Implications of Organic Computing for Reconfigurable Computing (RTR, DRA,…)

Hartmut Schmeck

Institut AIFB

Universität Karlsruhe (TH)

Overview

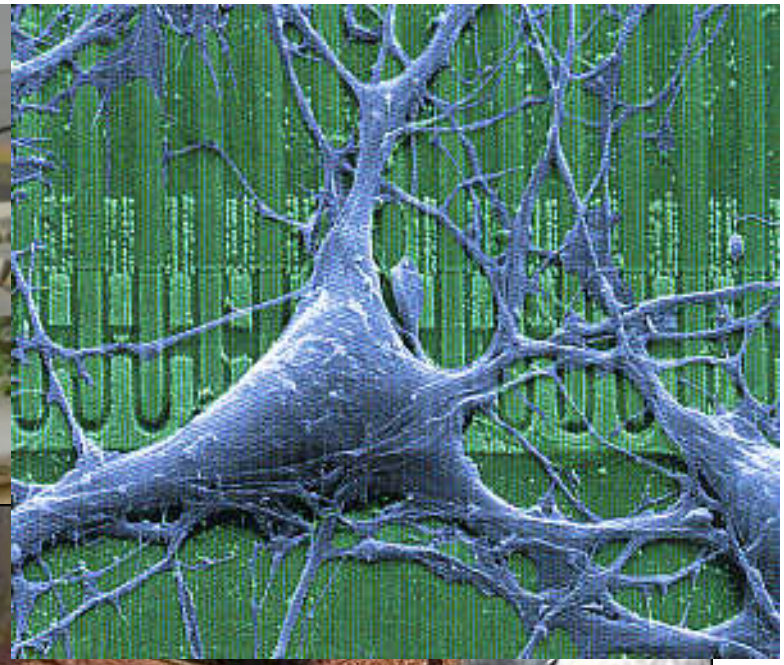
- Organic Computing
  - Motivation
  - Vision
- Challenges
- Implications for Reconfigurable Computing
- Conclusion

Talk focusses on ideas, only partially on work already done.

Inspired by

- **„Organic Computing Initiative“ (GI/ITG)**
- **Priority research programme „Organic Computing“**

What Organic Computing is *not about*



So, what is it about?

- Collections of **intelligent (embedded) systems** (scenarios **smart house, car, office, factory, shop, healthcare**,…).
- Potentially **unlimited networks (**large numbers, mobility)
- **Spontaneous local interaction** leading to unexpected global behaviour (**emergent phenomena** as a result of **self-organisation**)
- Necessity for providing **robust service** in dynamically changing environments.
- Necessity for **flexible behaviour** due to varying external objectives.
- **Design, management, and acceptance problems** with respect to increasingly **complex systems**.

⇒**We have to come up with good ideas for**

- **designing and managing unlimited, dynamical networks of intelligent devices,**
- **making best use of the available technology.**

Propositions and Vision
(GI/ITG-Initiative on Organic Computing, 2003)

- Information technology is moving towards ubiquitous networked computer systems.
- Complex ubiquitous systems need new concepts for organization and user interfaces to remain manageable and controllable.
- Future computer systems have to be designed with respect to human needs.
- Future computer systems have to be trustworthy.
- Future computer systems have to be robust, adaptive, and flexible.
- Systems having these properties will be life-like.
  We call them *Organic Computer Systems*.

Organic Computing

It is not the question,
whether adaptive and self-organising systems
will emerge,
but ***how*** they will be designed.

Vision for System Architecture >2010

- **Organic Computer Systems**
  - will possess lifelike properties.
  - will consist of autonomic and cooperating sub systems and will work, as much as possible, in a self-organised way.
  - will adapt to human needs,
  - will be controlled by objectives,
  - will provide customized service in a user-friendly way,
  - will be trustworthy

