Quantitative Emergence (QE)

DFG SPP 1183 Organic Computing (www.aifb.uni-karlsruhe.de/EffAlg/Projekt/otcqe)

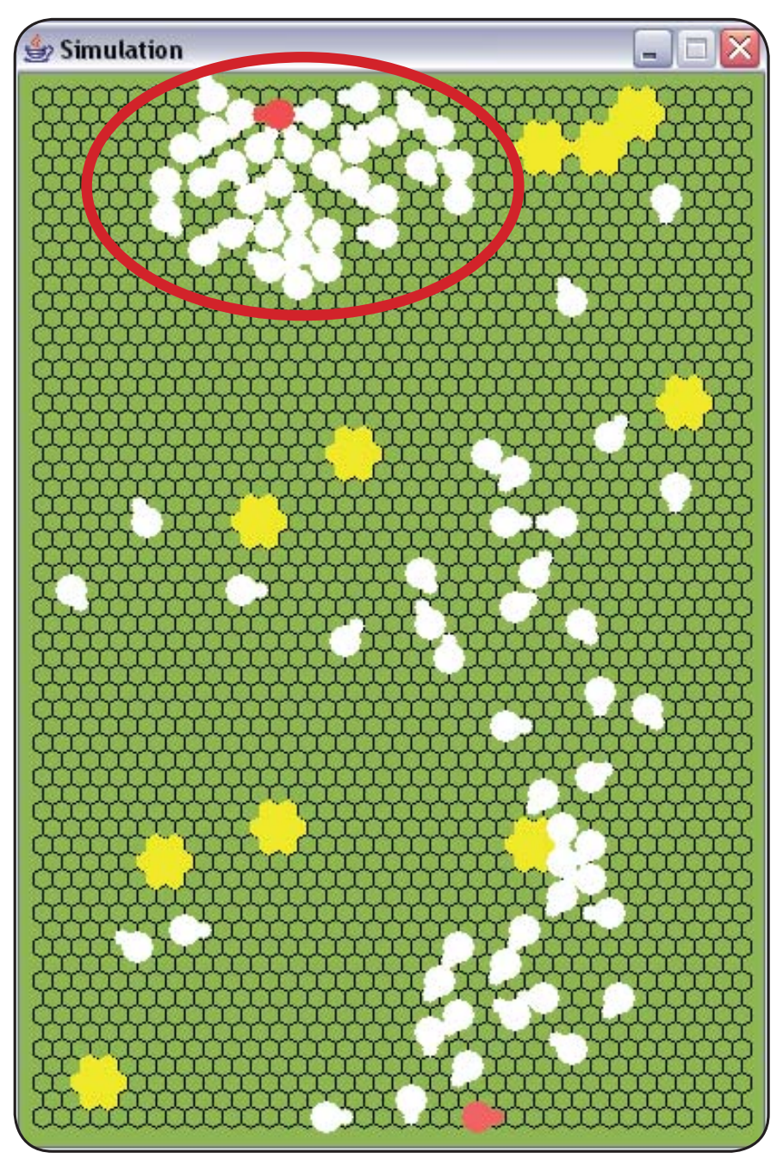
A research cooperation between the Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology (KIT), and the Institute of Systems Engineering – System- und Rechnerarchitektur, Leibniz Universität Hannover.

Prof. Dr. Hartmut Schmeck (Karlsruhe), ✉ schmeck@aifb.uni-karlsruhe.de
Prof. Dr.-Ing. Christian Müller-Schloer (Hannover), ✉ cms@sra.uni-hannover.de
PD Dr. Jürgen Branke (Karlsruhe), ✉ branke@aifb.uni-karlsruhe.de
Dipl.-Ing. Moez Mnif (Hannover), ✉ mnif@sra.uni-hannover.de, ☎ +49 (511) 762 19749
Dipl.-Wi.-Ing. Urban Richter (Karlsruhe), ✉ uri@aifb.uni-karlsruhe.de, ☎ +49 (721) 608 6586

What is emergence?

