

From Prescriptive Programming of Solid-State Devices to Orchestrated Self-organisation of Informed Matter

Klaus-Peter Zauner

School of Electronics and Computer Science,
University of Southampton, SO17 1BJ, United Kingdom
kpz@ecs.soton.ac.uk
www.ecs.soton.ac.uk/people/kpz/

Abstract. Achieving real-time response to complex, ambiguous, high-bandwidth data is impractical with conventional programming. Only the narrow class of compressible input-output maps can be specified with feasibly sized programs. Present computing concepts enforce formalisms that are arbitrary from the perspective of the physics underlying their implementation. Efficient physical realizations are embarrassed by the need to implement the rigidly specified instructions requisite for programmable systems. The conventional paradigm of erecting strong constraints and potential barriers that narrowly prescribe structure and precisely control system state needs to be complemented with a new approach that relinquishes detailed control and reckons with autonomous building blocks. Brittle prescriptive control will need to be replaced with resilient self-organisation to approach the robustness and efficiency afforded by natural systems. Structure-function self-consistency will be key to the spontaneous generation of functional architectures that can harness novel molecular and nano materials in an effective way for increased computational power.

1 Commanding the Quasi-universal Machine

The common conventional computer is an approximation of a hypothetical universal machine [1] limited by memory and speed constraints. Universal machines are generally believed to be in principle able to compute any computable function and are commonly used to define what can effectively be computed [2]. Correspondingly it is assumed that if processing speed and memory space of computers would indefinitely continue to increase, any computable information processing problem would eventually come within reach of practical devices. Accordingly time and space complexity of computation has been studied in detail [3] and technological advances have focused on memory capacity and switching speed [4]. But along with this there is another factor that limits realizable computing devices: the length of the program required to communicate a desired behaviour to the device [5]. The length of this program is limited by the state space of the

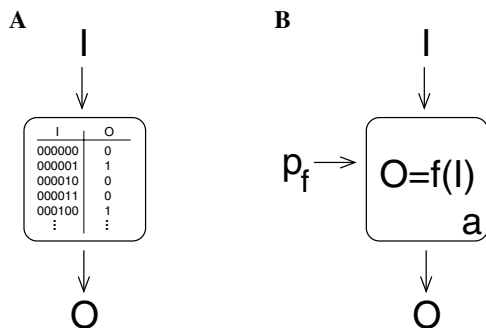


Fig. 1. Communicating a desired input-output map to a machine. The input-output map can in principle be thought of as a potentially very large lookup table that associates an output response with every input that can be discerned by the machine (A). For n bit input patterns (I) and a m bit output response (O) the, number of possible maps is 2^{m2^n} . To implement an arbitrary one of these maps on a quasi-universal machine, the mapping f has to be specified by the program p with respect of machine architecture a (B). Selecting an arbitrary map from the set of possible maps may require a specification of length: $\log_2 [2^{m2^n}] = m2^n$. Even for moderate pattern recognition problems (e.g., classifying low resolution images) the program length required for most mappings is impractical [6]

device and the capacity of the programmers. Even small problems can exhaust these practical limitations (cf. figure 1). As a consequence conventional computing architectures are restricted to the implementation of highly compressible input-output transforms [7]. The set of compressible maps is a small subset of the potential input-output functions—most behaviours cannot be programmed. Whether the inaccessible mappings would be useful for practical applications awaits further investigation. For complex ambiguous pattern recognition problems, a domain where organisms cope much better than existing technology, mappings of limited compressibility may be valuable.

The picture painted in figure 1 is optimistic with regard to the number of maps that can be implemented by a program of given size. It assumes that the machine architecture is not degenerate, i.e., any two programs that differ by at least one bit will give rise to distinct input-output maps. In practice, computing architectures often map many programs to the same input-output transformation. The transformation is usually implemented as a programmed sequence of elementary operations. The essence of each operation is to selectively discard information [8, 9]. As the number of elementary operations grows it becomes increasingly likely that all information regarding the input signals is dissipated and as a consequence the output of the computation is constant [10, 11]. Accordingly, the degeneracy of the mapping from program to input-output transform raises with increased program length in conventional architectures and the number input-output transforms accessible through programming is further reduced.

