A Research Framework for Interaction Computing

Paolo Dini[1] and Daniel Schreckling[2]

[1] Department of Media and Communications
London School of Economics and Political Science
London, United Kingdom
p.dini@lse.ac.uk
[2] Institute of IT-Security and Security Law
University of Passau, Passau, Germany
ds@sec.uni-passau.de

Abstract. This paper lays out an interdisciplinary research framework that integrates perspectives from physics, biology, mathematics, and computer science to develop a vision of interaction computing. The paper recounts the main insights and lessons learned in the past six years across multiple projects, gives a current definition of the problem, and outlines a research programme for how to approach it that will guide our research over the coming years. The flavour of the research is strongly algebraic, and the bridge to specification of behaviour of automata through new formal languages is discussed in terms of category theory. The style of presentation is intuitive and conceptual as the paper is meant to provide a foundation widely accessible to an interdisciplinary audience for five threads of research in experimental cell biology, algebraic automata theory, dynamical systems theory, autopoietic architectures, and specification languages, the first four of which are represented by more focussed technical papers at this same conference.

Keywords: Bio-Computing, Interaction Computing.

1 Introduction

This research is motivated by the fundamental question whether a biological ecosystem, or a subset thereof, could be used as a model from which to derive self-organising, self-healing, and self-protection properties of software. This research question is premised on the assumption that such biological properties can increase the effectiveness of information and communication technologies (ICTs) in various application domains, from ubiquitous computing, to autonomic communications, to socio-economic processes aimed at regional development, simply on the basis of their greater and spontaneous adaptability to user needs. Thus, this research addresses some of the non-functional requirements or software qualities of the underlying technology, which we refer to as software ecosystems [17].

This paper presents a research framework that aims to achieve a usable model of bio-computing, based on several years of research across several projects [16] [17] [18]. The application areas of interest ultimately are:

F.A. Basile Colugnati et al. (Eds.): OPAALS 2010, LNICST 67, pp. 224–244, 2010.

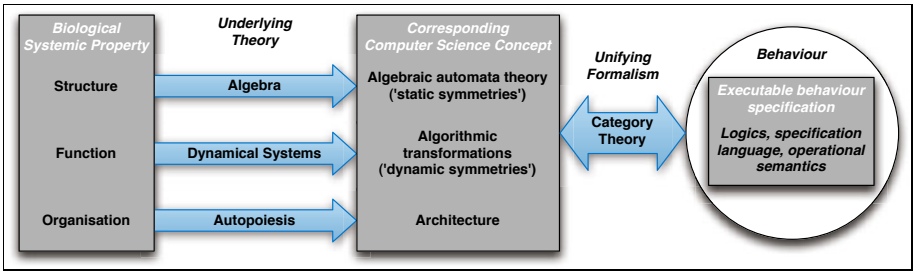

Fig. 1. High-level view of the theoretical research framework

– Service composition in the context of dynamic business workflow instantiation
– Biologically-inspired RESTful interaction framework
– Symbiotic security

Figure 1 gives a high-level view of the theoretical research framework that will be discussed and justified in more detail in the rest of this paper. The most important aspect of the theory that is emerging is that it needs to address three fundamental aspects of biology: structure, function, and organisation. Our preliminary results and insights point to algebra, dynamical systems, and autopoiesis, respectively, as the theories that can explain and/or model these aspects of biology and that need to be unified by a common mathematical framework that can effect a mapping to computer science. The target of these mappings appears to be a unification of the algebraic and algorithmic structure of automata, and novel ideas in software architectures and biological design patterns inspired by autopoiesis. Category theory is then able to relate any of the structures thus defined that have algebraic character to automata behaviour (which is also some kind of algebra) and from behaviour into a language which may be used to express (specify) some particular structures. Part of this language may be some kind of logic. Instantiation of this framework in modern distributed and web-oriented computing environments may be expressible compatibly with the Representational State Transfer (REST) architectural style [29]. It is important to emphasise that the term "structure" is quite overloaded in our work. It can refer to biological (physical) structure or to algebraic structure. Hopefully the different meanings will be clear from the context.

This paper outlines a research framework that is explored in greater depth in the following four companion papers at this same conference:

– A Research Framework for Interaction Computing (this paper)
– Numerical and Experimental Analysis of the p53-mdm2 Regulatory Pathway [62]
– Lie Group Analysis of a p53-mdm2 ODE Model [35]
– Transformation Semigroups as Constructive Dynamical Spaces [23]
– Towards Autopoietic Computing [9]

We now retrace the arguments and rationale that we have developed over the past six years in this area of research.