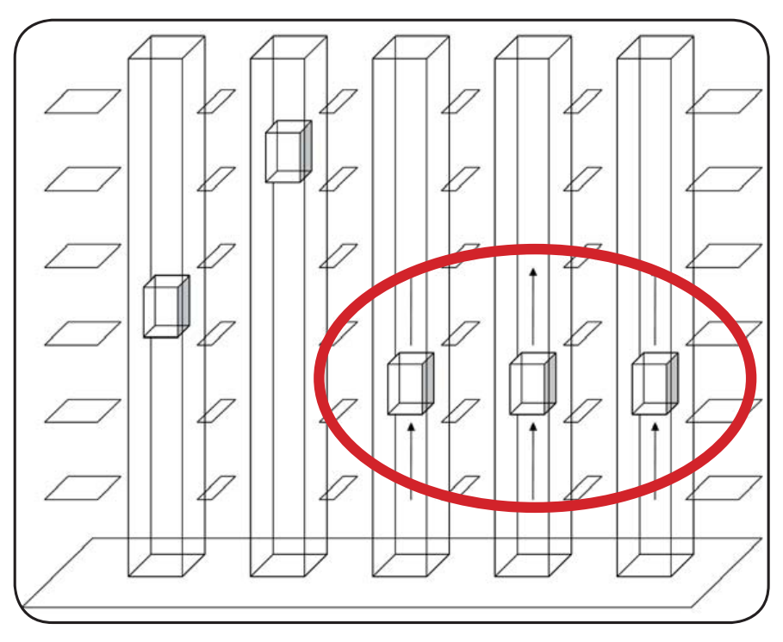
- “The whole is more than the sum of its parts.”
- Many examples in nature, e. g. flock of birds.
- Humans decide intuitively on the occurrence of emergence.
- Precondition: A large population of interacting elements without central control.

→ Emergence = self-organised order



Goals

- How to build self-organising technical systems?
- Automate the recognition of emergent behaviour. (formation of order pattern)
- Need of metrics to quantify emergence.
- Development of mechanisms to control emergence.