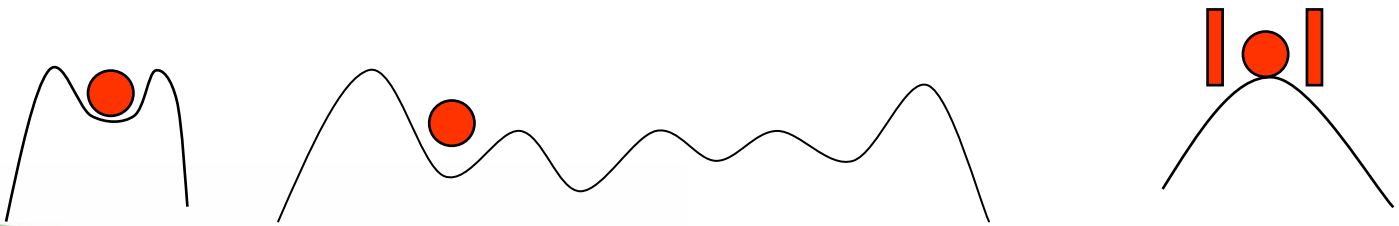
- **Self-organisation** allows for adaptive and context dependent behaviour:
  - self-configuring
  - self-optimizing
  - self-healing
  - self-protecting
  - self-explaining
  - …

Background, history

- Ubiquitous Computing (Mark Weiser, 1991)
- Intelligent Autonomous Systems (Siemens, ~1994)
- Pervasive Computing (~1996)
- Organic Computing (C. von der Malsburg 2001)
  basis: Neuro sciences, Molecular Biology
- Autonomic Computing (IBM, ~ 2001)
  self-x properties for server architectures
- Evo-Architecture for cars (DaimlerChrysler, ~2001)
- Organic IT (Forrester Res. April 2002)
  enterprise server architectures
- Organic Computing (GI/ITG 2002, DFG 2004)
  Future Workshops of GI,
  System architecture for I&C technology beyond 2010

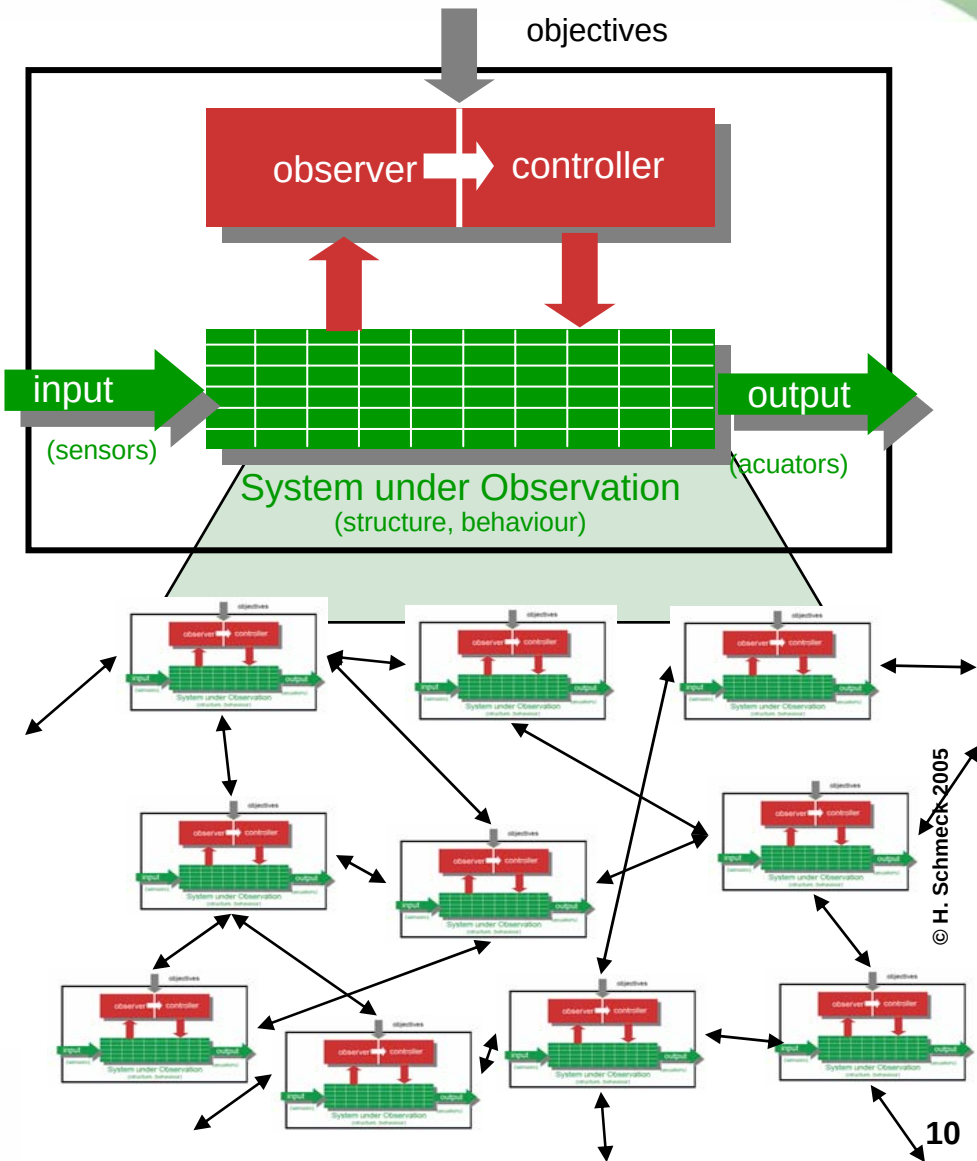
Challenges for system design and architecture