2 Historical Recap

The complexity and interconnections of the research activities that are gradually unfolding in the two projects make it necessary to provide a summary of past activities and to retrace the arguments that have led to the present research rationale. Hopefully this context will make it easier to understand and assess the relevance and validity of the current activities and of the activities that are planned for the remainder of the OPAALS project, and beyond. Accordingly, Figure 2 provides a graphical overview of our research in bio-computing over the past several years. The figure shows the main points that each report addresses (in some cases this is the title of the deliverable) along with the corresponding deliverable number, where by "main" we mean the topics that, in hindsight, were found to be most relevant in later deliverables, as a plausible theoretical and mathematical framework began to emerge.

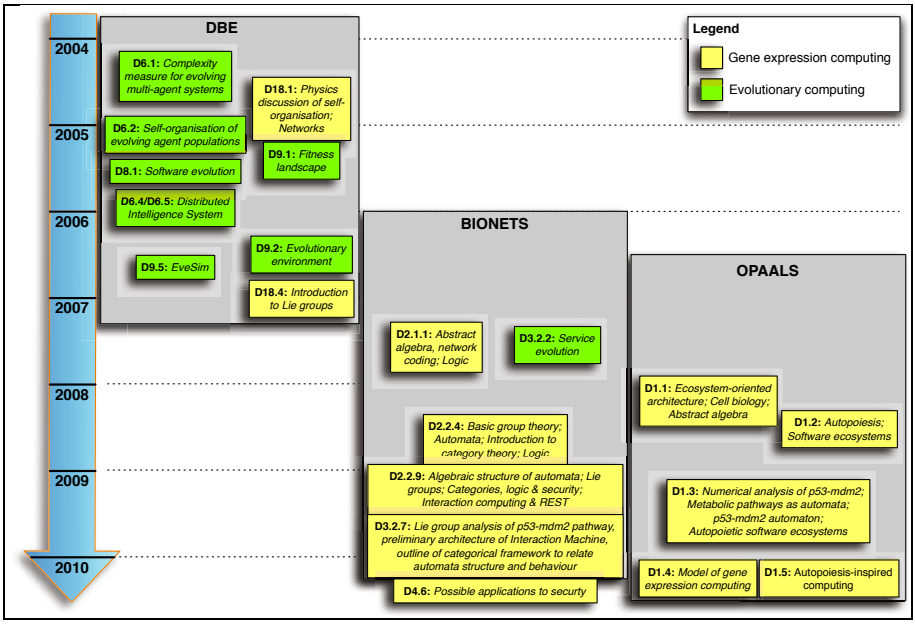

Fig. 2. History of relevant bio-computing reports across several projects

During the preparation of the DBE project, we proposed that the concept of ecosystem could be used not only as *metaphor*, but also as *model* for biologically-inspired computing. Ecosystems are characterised by self-organising and evolutionary processes. Whereas, strictly speaking, evolution is a form of self-organisation, by the latter term we refer to the order construction processes associated with cell metabolism and morphogenesis. In developing our theory of bio-computing, thus, we prioritise ontogeny over phylogeny.

2.1 Evolution and Self-organisation

The current research thread in gene expression or interaction computing began with a discussion of self-organisation through the minimisation of free energy, in DBE D18.1 [16]. Although the concept of free energy is very useful for understanding and modelling self-organisation in physical systems, unlike physical systems software systems are abstract. Thus, the successes of statistical physics are not readily transferrable to software due to the absence of an interaction potential energy and of the concept of temperature in the latter. Of course, the wealth of probabilistic methods based on uniform and nonuniform probability distributions do a good job at achieving an analogous effect; but such effect is contrived in the sense that it is imposed on the digital information which, if left to its own devices, would forever lie still in the 'current state'. However, the users provide a continuous input of information, which we can regard as analogous to the Sun's energy as the fundamental driver of the biosphere. Thus, even if we do not have a proper 'temperature', we do have a continuous flow of information through the system and a continuous poking and prodding by the users that can be seen as analogous to a certain level of thermodynamic 'mixing'. If we abstract a complex distributed computation and communication system as a set of coupled finite-state machines, user inputs become 'waves' of signals that propagate through the system, carried by the interactions between the state machines. The puzzle of self-organisation, thus, could be cast as the problem of deriving appropriate constraints in the execution paths of the state machines that can lead to the construction of ordered structure and behaviour by harnessing the 'energy' (information) flowing through the (open) system.

Clearly the problem posed in this manner is not trivial. In the DBE project we therefore developed an Evolutionary Environment ("EvE") in parallel with more mathematical research [32, 39, 55, 5, 6]. Although we were able to achieve some level of optimisation of the distribution of services in the ecosystem through a neural networks-based Distributed Intelligence System [7, 8], the evolution of the services to satisfy a particular user request was not achieved. It appeared that using services as the atomic units of evolution was not sufficiently granular to respond adequately to different contexts. On the other hand, breaking services down to apply genetic algorithms to the code itself is still too difficult for engineering applications.