For a machine to be programmable, additional restrictions come into play. Programming is here equated with an engineering approach in which mental

conception precedes physical creation (cf. [12]). It necessitates the possibility for the programmer to anticipate the actions of the available elementary operations. Only if the function of the elementary operations can be foreseen by the programmer can a desired input-output map be implemented by incrementally composing a program. Accordingly, the machine's architecture has to adhere to a fixed, finite user manual to facilitate programming. To achieve this, numerous potential interactions among the components of the machine need to be suppressed [13]. Programmability is achieved by using relatively large networks of components with fixed behaviour. This however does not allow for the efficiency afforded by networks of context sensitive components [14].

As outlined above, conventional programming is not always the most suitable way of implementing an input-output map. Some maps cannot be compressed into programs of practical length, and the need for programmability precludes hardware designs that elicit functionality from a minimum of material.

2 Learning, Adaptation, Self-organisation

Programmability is not a strict requirement for information processing systems. The enviable computing capabilities of cells and organisms are not implemented by programs. Non-programmed information processing is not limited to nature. Artificial neural networks provide a technological example of a system where programmability has been given up for the benefit of parallel operation [15]. Freeing the computing architecture from the need for predictable function of elementary components opens up new design degrees of freedom. Firstly, the fan-in for an elementary component could be increased by orders of magnitude. It may be interesting to note that neurons in the cortex of the mouse have on average 8000 input lines [16]. Secondly, there is no need for all components to operate according to identical specifications. This opens a path to broadening the material basis of computation by allowing for computational substrates the structure of which cannot be controlled in detail. And likewise, thirdly, the operation of the elementary components can depend on their context in the architecture, thus greatly increasing the number of interactions among the components that can be recruited for signal fusion.

Utilising these design degrees of freedom requires the development of new training algorithms for the resulting networks. Evolutionary methods that take the performance of a network as prediction for the performance of a randomly modified network are particularly suitable. They can cope with the complexity and inhomogeneity of architectures based on context sensitive components and benefit from the increased dimensionality in such networks [17, 18, 19]. Clearly, giving up programmability will not by itself increase the number of input-output transforms that can be implemented on a given system. The feasible length of training for the system draws the limit. However, the complexity of the input-output transformations is not restricted by the need for compact specification.

3 Orchestrating Informed Matter

Techniques for producing biomaterials and manufacturing nano-materials are rapidly developing. We already see materials with unprecedented characteristics arriving at an increasing rate. But so far computer science is not on a path to harnessing these new materials for increased computational power. Training the materials to act as logic gates is unlikely to be fruitful.

Present computing concepts enforce formalisms that are arbitrary from the perspective of the physics underlying their implementation. Nature's gadgets process information in starkly different ways than do conventionally programmed machines [20]. They exploit the physics of the materials directly and arrive at problem solutions driven by free energy minimisation while current computer systems are coerced by high potential barriers to follow a narrowly prescribed, contrived course of computation [21]. The latter is only possible in a macro-physical device and comes at the cost of using a large fraction of the material of the processing device for enforcing adherence to the formalism rather than actual information processing. As devices shrink to micro-physical scale it becomes increasingly difficult to isolate the operation of the device from the physics of their implementation. However, as outlined in the preceding two sections, information processing can be implemented without a formalism that abstracts away from the underlying computing substrate.

Nature provides a large collection of implementations that employ computation driven directly by physics for sophisticated information processing tasks. This mode of computation is most clearly demonstrated by single-cell systems. A seed weighs time series of multiple ambiguous sensory signals to make the decision to grow, a vital decision from which it cannot retract [22]. Bacteria assess their chemical surroundings to adjust their motions in accordance with a complex trade-off among attractive and repulsive factors [23, 24]. Being too small to afford isolation from the micro-physics of their material components the information processing operations in these systems necessarily have to follow a course of computation inherent in molecular interactions.

