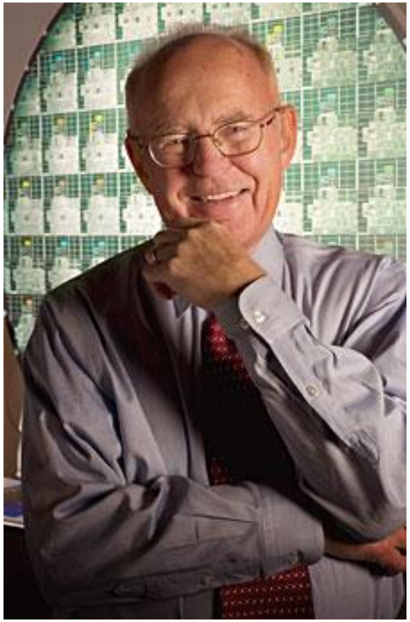


Example: Moore‘s Law

Christian-Albrechts-Universität zu Kiel



Gordon Moore in 1965:
„The complexity of integrated circuits (IC) with minimal component cost is roughly doubled every two years!“

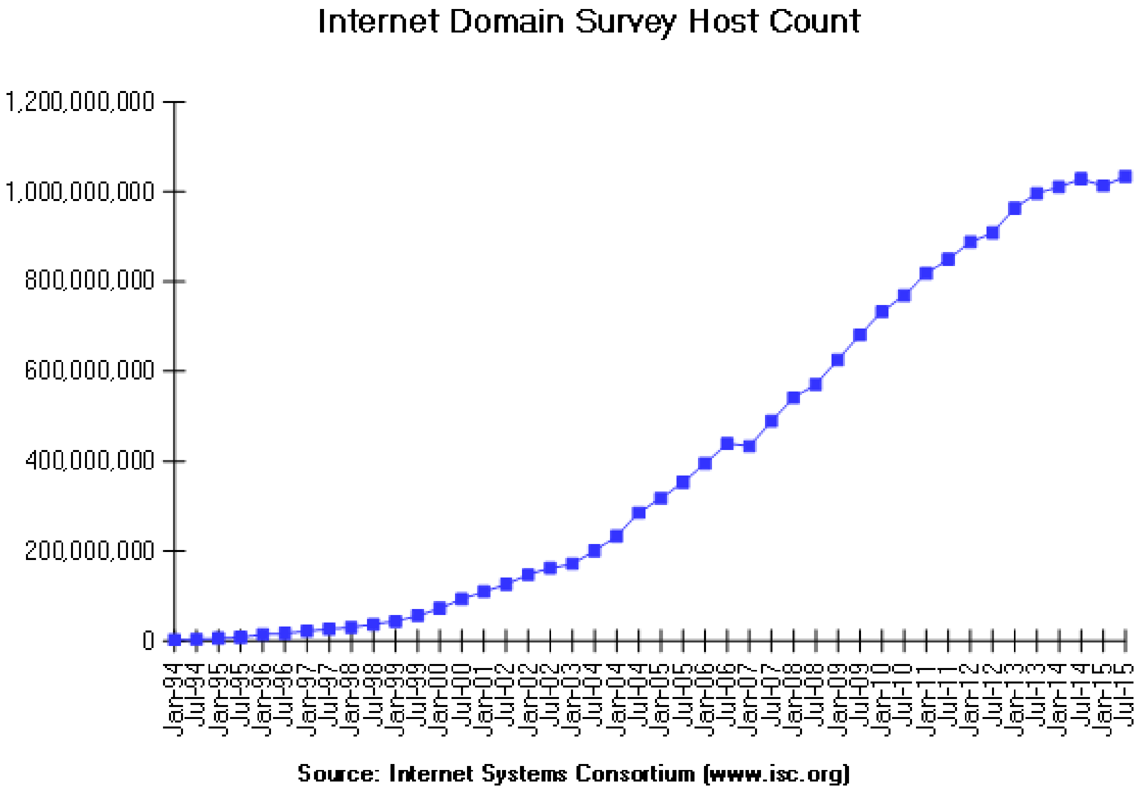


Source: intel.com

Example: Internet hosts

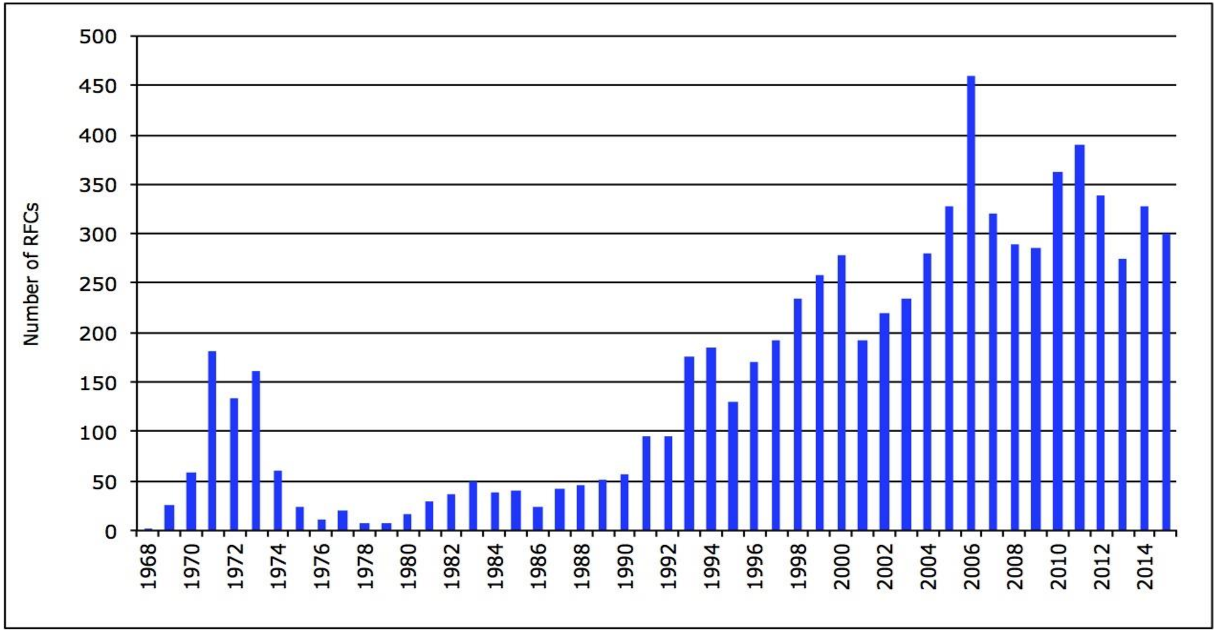
Christian-Albrechts-Universität zu Kiel

Development of the number of hosts reachable via IP address:



Source: Internet Systems Consortium at www.isc.org

Example: Requests for Comments



- The Requests for Comments (RFC) are a series of technical and organisational documents for the Internet (originally: Arpanet) that has been launched on April 7th, 1969.