The problem seemed to be a lack of understanding of the structural and dynamical features of ecosystems that need to be satisfied in order to support an effective evolutionary framework. Put simply, because evolution is a weak and slow process that, in order to avoid instabilities (death of the phenotype), can only make extremely small modifications to a given genotype, the ecosystem itself must already be highly performant, in the sense that its 'components' must already be quite compatible with one another and must already be close to satisfying a given fitness requirement. This implies the need for a holistic approach, whereby the ecosystem is in some sense 'bootstrapped' all at once through a massively parallel process in which hundreds if not thousands of requirements are satisfied simultaneously and compatibly with one another.

Our objective, therefore, is to find a balance between evolutionary computing and what we are calling gene expression computing. We seek an integration of the two approaches that is analogous to what DNA has been able to achieve: the same molecule is a carrier of hereditary traits across generations whilst also guiding the

morphogenesis and metabolism of the individual organism. Based on our experience in these projects, we feel that the problem of gene expression computing must be solved first, before we can hope to achieve effective evolutionary behaviour. Figure 3 shows how the abstract concept of Interaction Computing can be instantiated into different contexts.[1] Gene expression computing refers to the nuts and bolts of cellular pathways and how they are able to construct order and exhibit stable and robust behaviour; so it is a model oriented towards a *local* perspective. Autopoiesis-Inspired Computing, on the other hand, looks at *global* properties of the cell and of autopoietic systems, and tries to map these properties to computer architectures that replicate autopoietic behaviour or its subsets (such as operational closure). Autopoiesis-Inspired Computing is discussed in another paper in this same conference [9]. Finally, Symbiotic Computing is more specifically focussed on the ecosystemic properties of interdependence and synergy, and it is being pursued in the BIONETS project in particular as regards software security.

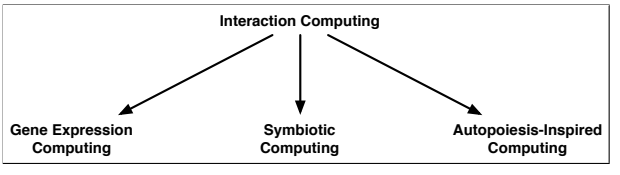

Fig. 3. Different possible models of computation derived from Interaction Computing

This prioritisation of ontogeny over phylogeny implied that an in-depth investtigation of the physics and mathematics of (non memory-based) self-organisation was necessary in order to understand what features could be transferred to software. Because, in addition to the minimisation of free energy, both cell biology and ecosystems are characterised by non-linear processes, we realised that we faced a 'double jeopardy': not only does it seem challenging to translate non-linear behaviour into automata or algorithmic constraints, as above, but the non-linear behaviour itself is in most cases the signature of systems that are not even integrable. In spite of the daunting stack of challenges that was taking form, we kept focusing on the fact that biological systems at all scales *are* able to cope with these challenges: they do an extremely good job at producing ordered structures and behaviour, in spite of their complexity and of the non-integrability of most mathematical models of biological phenomena (which could be related to their non-computable aspects). This was encouraging (if a biological system can manage this, there must be a way to formalise it), even if it suggested to us that new ways to think of complex physical and biological phenomena were likely to be needed.

2.2 Symmetry

Based on our previous experience in applied mathematics and physics of the usefulness of the concept of symmetry, our starting point was to assume that the same

[1] No references are given for these terms because we invented them – and are in the process of developing formalisations for them.

concept was likely to play an important role also here. Our intermediate results so far have confirmed this hunch. Symmetry is a very general concept in mathematics that formalises the notion of invariance or regularity. In mathematics, a symmetry is a *transformation* that leaves some property of a mathematical object invariant. Now, it is a truth universally acknowledged (and easily proven) that the invertible transformations of a mathematical object that leave some property of its structure invariant form a group.[2] Therefore, the mathematical study of symmetries and regularities must necessarily rely on algebra.

The above statement should be taken as a *necessary* rather than as a *sufficient* condition. In other words, a technical system that interfaces at some level with human users and that is meant to support socio-economic processes must be open to new information and must allow for the emergence of new structures and patterns. Even if such a requirement were not enforced or relevant (i.e. if all we were trying to do was to develop an artificial life environment), the wish eventually to replicate and support evolutionary behaviour implies that the emergence of new forms must be supported. Our current understanding of algebra is not necessarily sufficient to develop the best mathematical framework for the formalisation of emergent behaviour and open-ended evolution. By the same token, however, the system must also be stable and reliable, since it is meant also to uphold robust (self-healing!) engineering applications and non- functional requirements. It must behave similarly in similar contexts; hence, it must embody a fair amount of regularity and predictable behaviour. This is what mathematics, and algebra in particular, formalises. Again, we wish to emulate the delicate balance between order/reliability and unpredictability/openness that biology has been able to fine-tune and leverage to produce stable but ever- changing life-forms of unbelievable complexity.