In physics-driven information processing architectures the structure of an architecture and its processing function are inseparable. Accordingly physics-driven information processing is closely related to self-organisation. Self-organisation is taken here as a process that forms an organised functional structure and is essentially supported by the components of the structure themselves rather than a process conducted by an external infrastructure. The possibilities of self-organisation are exemplified by biological growth and development. Self-assembly of molecular structures is a relatively simple self-organisation process in the aforementioned sense. The formation of viruses [25] is the prototypic example of molecular self-assembly, but numerous functional structures in the cell, e.g., ribosomes [26] form in a similar fashion. Artificial self-assembly systems have been designed with inorganic [27], organic [28], and bio-molecules [29, 30] and provide a potential route to realizing artificial molecular information processors [31].

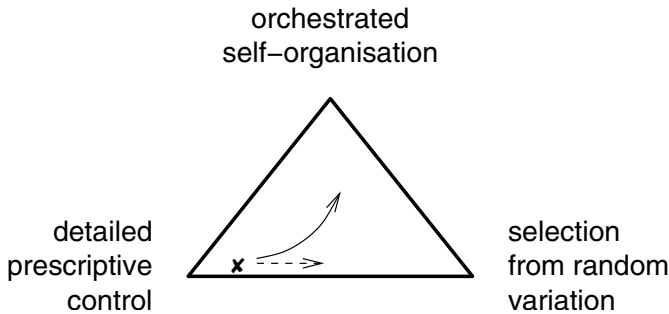


Fig. 2. Implementation paradigms for a computational device. Present conventional computer technology is indicated near the lower left corner. Random variation enters unintentionally in the production process. With increasing miniaturisation control will become increasingly more difficult (dashed arrow). Resilient architectures that can cope with wide component variation and the deliberate use of self-organisation processes provide the most likely path to complexification of computing architectures (bent arrow)

The practical implementation of an input-output map can adhere in varying degrees to different paradigms as illustrated in figure 2.

As engineering extends to base-components at nano meter scale, the fine-grained prescriptive control familiar from the macro-physical world becomes increasingly difficult to achieve. Accordingly the proportion of products conforming to specification is falling. To some extent it is possible to compensate for the loss of control during production in a subsequent selection procedure, as is common practice. Selecting functional structures from a pool of randomly created structures provides a first approach to nano-materials where detailed control is not feasible or not economical. If the process of structure formation is repeatable then the selection from random variation can be iterated for evolutionary progress.

A key driver for miniaturisation, however, is its potential for highly complex systems within a small volume. A production process relying on selection alone will not deliver such systems for two reasons. Firstly, the falling proportion of functional components will make the selection approach (indicated by the dashed arrow in figure 2) increasingly inefficient, particularly if systems with high component count are desired. Secondly, for selecting the products that exhibit correct functionality it is necessary to identify them. Doing so by testing incurs a high cost and exhaustive testing becomes prohibitive with rising system complexity.

To arrive at economic nano-scale systems of high complexity it is necessary to leave the implementation approach indicated by the baseline of the triangle in figure 2. One possible solution lies in the self-assembly properties of suitably formed base components. The challenge is to design base components in such a way that they will spontaneously form a desired architecture. This course of engineering (indicated by the bend arrow in figure 2) will require the considera-