- All basic building blocks and standards of Internet technology initially started as RFC, including Email, IP, URL, or calendar formats.

Example: Glass‘ Law

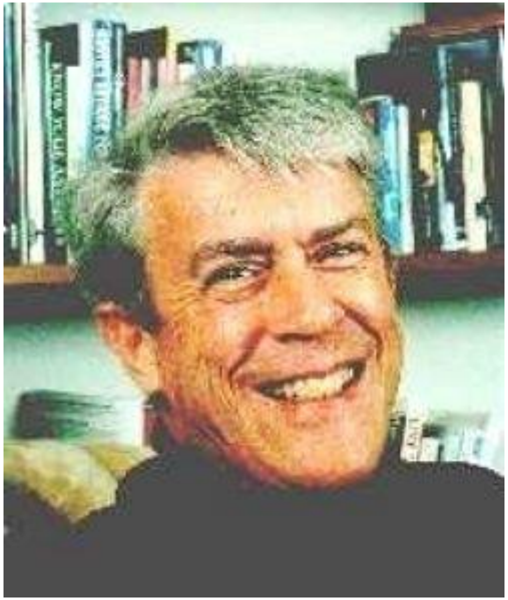
In software engineering:

Glass states that “IT complexity is indirectly related to functionality, in that a 25% increase in functionality increases complexity by 100%”.

- For every 25% increase in the business functionality in a service, there is a 100% increase in the complexity of that service.
- For every 25% increase in the number of connections in a service, there is a 100% increase in the complexity of that service.