2.3 Lie Groups

In DBE D18.4 [15] we therefore began a discussion of the method of Lie groups for the solution of differential equations, since it is the most general method that applies equally well to linear and non-linear systems. At that time we were aware that a method developed for continuous systems would be difficult to apply to discrete automata, but we were also aware of the fact that generalisations of Lie groups have been applied to discrete dynamical systems.[3]

The relevance of an algebraic perspective was strengthened by observing how finite ring and field theory has been used in network coding. An examination of network coding was motivated initially by the BIONETS project, where we thought that the ability to reconstruct missing information from a bitstream might have been extended towards self-healing properties of software, or perhaps the reconstruction of the whole phenotype from a partial specification. However, it soon became apparent that the value of the exercise was more as an example of abstract algebra that was relatively accessible to computer scientists than as a technique that could be directly relevant to evolutionary or gene expression computing. Because this algebraic theory deals with discrete finite sets, it not only demonstrated another area of applications

[2] Paraphrased from Stewart ([61]:xxvii).
[3] See Maeda [42] and Peter Hydon's work at
`http://personal.maths.surrey.ac.uk/st/P.Hydon/sym.htm`

where algebra is relevant but, by providing a basis for the more difficult group theory, it also brought us one step closer to the mathematical formalisation of symmetries in the context of computer science. This abstract groundwork was reported in both projects [20, 17].

At about the same time we ran across the work of the Cuban HIV researchers Sanchez, Morgado, and Grau [57, 58, 56], an interdisciplinary research team composed of a biochemist, a mathematician, and a computer scientist. AIDS research is concerned with, among other things, mutations in the DNA of the HIV virus. Mutations that impede the ability of this virus to function are good news for us. The operational effectiveness of a particular strand of DNA is dependent on the geometry of the proteins (enzymes) that are synthesised from it through gene expression, because this geometry has to match the complementary geometry of its substrate for the enzyme to be effective. The 3-D shape of an enzyme depends on the folding of the strand of aminoacids built by the ribosomes from the corresponding tract of DNA, by applying the genetic code.[4] Protein folding depends to a large extent on polar bonds which, in turn, depend on the hydrophobicity of the aminoacids along the chain. The hydrophobicity of an aminoacid depends on the second base of the corresponding 3-base codon. We know empirically that mutations are most likely to occur in the middle or second base of a codon. Now the surprising fact is that, if a codon undergoes a mutation (most likely to happen in its second base) to a new codon, the hydrophobicity of the new aminoacid will be very similar to the original aminoacid's. Furthermore, it turns out that if the 20 aminoacids are arranged in order of increasing hydrophobicity the corresponding codons form a partial order, in fact a 64-node Boolean lattice.

Thus, a particular assignment of the bases to the field extension GF (2^2) (represented by the 4 nucleotite bases) leads to a Boolean lattice (as a third direct product of the 2×2 base lattice due to the fact that each codon is formed by three bases) whose minimum and maximum elements are the codons that correspond to the least and most hydrophobic aminoacids, and this assignment leads to a self-consistent partial order for the rest of the codons that matches corresponding levels of hydrophobicity. The relevance of this finding is that this particular algebraic structure corresponds to what amounts to hydrophobicity as a *continuous function* of codon mutation. In other words, the operational semantics of the DNA code are fairly robust with respect to mutations. This is not good news for AIDS research, because it confirms the observation that mutations of the HIV virus are likely to remain as deadly as the originals. However, the same effect underpins the stability of any other organism with respect to pertur- bations brought by genetic mutations, i.e. it takes a relatively improbable large mutation to upset the functioning of a particular phenotype. In other words, the robustness of the most fundamental 'architectural' feature of biology, the DNA code, is formalisable through an equally fundamental algebraic structure. Boolean algebras are not uncommon, however. So the fact that a particular data set forms a partial order or even a Boolean lattice (slightly more restrictive) is not necessarily of great significance.

[4] The genetic code is a many-to-1 map from the 64 codons to the 20 aminoacids. Each codon is composed of 3 bases, each of which can assume one of the 4 values A, G, T, E. Hence, 4 bases occupying 3 possible slots: $4^3 = 64$.

In their more recent work Sanchez, Morgado, and Grau [56] report that the codons actually carry additional structure, in particular they form a Lie algebra. A Lie algebra is a vector space whose elements satisfy an additional binary operation, the Lie bracket. Because the set of codons can be seen not only as a Boolean algebra but also as the Galois field extension $GF(2^6)$, it already was isomorphic to a (discrete and 3-dimensional) vector space over the finite field $GF(2^2)$, so this means that the codons also satisfy the Lie bracket, as an additional constraint. The physical significance of this fact is not clear; however, we know that a Lie algebra can also be seen as the tangent space to a Lie group at its identity, and a Lie group is the only algebraic structure that can sometimes help us in solving non-linear dynamical systems – for example the non-linear dynamical systems that formalise cell metabolic and regulatory pathways. Therefore, once again not only does the algebraic approach seem justified, but the need to develop a unified theory between (discrete) finite group theory and (continuous) Lie group theory around dynamical systems arising from cellular processes appears increasingly likely.