- Provide systems with sufficiently large degrees of freedom for **adapting in a self-organized way** to different requirements.
- Systems have to be aware of
  - what type of service they can provide,
  - what type of service they need from others,
  - what the current environment wants to get done.
- Systems should have a “desire” to be active (→incentives?).
- Systems should be robust and adaptive wrt external disturbances
- Systems should be flexible wrt changed requirements
- There will be a need for “controlled self-organization”.



Control of self-organised behaviour: Observer-Controller Architecture

- Observe and measure the current state of an organic system.
- Detect deviations from “standard behaviour”.
- Initiate adequate actions in order to (re-)satisfy system constraints.
- Possibly multiple layers, distributed.
- What will be influenced/controlled?
  - environment
  - communication
  - functions of components
  - …
- Who will influence/control?



Consequences for research

Self-organizing technical systems are becoming a key topic for academic and industrial research.

What do we need?

- Get inspired by nature (ants, bees, termites, fish, birds,…):
  Evolutionary Algorithms, Ant Colony Optimization,
  Swarm intelligence, DodOrg,…
- Multi-Agent systems
- Observer/Controller-Architectures
  multithreaded processors, concepts for organic middleware
- Results from control theory (model predictive control??)
- **Reconfigurable computing systems,**
  adaptive / autonomic SoC
- …

Implications for Reconfigurable Computing (1)

Degrees of freedom:

- RTR / DRA / CRC,… should provide necessary degrees of freedom to allow for adaptivity and flexibility.

Self-configuration:

- We know how to configure and reconfigure, even dynamically at runtime, but how can the device do this by itself?
  - Provide library of various situation-dependent behaviours, store it locally, and let classifier choose appropriate bitstream.

Control by objectives:

- Specify behaviour, not implementation/program/configuration
- Needs ways of (self-)generating appropriate configurations
  → **evolvable hardware?**
    - Time for generation of acceptable configurations?
    - Methods for testing / performance evaluation?
    - Validation ?

Implications for Reconfigurable Computing (2)

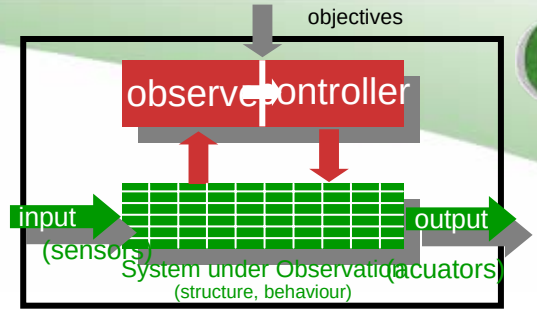
Robustness:

- Make designs robust against occurrence of faults
  (could be done by using (standard) methods of fault tolerance, does not necessarily need reconfiguration, but might profit from it).
- Identify parts of the design that have to be changed/replaced dependent on certain parameters of the environment.
  → *needs observer/controller-architecture and RTR*

Flexibility:

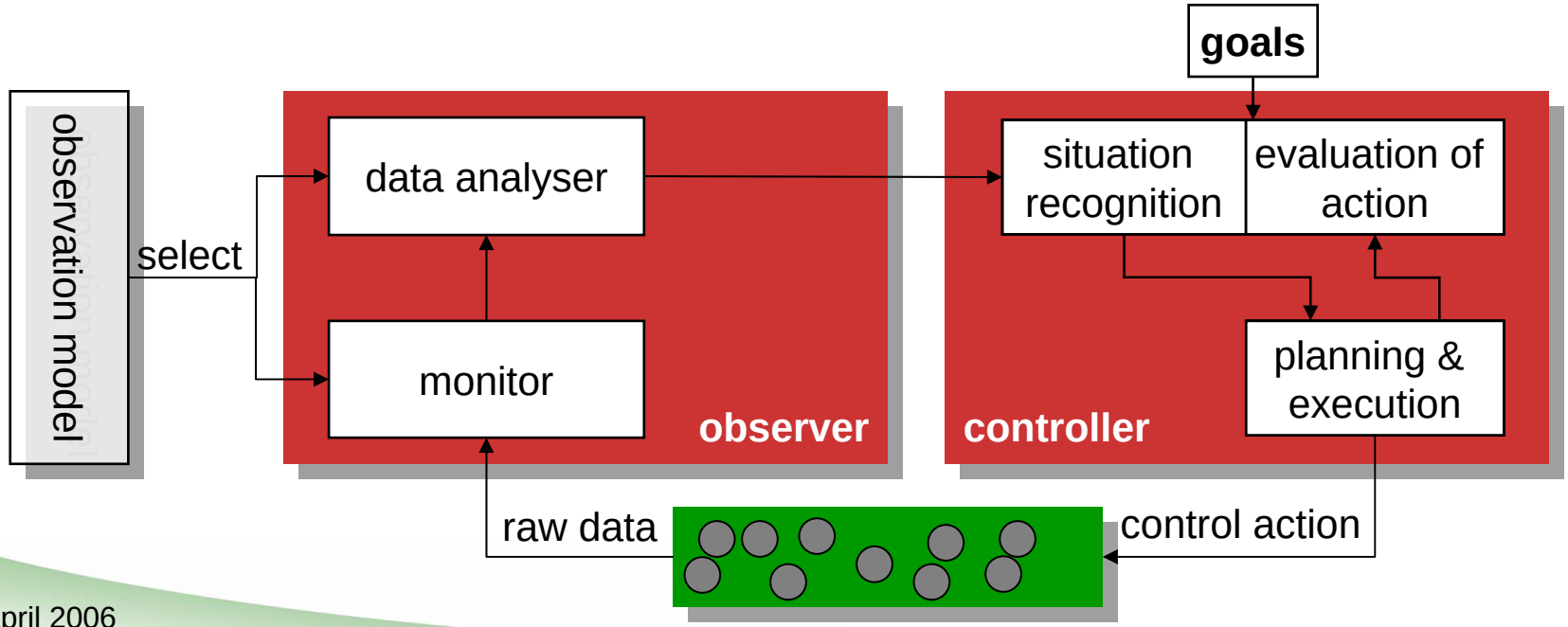
- Selection from predetermined alternative configurations.
- Needs observer to detect changes in environmental requirements that imply necessity for reconfiguration.

Implications for Reconfigurable Computing (3)



Observer/Controller Architecture:

- Needs observation models
  - Selection of parameters characterizing environment and system
- Needs various methods for filtering and analysis, decision support.
- **Needs reconfigurable implementation**.



Implications for Reconfigurable Computing (4)

Trust:

- Support methodology to establish trust in reliable functionality of the system.
- Provide certification of correctness and integrity of configuration bitstreams.

Self-protection, security:

- Provide mechanisms for checking and verifying the identity of an FPGA.
- Provide mechanisms for checking the integrity of configuration bitstreams.
  (needs on-chip observer-controller architecture, e.g. see Adi's ideas on FPGA-security)

Conclusion

- Organic Computing is an urgently needed vision for system architecture design and management.
- Organic Computing needs (self-)reconfiguration, in particular runtime reconfiguration (and dynamically adaptable behaviours)
- A key concept of organic computing is self-organisation: it will not be possible to achieve adaptive behaviour if it relies on external management and control.
- Reconfigurable Computing could contribute in several central ways to achieve organic behaviour of technical systems.
- **Or, might pentium processors provide the same in a much more cost-effective way?**

Thanks for your attention!

Discussion?