Robert L. Glass

American software engineer and writer

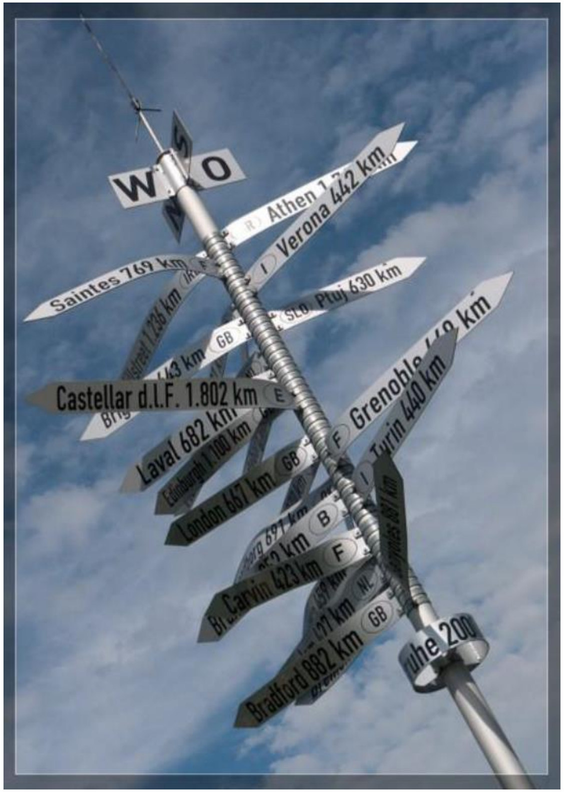


Source: amazon.com

Agenda

Christian-Albrechts-Universität zu Kiel

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings



Outages and failures

Christian-Albrechts-Universität zu Kiel

Observation: Major outages due to complexity

- Skype outage 2007
  → Safety-critical update by Microsoft caused massive rebooting of an enormous number of computers world-wide.
  → Result: Distributed Denial of Service
- Google Mail collapse 2009:
  → Maintenance work within a European data centre caused iterative assignment of responsibility to neighbouring data centres.
  → Result: Cascading breakdowns of computing and data centres.
- E.ON Blackout 2006
  → Ems cable was turned down for passing ship.
  → Result: Cascade of power network breakdowns (DE,NL,BE,FR).

Complexity

Christian-Albrechts-Universität zu Kiel

Observations from these examples:

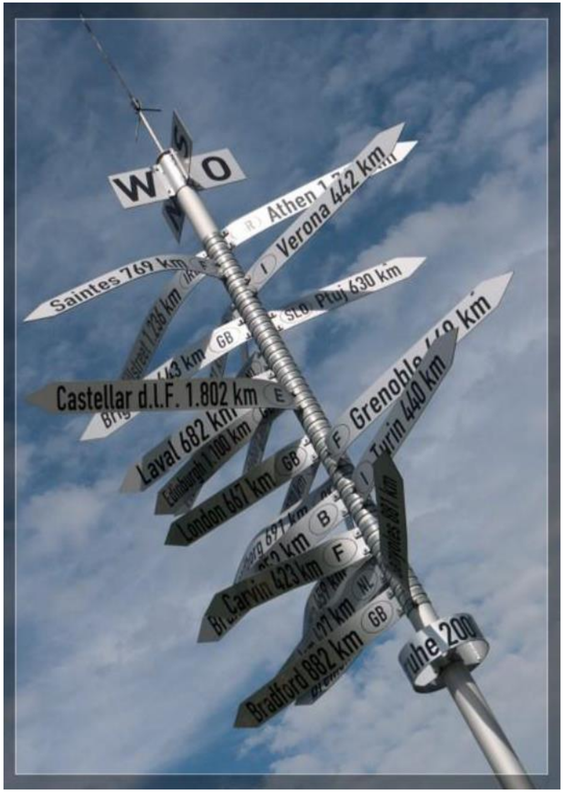
- An exponential increase in complexity observed in different aspects of technical development.
- Increasing interconnectedness, hidden causal dependencies.
- Question: How to master this complexity?
- Traditional concepts, processes, and methods have come to an end.
  → This manifests itself in observable failures and outages and the dramatic increase of administration effort.