The investigation of DNA as a Lie algebra will be performed in future projects because first we need to assess the feasibility of the Lie group perspective in the solution of cell metabolic and regulatory pathways. Thus, our shorter-term objective is to extend the work begun in DBE D18.4 and perform a Lie group analysis of the p53-mdm2 regulatory pathway (see [35] in this same conference, which is based on [18] and [40]).

2.4 Functional Completeness

There is one more topic that provides an important background to our research: functional completeness [34]. The interesting aspect of this point of view is that it resonates with the physics and engineering research literature around a concept that seems at first unrelated to our discussion: choice of variables.

It is well-known in the modelling of physical phenomena that a judicious choice of coordinate system and/or of the representation of the dependent and independent variables can simplify the mathematics greatly, at the same time providing useful insights into the nature of the problem under study.[5] The choice of coordinate system is perhaps easier to see, for example when choosing cylindrical coordinates to describe fluid flow through a circular pipe. Many physical problems, however, can also be characterised by groupings of variables that also simplify the mathematics considerably. This was first noticed in the 19th Century by experimental researchers in a variety of applied and scientific disciplines, who noticed that particular dimensionless groupings of variables could sometimes lead to the collapse of data clouds and families of data sets onto single curves. The practical usefulness of this fact was soon to be investigated more rigorously, leading eventually to Lie's group-theoretical methods for differential equations.

The general epistemological principle we can derive from this is that in many complex problems increasing complexity of the variables used to describe them often appears to simplify the mathematical model, in some cases leading to an analytical solution. This same principle could be relevant to the problem of bio-computing

[5] E.g. see the famous Buckingham Pi Theorem [10] and generalisations thereof.

when, as Horvath has done, we generalise the fundamental structures of computer science to more complex structures.

In particular, digital computers today are able to perform any computation because they are functionally complete. This means that there is an algebraic structure, in this case a Boolean algebra, such that any *n*-ary function can be represented by a corresponding propositional logic expression (or 'polynomial') that is implementable as logic gates. It has been known for many years that one can use more general algebraic structures to achieve equally functionally complete computational models. Horvath investigated whether a semigroup can have the functionally complete property expressible as more general 'polynomials' than propositional logic. He proved that the answer is Yes, as long as the semigroup is a finite simple non-abelian group (SNAGs).[6] In group theory, SNAGs play a role similar to prime numbers in number theory, thus the possible ramifications of this fact are quite intriguing. Because, even though they are somewhat special, there are infinitely many such groups, this means that we could build a 'more complex' computer science using more complicated fundamental structures.

What does this ultimately mean and what would this buy us? In terms of Turing computability, these different ways of thinking of computing would not change anything. We would compute problems of the same complexity class. However, we argue that it is worth investigating what kind of computations we might be able to perform, and how, but using SNAGs rather than Boolean algebra as the fundamental starting point for computing. Another analogy that may help clarify this point is to compare the use of Assembly language versus objects. One can program anything in Assembler, and in fact any program is eventually compiled down to binary code, but it's a lot easier to program classes and let the compiler do the hard work.

With this historical background in mind we now turn to the problems we are currently facing in our research.

3 Current Research Questions

3.1 Abstraction Level

Cell metabolism relies on ultimately undirected bottom-up and random/stochastic processes that can only 'execute' through the spontaneous interaction of the various components. The interactions are driven by a combination of electrostatic forces (usually conceptualised as minimising the potential energy of interaction) and most probable outcomes (maximisation of entropy), which can be modelled together as the minimisation of free energy. In spite of this fundamental randomness, however, a healthy cell behaves in an organised and finely balanced way that is more evocative of a deterministic, even if very complex, machine than of random chaos. The cell in fact has a definite physical structure and executes well-defined 'algorithms' in the form of cellular processes (several hundred per cell type) such as metabolic or regulatory biochemical pathways. This suggests a description and modelling of cell behaviour at a level of abstraction that is higher than the molecular, and through mechanisms or constraints that are complementary to stochastic processes.

[6] Every group is also a semigroup, but not conversely of course.

In particular, our perspective views the stochastic nature of cell biochemistry mainly as a mechanism of dimensional reduction[7] that does not necessarily need to be emulated in any detail. For example, a gene expresses hundreds of mRNA molecules which, in turn, engage hundreds of ribosomes for no other reason than to maximise the probability that a particular, *single* genetic instruction will be carried out, such as the synthesis of a particular enzyme. As a consequence of this dimensional reduction (hundreds to 1), a higher level of abstraction than that at which stochastic molecular processes operate is justified in the modelling approach – in particular, a formalisation that retains, and builds on, the discrete properties of cell biology.[8] However, even the resulting lower-dimensional system can't plausibly be imagined to perform the complexity of a cell's functions driven simply by a uniform distribution of interaction probability between its (now fewer) components. Additional structure and constraints must be at play.