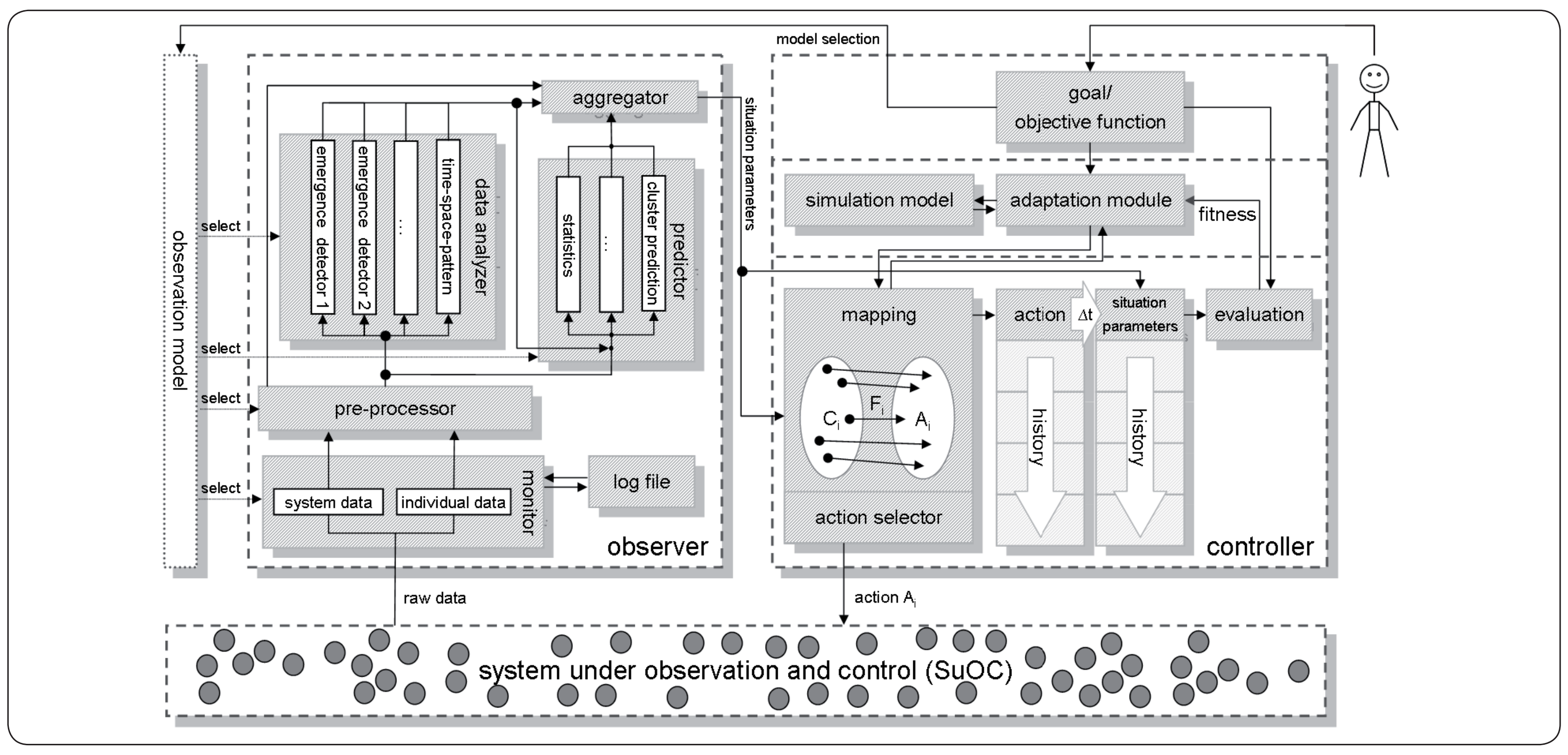


Emergent phenomena: Order pattern in a chemical reaction, clustering behaviour of densley packed chickens in cages, and bunching effect of an elevator group.

Observer/controller architecture

- The system under observation and control (SuOC) consists of a set of interacting intelligent autonomous units.
- The observer measures and reports the system behaviour to the controller.
- The controller applies adequate changes to the SuOC with respect to concrete measurement to achieve a given goal.

→ Controlled emergent behaviour



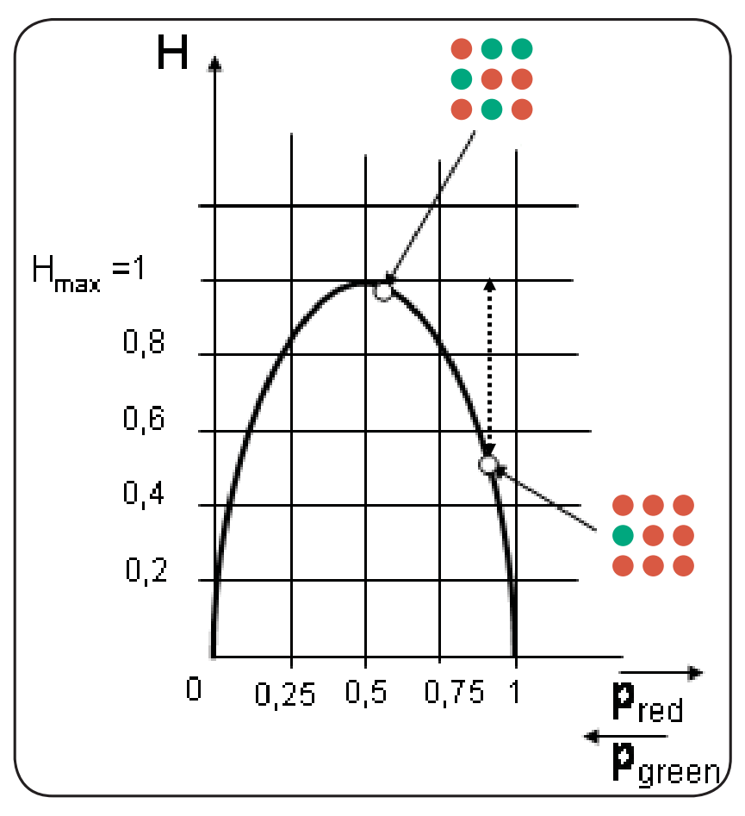
Computation of emergence values

- System S with N elements
- Select an attribute A common to all system elements.
- Build a relative frequency of the occurrence of each attribute value.