Has information and communication technology come to a dead end?
Is there no hope for mastering complexity anymore?

Agenda

Christian-Albrechts-Universität zu Kiel

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings



Nature as inspiration

Complexity

- Complexity is a common phenomenon in nature!
- A variety of well-adapted solutions can be observed.
- Common theme: reduction of complexity by collaboration of autonomous entities.

Natural systems …

- … are typically highly complex systems themselves.
- … have evolved over billions of years.
- … show self-* properties.
- … are changeable and flexible, robust against disturbances, resilient, optimised.

Nature as inspiration

- Not: imitation of natural systems
- But: transfer of basic mechanisms

Nature as inspiration (2)

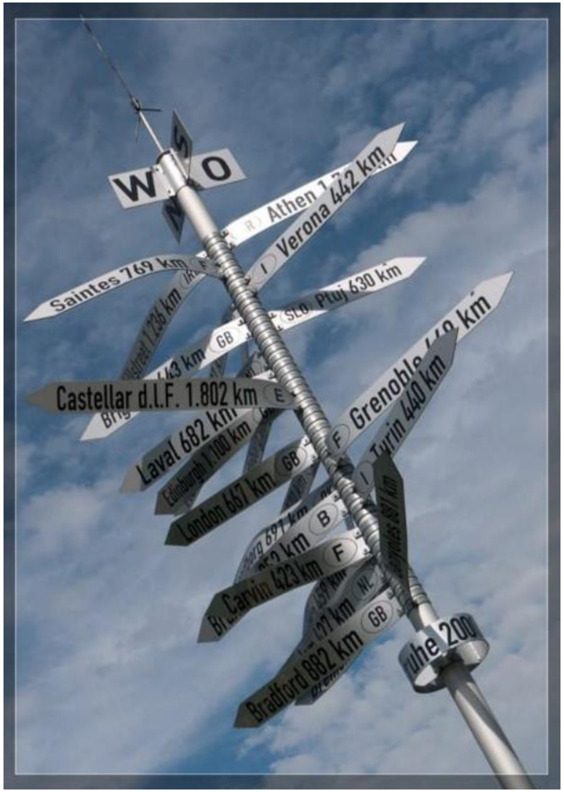
Christian-Albrechts-Universität zu Kiel

How is complexity mastered in nature?

- System consist of potentially large collections of individuals.
- The individuals act autonomously without central control.
- Each individual is characterised by self-* properties: self-learning, self-adaptation, self-protection, and so on.
- Decisions are taken by autonomous entities based on local knowledge.
- Entities interact and cooperate, resulting in a macro-level behaviour of the entire system.
- Some system properties appear as emergent effect.
- Each individual is self-motivated, i.e. it follows its own goals (these goals do not necessarily have to be in line with the entire system).

Agenda

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- **Definition of Organic Computing**
- Aspects
- Conclusion
- Further readings



Organic Computing

Christian-Albrechts-Universität zu Kiel

A paradigm shift in system engineering

- Goal:
  - Move traditional design-time decisions to runtime.
  - From engineers into the responsibility of systems themselves.
- Build collections of smaller systems instead of monolithic structures.
- Allow for autonomous decisions.
- Base these decisions on local knowledge without having access to a global world model.
- Let the distributed entities interact and cooperate with each other.
- Establish a feedback mechanism for each entity: learn from past actions.

Organic Computing (2)

A brief definition of Organic Computing

- Organic systems consist of autonomous sub-systems:
  - These sub-systems possess sensors and actuators.
  - Sub-systems interact with each other and the environment.
  - No global (or: system-wide) control necessary.
- The resulting interconnected OC systems have to:
  - organise themselves,
  - be adaptive and flexible, and
  - learn the optimal behaviour – in order to be able to adapt themselves autonomously to previously unknown situations.
- Most of the decisions are taken based on local knowledge and without user / system-wide influence.

OC means to move design-time decisions to runtime!