3.2 Dynamic Stability

The presence of additional constraints is evident from the internal physical structure or topology of the cell. For example, the citric acid cycle that metabolises energy from sugar takes place within the mitochondrion, isolated from the rest of the cell. But cellular macrostructures such as the mitochondrial membrane are too coarse to explain the bewildering complexity of parallel processing that takes place even within the mitochondrion itself. There must be constraints operating at a finer granularity that support specific reaction pathways over others and that prevent the cytoplasm from becoming a well-mixed solution of compounds of uniform concentration reacting indiscriminately with one another. In other words, even if the precise form of these additional constraints that keep cellular processes running smoothly is far from evident, their existence is implicit in the complex and *dynamically stable* operation of the metabolic and regulatory pathways.

Dynamic stability is only an intuitive concept at this point, which can be thought of as the signature of certain types of non-linear behaviour and for which a precise mathematical definition does not exist yet, although research in related fields is growing ([66, 43]). However, we can say that dynamic stability is a generalisation of the well-trod engineering principle of stable design, which tends to keep human machinery within its linear regime in fear of catastrophic failure if instabilities or resonances are allowed to grow. But linear systems are information-poor and cannot sustain rich and complex behaviour. Biology has been able to harness the expressive power of non-linear behaviour whilst maintaining adequate stability, thereby capturing the 'sweet spot' between order and chaos. From the point of view of information theory, linear systems tend to have a discrete power spectrum, whereas

[7] In dynamical systems theory, dimensional reduction refers to a reduction in the number of degrees of freedom of a system. Since biochemical systems are composed of thousands to millions of elements, the time evolution of each of which is governed (for the sake of argument within a Newtonian framework) by at least three separate ordinary differential equations (ODEs), successful abstraction and dimensional reduction can lead to significant theoretical insight and savings in CPU requirements.

[8] Notice that the statistical nature of the metabolic step carries a built-in robustness, i.e. if something is wrong with one of the proteins being generated, the metabolic cycle as a whole can proceed unhindered.

chaotic systems have a flat or continuous 'white noise' spectrum. An example of a human creation that strikes a balance between these two extremes and that is at a similar level of abstraction as software is music, which was discovered to be uniformly 1/f-noise, 30 years ago [63]. This provides motivation for why we think that mapping the greater expressive power of non-linear behaviour into computer science concepts will lead to a correspondingly greater power to 'compute' unprogrammed behaviour in real time.

The fact that the cell is not a well-mixed solution tells us, as is well-known, that it must not be in thermodynamic equilibrium. Prigogine's work [48] is deeply significant because it showed that ordered structures form in open systems under conditions of disequilibrium – maintained as such by a constant energy flow. Thus, although the phenomena he studied (e.g. the toroidal vortices of Rayleigh- Benard convection) are much simpler than what happens inside a cell, his insights give us a relatively concrete example of what a 'dynamical structure' might look like. The dynamic stability of cellular processes then constitutes a generalisation of Prigogine's ordered structures. Therefore, treating cellular processes as automata, or discrete low-dimensional dynamical systems, appears to be the most appropriate level of abstraction and entry point to understand biological construction of order in a way that is relatively easy to transfer to computer science.

3.3 Structure and Function in Biology and Computer Science

To make progress in this direction, we take as a starting hypothesis that the dynamically stable operation of the cell is critically dependent on two additional forms of structure that are more abstract than physical structure and that can be formalised mathematically as follows (see Figure 4):

- Time-independent algebraic structure of the automata modelling the cellular pathways. Algebraic structure gives rise to what we are calling static symmetries.
- Time-dependent Lie group structure of the dynamical systems modelling the same cellular pathways. This form of structure is formalised through a mix- ture of algebra and geometry and gives rise to what we are calling dynamic symmetries.

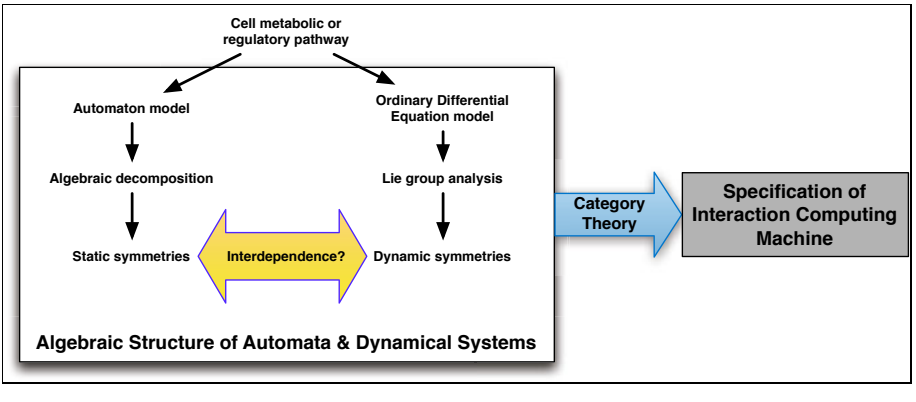

Fig. 4. Mathematical analysis workow to uncover biological symmetries

The relevance of the relationship between structure and function to all types of engineering and applied thinking motivates us to investigate how these two kinds of mathematical structure are related. The benefit of such a relationship would be the ability to specify desired behavioural properties and derive the corresponding structural properties.