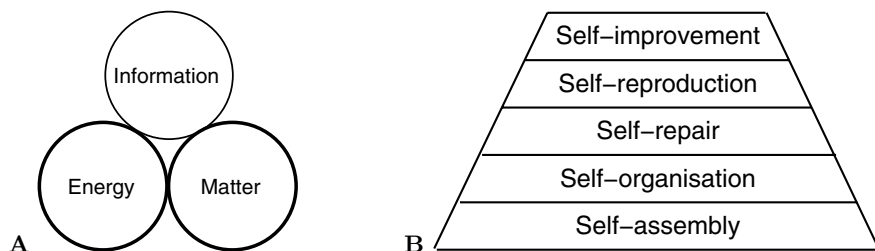


Fig. 3. To engineer self-organised systems, the traditional design considerations of energy and matter need to be augmented to include information (**A**). The potential gain of adding the information paradigm to our engineering toolkit can best be estimated by a view at nature. The biological world exhibits a hierarchy of self-processes that lead to increasingly more complex organisations of matter (**B**)

tion of information in addition to the traditional considerations of materials and energy (Figure 3A).

The concept of *informed matter* [32], i.e., molecules deliberately designed to carry information that enables them to interact individually, autonomously with other molecules, provides a basis for heterogeneous three-dimensional fabrication. Combining the abstract concepts of artificial chemistry [33] with the physics of supramolecular chemistry [28, 34] conceivably will enable the orchestration of self-organisation to arrive in practical time scales at physics-driven architectures.

This path to organising matter requires an information paradigm that does take physics into account. Molecules cannot be instructed in the way conventional computers are programmed, because their interactions and behaviour cannot be limited to simple abstractions. A methodology more akin to mechanical engineering than to software engineering, conceptually depicted in figure 4, is required.

4 Perspectives

Potential application domains for the principles outlined in the previous section are architectures in which the amount of matter necessary to implement a required function is important. Examples are pervasive computing and space exploration devices. A likely early application niche is the area of autonomous micro-robotic devices. With the quest for robots at a scale of a cubic millimetre and below molecular controllers become increasingly attractive [35, 36], and initial steps towards implementation are underway [37]. Coherent perception-action under real-time constraints with severely limited computational resources does not allow for the inefficiency of a virtual machine that abstracts physics away. For satisfactory performance the robot's control needs to adapt directly to the reality of its own body [38]. In fact the body structure can be an integral part of the computational infrastructure [39]. A second application domain is

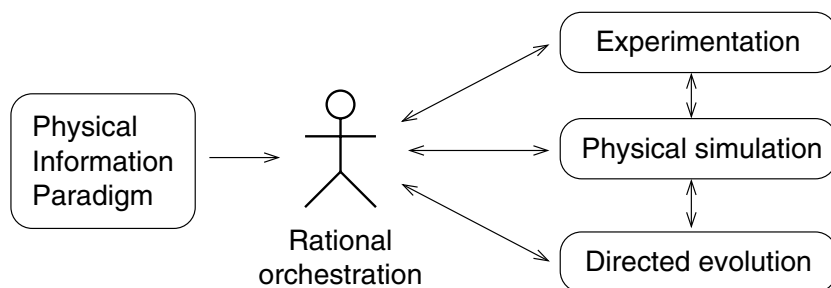


Fig. 4. Orchestration of informed matter. The interplay of experimental data, physical simulation of component behaviour in a systems context and adaptation methods such as directed evolution will play an important role in the process of engineering informed matter building-blocks that self-organise to spontaneously form architectures with desired functionality

bioimmersive computing. Components of a computational architecture could be encoded in the genome of a host cell and upon expression the computational machinery would autonomously form within the cell [40, 41, 42]. Any computational device small enough to fit within a cell will be severely limited in the amount of matter that can be used to implement it. It will need to employ the physics of its material directly for realizing its operations.

The 18 million organic compounds we known today comprise a negligible part of the space of possible organic molecules, estimated to 10^{63} substances [43]. Nature offers a glimpse at what is available in this space of possibilities with organised, adaptive, living, thinking, conscious matter. Following Lehn’s trail-blazing call to “ultimately acquire the ability to create new forms of complex matter” [32] will require information processing concepts tailored to the microphysics of the underlying computational substrate.