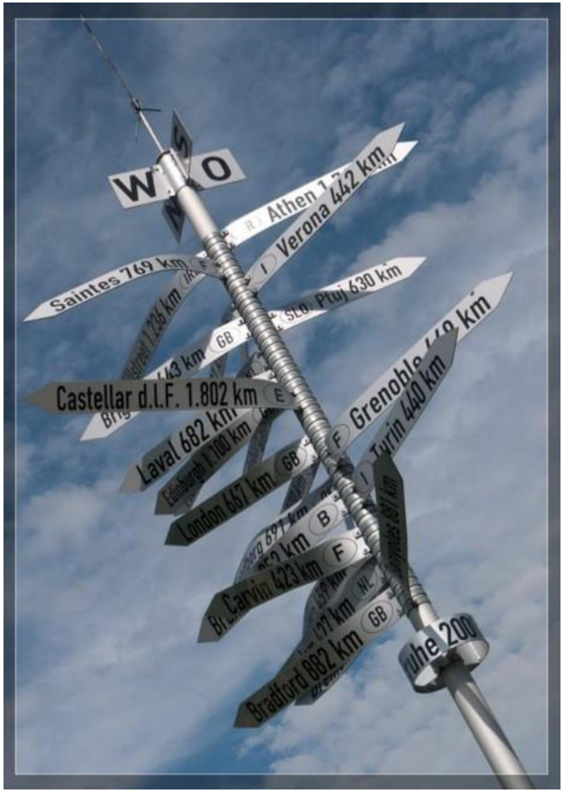
Intelligent Systems and Organic Computing

- OC is used as a synonym for “engineering intelligent systems“ in the context of this lecture
- There are several initiatives with the same / a similar motivation or scope, e.g.:
  - Autonomic Computing
  - Proactive Computing
  - Complex Adaptive Systems
  - Self-aware Systems
  - (Multi-Agent Systems)
  - …
- They all have in common that an individual system perceives, analyses, and acts upon gathered information of the environment by making use of machine learning techniques.

Agenda

Christian-Albrechts-Universität zu Kiel

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- **Aspects**
- Conclusion
- Further readings



Aspects of Organic Computing

Christian-Albrechts-Universität zu Kiel

I. Nature as inspiration

- Natural and social systems are perfect examples of how complexity can be mastered by self-organisation and self-adaptation.
- Understand basic principles of such systems which we can adapt and transfer to utilise them in a technical context.

II. Systems

- Organic Computing is about systems engineering.
- Define what a system is and how it is organised in general.
- Clarify what complexity means.

III. Terminology

- Specification of basic terms such as self-organisation, robustness, autonomy.
- Quantification methods for important aspects such as emergence, degree of self-organisation or autonomy.

Aspects of Organic Computing (2)

Christian-Albrechts-Universität zu Kiel

IV. Building Organic Computing systems

- Design concepts for individual context-aware systems with organic properties
- Macro view on ecosystems based on social mechanisms

V. Paradigm shift

- Organic Computing means moving traditional design-time decisions to runtime
- Summary of implications of this transfer
- Impact on design processes: Organic Computing meta design process

VI. Basic methods

- Specialised technology is needed to establish self-* properties
- Most importantly: autonomous learning and knowledge modelling at runtime
- In addition: Techniques for, e.g., optimisation (“good enough” rather than “optimal”), interaction schemes, and detection of mutual influences

Aspects of Organic Computing (3)

VII. Sensor-based perception

- The current conditions are perceived by means of sensors
- Sensor information is error-prone and uncertain.
- This information needs to be processed, transformed, analysed, and assessed before using it in the decision processes.
- Conditions need to be classified, clustering processes are required, and unexpected behaviour must be assessed by machine learning techniques.
- Sophisticated models are learned, utilised, and maintained throughout the entire operation process of a system.

VIII. Applications

- A technical system serves a specific purpose.
- Examples for real-world systems based on Organic Computing technology.
- Application fields: road traffic, data communication, energy grid, distributed computing, smart cameras, and others.

Aspects of Organic Computing (3)

Christian-Albrechts-Universität zu Kiel

IX. Related fields

- A research initiative seldom comes out of nowhere.
- It usually has its roots in some preceding endeavours, is typically accompanied by other parallel research efforts that share the same motivation and parts of the ideas, and eventually provides the fundament for some subsequent novel ideas and concepts.