In its simplest form, a finite-state automaton is a nite set of states acted upon by a semigroup of transformations. Until the 1960s the general consensus was that semigroups were too unstructured for anything useful to be done with them. This perception was changed by one of the landmark theorems in this field, the Krohn-Rhodes prime decomposition theorem for finite semigroups [38], which proved the existence of a much greater amount of structure in semigroups. The relevance of semigroups to automata has then made this mathematical theory of increasing interest to computer science over the past 40 years. Furthermore, the non-linear character of automata ([36]) suggests that they are the right instrument to model the enormously intricate feedback loops of discrete cellular processes. This observation is greatly strengthened by the current research of the Biocomputation Laboratory at the University of Hertfordshire, UK ([47, 26, 25, 24, 27]), in which several examples of cell regulatory and metabolic pathways are shown to be formalisable as finite-state automata. The application of Krohn- Rhodes decomposition to the corresponding semigroups then reveals the presence of a rich algebraic structure in the form of permutation groups and non-invertible components (flip-flops) at different levels of their hierarchical decomposition.

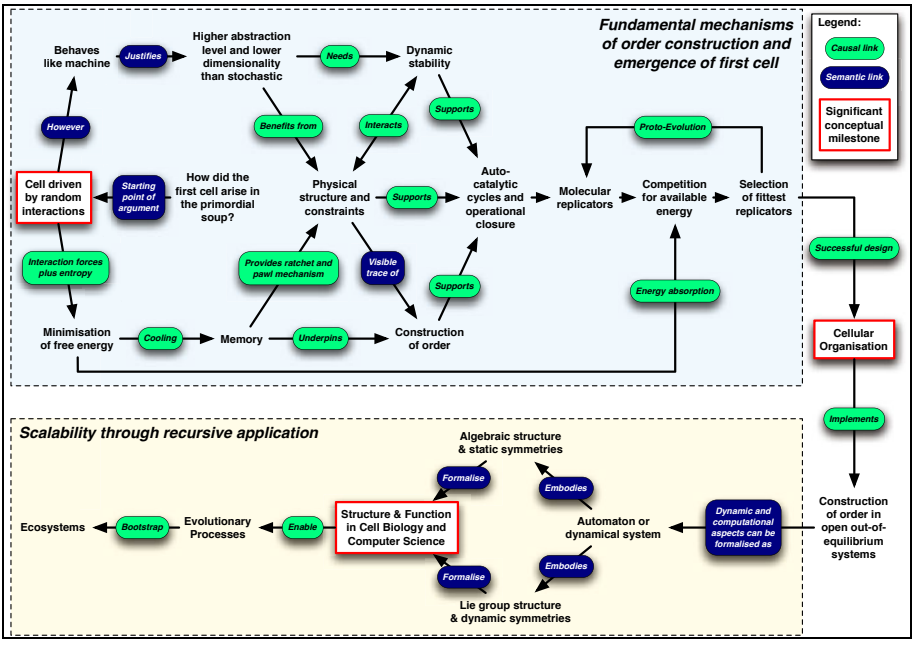

Fig. 5. Causal-semantic workflow summarising a part of the research rationale

The algebraic structure of automata does not account for their time-dependent or dynamic behaviour. Therefore, a significant challenge we face is how to make sense of the often non-integrable dynamical behaviour of non-linear systems. Systems biology, in fact, relies heavily on the numerical solution of the ordinary differential equations (ODEs) derived from the chemical rate equations modelling the cellular pathways, simply because no analytical solutions exist. However, as we mentioned above it is well-known that in many cases systems of coupled non-linear ODEs embody so-called global symmetries obtainable through Lie groups analysis [49]. Although global symmetries are quite constraining and are correspondingly difficult to find, this is not necessarily a drawback since biological systems exhibit ordered behaviour only within certain ranges of their parameters (e.g. temperature). In other words, Lie groups can help us solve mathematical models that are clearly very idealised approximations to how real systems work. However, the important point is that they do capture and formalise the concept of order in dynamical behaviour, which we have loosely called 'dynamic stability' above. It is not unreasonable to claim, therefore, that the symmetries corresponding to 'local' or parameter-limited ordered biological behaviour could be found through an extension of Lie's theory to less rigidly defined mathematical structures such as groupoids, as well as to discrete dynamical systems due to their closer relevance to automata:

> There are plenty of objects which exhibit what we clearly recognize as symmetry, but which admit few or no nontrivial automorphisms. It turns out that the symmetry, and hence much of the structure, of such objects can be characterized algebraically, if we use groupoids and not just groups. ([64]; quoted in [31])

Groupoids are like groups except that the group operation (usually functional composition) is defined only for some and not all of the elements.