Acknowledgements. Comments by Stefan Artmann, Srinandan Dasmahapatra, and Denis Nicole are gratefully acknowledged.

References

1. Turing, A.M.: On computable numbers with an application to the Entscheidungsproblem. In: Proceedings of the London Mathematical Society. Volume 42. (1937) 230–265 Corrections, Ibid vol. 43 (1937), pp. 544–546. Reprinted in *The Undecidable*, M. Davis, ed., Raven Press, New York, 1965.
2. Minsky, M.L.: Computation: Finite and Infinite Machines. Prentice-Hall, Englewood Cliffs, N.J. (1967)
3. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
4. Compañó, R.: Trends in nanoelectronics. *Nanotechnology* **12** (2001) 85–88
5. Chaitin, G.J.: On the length of programs for computing finite binary sequences. *J. Assoc. Comput. Mach.* **13** (1966) 547–569

6. Conrad, M., Zauner, K.P.: Conformation-based computing: a rational and a recipe. In Sienko, T., Adamatzky, A., Rambidi, N., Conrad, M., eds.: *Molecular Computing*. MIT Press, Cambridge, MA (2003) 1–31
7. Zauner, K.P., Conrad, M.: Molecular approach to informal computing. *Soft Computing* **5** (2001) 39–44
8. Landauer, R.: Irreversibility and heat generation in the computing process. *IBM Journal* **5** (1961) 183–191
9. Landauer, R.: Fundamental limitations in the computational process. *Berichte der Bunsen-Gesellschaft* **80** (1976) 1048–1059
10. Langdon, W.B.: How many good programs are there? How long are they? In Rowe, J., et al., eds.: *Foundations of Genetic Algorithms FOGA-7*, Torremolinos, 4–6 September, Morgan Kaufmann (2002)
11. Langdon, W.B.: The distribution of reversible functions is normal. In Riolo, R., ed.: *Genetic Programming Theory and Practice*, Ann Arbor, 15–17 May, Proceedings, Dordrecht, Kluwer Academic Publishers (2003)
12. Pfaffmann, J.O., Zauner, K.P.: Scouting context-sensitive components. In Keymeulen, D., Stoica, A., Lohn, J., Zebulum, R.S., eds.: *The Third NASA/DoD Workshop on Evolvable Hardware—EH-2001*, Long Beach, 12–14 July 2001, IEEE Computer Society, Los Alamitos (2001) 14–20
13. Conrad, M.: Scaling of efficiency in programmable and non-programmable systems. *BioSystems* **35** (1995) 161–166
14. Conrad, M.: The price of programmability. In Herken, R., ed.: *The Universal Turing Machine: A Fifty Year Survey*. Oxford University Press, New York (1988) 285–307
15. Partridge, D.: Non-programmed computation. *Communications of the ACM* **43** (2000) 293–302
16. Schüz, A.: Neuroanatomy in a computational perspective. In Arbib, M.A., ed.: *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA (1995) 622–626
17. Cariani, P.: To evolve an ear: epistemological implications of Gordon Pask’s electrochemical devices. *Systems Research* **10** (1993) 19–33
18. Thompson, A., Layzell, P., Zebulum, R.S.: Explorations in design space: Unconventional electronics design through artificial evolution. *IEEE Trans. Evol. Comp.* **3** (1999) 167–196
19. Miller, J.F., Downing, K.: Evolution in materio: Looking beyond the silicon box. In: 2002 NASA/DoD Conference on Evolvable Hardware (EH’02), July 15 - 18, 2002, Alexandria, Virginia, IEEE (2002) 167–176
20. Conrad, M.: Information processing in molecular systems. *Currents in Modern Biology (now BioSystems)* **5** (1972) 1–14
21. Zauner, K.P., Conrad, M.: Parallel computing with DNA: toward the anti-universal machine. In Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P., eds.: *Parallel Problem Solving from Nature: PPSN IV*. Volume 1141 of *Lecture Notes in Computer Science*, Berlin, Springer-Verlag, Berlin (1996) 696–705
22. Conrad, M.: The seed germination model of enzyme catalysis. *BioSystems* **27** (1992) 223–233
23. Adler, J., Tso, W.W.: “Decision”-making in bacteria: Chemotactic response of *Escherichia coli* to conflicting stimuli. *Science* **184** (1974) 1292–1294
24. Scharf, B.E., Fahrner, K.A., Turner, L., Berg, H.C.: Control of direction of flagellar rotation in bacterial chemotaxis. *Proc. Natl. Acad. Sci. USA* **95** (1998) 201–206
25. Wikoff, W.R., Johnson, J.E.: Virus assembly: Imaging a molecular machine. *Current Biology* **9** (1999) R296–R300