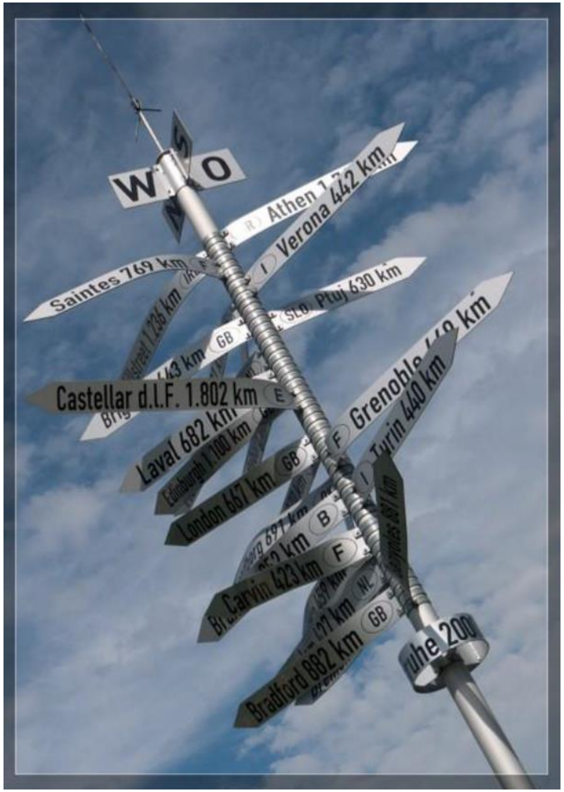
X. Outlook

- Trend towards more complexity is still unbroken.
- What are current and future challenges for Organic Computing research?

Agenda

Christian-Albrechts-Universität zu Kiel

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- **Conclusion**
- Further readings



A brief history of Organic Computing

Christian-Albrechts-Universität zu Kiel

- 1999: von der Malsburg coined the term “Organic Computing” as combination of neuro sciences, molecular biology, and software.
- 2000: Tennenhouse presented his vision of “Proactive Computing“
- 2002: Workshop on future topics in technical computer science
- 2002: Forrester Research study on “Organic IT“
- 2003: Autonomic Computing proposed by IBM
- 2003 and 2004: OC Philosophy Workshops at Kloster Irsee
- 2005: Special Priority Programme 1183 “Organic Computing“ funded by DFG: Schmeck (KIT), Müller-Schloer (Hannover), and Ungerer (Augsburg)
- 2006: First Dagstuhl Seminar on Organic Computing
- 2006: First professorship on Organic Computing in Hannover
- 2009: Research Unit “Trustworthy Organic Computing Systems“ funded by DFG
- 2010: Collaborative Research Centre “Invasive Computing“: Erlangen
- 2012: First full professorship on Organic Computing in Augsburg
- 2012: Establishment of “Special Interest Group“ within the GI

C | A | U
Christian-Albrechts-Universität zu Kiel

Well, this has been the initial plan…

Starting point

Natural processes as inspiration

- Ants
- Dissipative structures
- Swarms
- Brain
- …

Phase 1

Organic Computing principles

Emergence Self-*

- Autonomy
- Self-*
- Emergence
- Awareness
- Cooperation
- Competition

Phase 2

Organic Computing technology

Organic Computing toolbox

- Basic technologies
- Observer/Controller
- Guarding
- Helper Threads
- Embedded learning
- Complex system architectures