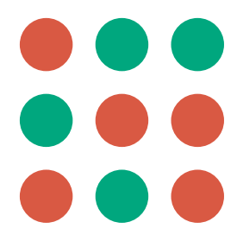
→ Probability distribution

- Compute the entropy according to Shannon.

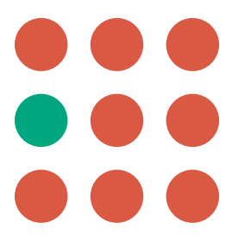
$$H_A = -\sum_j p_j \cdot \mathrm{ld} p_j$$



→ Example: A system of 9 balls, select the attribute color.



$$F_{red} = \tfrac{5}{9},\ F_{green} = \tfrac{4}{9}$$
$$H_{colour} = 0.991$$



$$F_{red} = \tfrac{8}{9},\ F_{green} = \tfrac{1}{9}$$
$$H_{colour} = 0.503$$

- Compute the emergence values

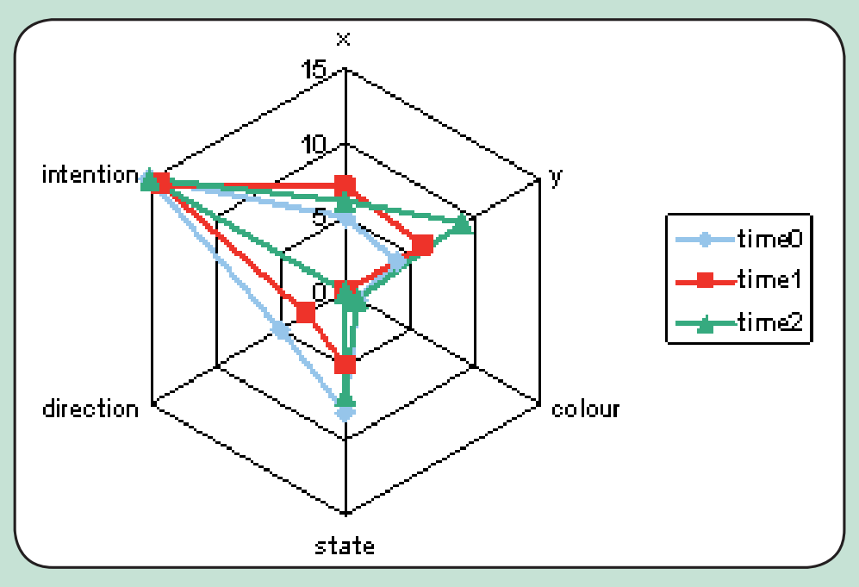
$$M_A = \Delta H_{emergence} = H_{max} - H_A$$

or the relative emergence values.

$$m_A = \frac{H_{max} - H_A}{H_{max}}$$

- A system is characterised by a vector of emergence values.
- Emergence fingerprint represents the order pattern.

→ Kiviat graph



Controlled self-organisation

- Influence the system such that a desired emergent behaviour appears.
- Disrupt an undesired emergent behaviour as quickly and efficiently as possible.
- Construct the system in a way that no undesired emergent behaviour can develop.

→ Three general types of control

- Influencing the local decision rules of the simple agents modifies the local behaviour of the individual.
- Influencing the system structure (communication system, neighbourhoods) modifies interaction patterns.
- Influencing the environment allows indirect control of the SuOC.

→ Cope with the complexity of today’s technical systems by the use of distributed self-organised systems.

→ Characterise the development of self-organisation and emergence with concrete measurement categories to get a better understanding of these phenomena.

→ Guide a self-organising system to reach a given target and to avoid unwanted behaviour.

→ Realisation of self-organising technical systems, which are at the same time reliable, robust, and adaptive.

Deutsche Forschungsgemeinschaft
DFG