26. Ban, N., Nissen, P., Hansen, J., Moore, P.B., Steitz, T.A.: The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science* **289** (2000) 905–920
27. Müller, A., Beckmann, E., Bogge, H., Schmidtmann, M., Dress, A.: Inorganic chemistry goes protein size: a mo368 nano-hedgehog initiating nanochemistry by symmetry breaking. *Angew. Chem. Int. Ed. Engl.* **41** (2002) 1162–1167
28. Lehn, J.M.: Supramolecular chemistry—scope and perspectives: Molecules, supermolecules and molecular devices. *Angewandte Chemie, Int. Ed. Engl.* **27** (1988) 90–112
29. Seeman, N.C.: DNA in a material world. *Nature* (2003) 427–431
30. Zhang, S.: Fabrication of novel biomaterials through molecular self-assembly. *Nature Biotechnology* **21** (2003) 1171–1178
31. Conrad, M.: Quantum mechanics and cellular information processing: The self-assembly paradigm. *Biomedica Biochimica Acta* **49** (1990) 743–755
32. Lehn, J.M.: Supramolecular chemistry: from molecular information towards self-organization and complex matter. *Reports on Progress in Physics* **67** (2004) 249–265
33. Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries—a review. *Artificial Life* **7** (2001) 225–275
34. Whiteside, G.M., Mathias, J.P., Seto, C.T.: Molecular self-assembly and nanochemistry: A chemical strategy for the synthesis of nanostructures. *Science* **254** (1991) 1312–1319
35. Ziegler, J., Dittrich, P., Banzhaf, W.: Towards a metabolic robot control system. In Holcombe, M., Paton, R., eds.: *Information Processing in Cells and Tissues*. Plenum Press, New York (1998) 305–317
36. Adamatzky, A., Melhuish, C.: Parallel controllers for decentralized robots: towards nano design. *Kybernetes* **29** (2000) 733–745
37. Adamatzky, A., de Lacy Costello, B., Melhuish, C., Ratcliffe, N.: Experimental implementation of mobile robot taxis with onboard Belousov-Zhabotinsky chemical medium. *Materials Science & Engineering C* **24** (2004) 541–548
38. Elliott, T., Shadbolt, N.R.: Developmental robotics: manifesto and application. *Phil. Trans. R. Soc. Lond. A* **361** (2003) 2187–2206
39. Hasslacher, B., Tilden, M.W.: Living machines. *Robotics and Autonomous Systems* (1995) 143–169
40. Atkinson, M.R., Savageau, M.A., Myers, J.T., Ninfa, A.J.: Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell* **113** (2003) 597–607
41. Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., Shapiro, E.: An autonomous molecular computer for logical control of gene expression. *Nature* **429** (2004) 423–429
42. Blake, W.J., Isaacs, F.J.: Synthetic biology evolves. *Trends in Biotechnology* **22** (2004) 321–324
43. Scheidtmann, J., Weiß, P.A., Maier, W.F.: Hunting for better catalysts and materials—combinatorial chemistry and high throughput technology. *Applied Catalysis A: General* **222** (2001) 79–89