Phase 3

Technical applications

- Traffic control
- Smart Home
- Energy grid
- Factory
- Health and care
- …

Conclusion

Christian-Albrechts-Universität zu Kiel

This chapter

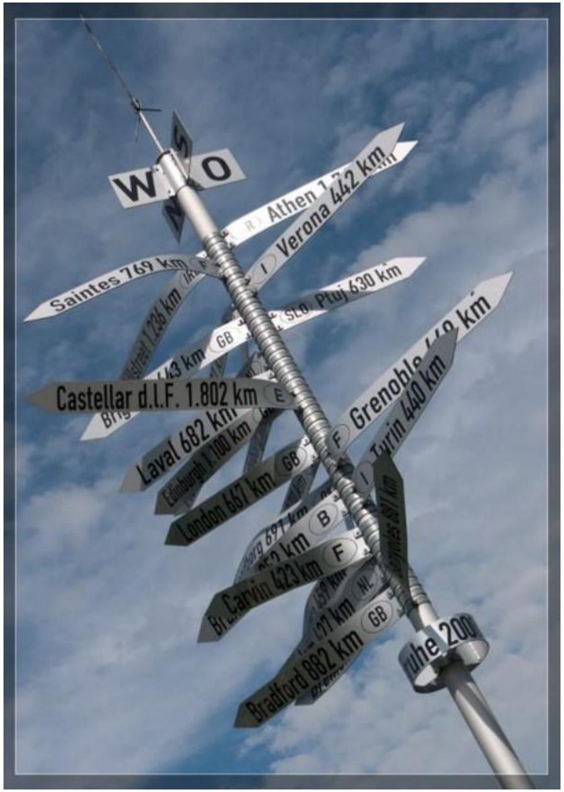
- Outlined to the trend towards ubiquitous interconnectedness.
- Illustrated how complexity arises in technical systems.
- Named examples where complexity has caused outages.
- Explained how Organic Computing can learn from nature.
- Defined what Organic Computing is about
- Discussed the most important aspects of Organic Computing and outlined the lecture.

At the end of this chapter, students should be able to:

- Describe the motivation for Organic Computing.
- Give examples for raising complexity in information and communication systems.
- Explain why natural processes can serve as inspiration to master complexity.
- Summarise which aspects are covered by Organic Computing.

Agenda

- Motivation
- Examples for growing complexity
- Prominent outages with severe impact
- Nature as inspiration
- Definition of Organic Computing
- Aspects
- Conclusion
- Further readings

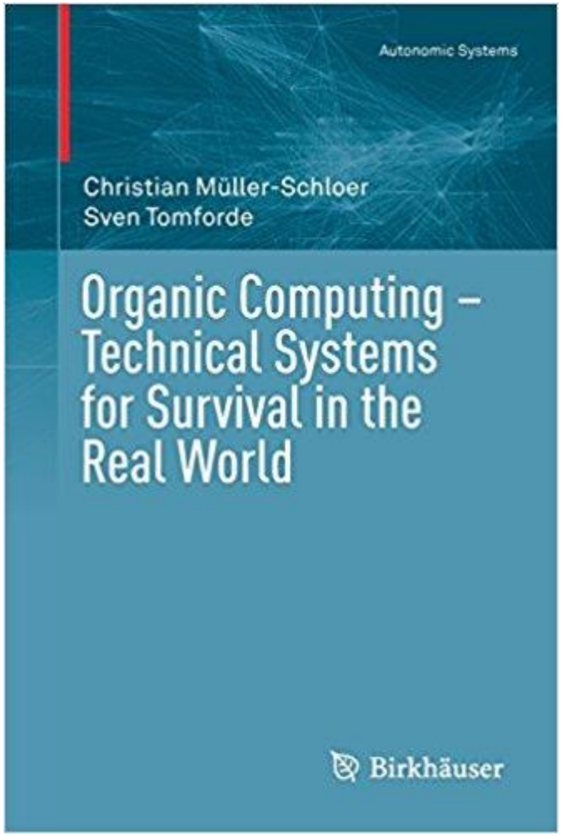


Further readings

Christian-Albrechts-Universität zu Kiel

"Organic Computing" and its characteristics are defined based on the book:

- Christian Müller-Schloer and Sven Tomforde: Organic Computing – Technical Systems for Survival in the Real World, Birkhäuser Verlag, Basel, 2018, ISBN 978-3319684765



Further readings (2)

Christian-Albrechts-Universität zu Kiel

Further sources:

- Müller-Schloer, Christian, and Hartmut Schmeck. "Organic Computing: Quo vadis?." Organic Computing—A Paradigm Shift for Complex Systems. Springer, Basel, 2011. 615-627.
- Tomforde, Sven, Bernhard Sick, and Christian Müller-Schloer. "Spotlight on Organic Computing." Herausgegeben von Prof. Dr. Bernhard Sick, Universität Kassel (2017): pp. 1 – 8.
- Krupitzer, Christian, and Sven Tomforde. "The Organic Computing Doctoral Dissertation Colloquium: Status and Overview in 2019." INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik–Informatik für Gesellschaft (Workshop-Beiträge). Gesellschaft für Informatik eV, 2019.

End

- Questions….?