Figure 5 gives an overall summary of the rationale of the research workflow and of some of the concepts we have discussed so far. Having summarised the main concepts of the mathematical theory, we now start building a bridge towards computer science.

3.4 Behaviour-Based Specification

It appears obvious that several parts of interaction computing systems could be described by existing formal specification frameworks or formal system, such as VDM [3], Z notation [60], CCS [45], π-calculus [46], CSP [33], LOTOS [4], ACP_τ [2], etc. While there are languages which are very similar to our approach, and Aspect-Oriented Programming (AOP) is certainly one of them, the reason for developing a new language is fundamentally different. Interaction computing is highly different from existing systems in terms of its concurrency, its interdependability, its realisation of functionality, its non-deterministic and probabilistic computation, and its modularity. Modifications of some specification languages may support all these properties. This has been shown in the past, for example, for Z. Step after the step the original language was extended with new features, such as non-determinism or the full support of temporal logic. This valuable engineering process extends a language such that it fits a certain need. However, this requires that the actual problem the language describes is similar.

Our problem is interaction computing and instead of trying to describe interaction computing using an existing language, adapting it to our needs, we take the opposite approach and start with analysing the problem first, i.e. its dynamical and structural properties. In the course of our research we will learn about this structure and identify basic functional components inspired by biology. This will also determine the primitives of the language. On top of that, our language will be based on behaviour the system to be described should exhibit. Here, the internal structure of the components realising this behaviour is not essential. They are hidden from the specification as they are far too complex. This is in strong contrast with existing formal specification methods which try to describe the actual functionality but not the behaviour. Here we define functionality as the actual functions which have to be executed to implement a certain behaviour.

Thus, the functionality of an interaction machine describes in detail the internal states and transitions the machine has to go through in order to achieve its desired behaviour, i.e. the specification would follow a white box characteristic approach. In contrast, the behaviour describes the observable or expected effects of a black box. Thus, behaviour strongly abstracts from the internal structure and gives a wider flexibility to its implementation. This takes the established high-level programming and specification languages one step further. While they already abstract from the hardware level and use higher-order programming language constructs, the biologically-inspired interaction computing specification language even abstracts from functionality and lifts programming and specification to the behavioural level. In our work we are studying how the two concepts of machine structure and its behaviour are strongly linked in categorical terms [40]. In particular, we show how a category of behaviour is directly linked to a category of machines realising this behaviour.

Additionally, to be able to transform an existing specification into an executable form, the specification language requires some operational semantics which allows us to translate a behaviour specification into interaction machines and their execution steps. Similar to functional or logical specification languages, the realisation of such an approach in an executable instance includes several implicit steps wihch are not explicitly stated in a machine specification. In interaction computing this process is even more complex because even simple operations are realised by multiple interactions between multiple machines. Adapting the operational semantics of an existing language becomes infeasible. Thus, we follow the general design process which tries to develop a language which actually fits best our needs.

Finally, we do not refuse the use of existing formal systems. In fact, our work already uses mechanisms [1] which allow us to transform one logic into a comparable one, to recognise the well-established correspondence between coalgebras and temporal logics (see also BIONETS deliverable D2.2.4 [22]), or which compare their internal structures. If we find that our systems possess properties which are describable by existing formal systems, we will opt for them, of course.

Thus, this work aims to develop the basis of an ‘environment specification’ language, which can be seen as a higher-abstraction software engineering specification language addressing both the structure and content of bio-inspired digital systems. Figure 6 shows at a high level how category theory can enable a mapping from algebraic and coalgebraic structures to algebraic and coalgebraic logic, as an

initial step in this direction. This work is in progress ([20, 22, 21, 19, 59, 18, 40]) and elaborates concepts which map algebraic structure corresponding to automata into categories of behaviour.

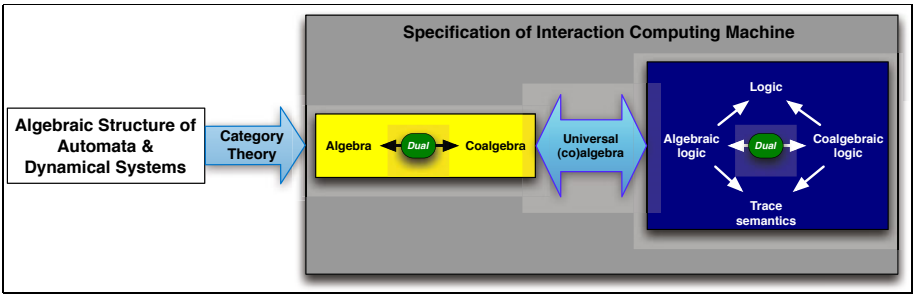

Fig. 6. Mapping of algebraic structures to logic structures through category theory

3.5 Organisation in Biology and Computer Science

The reliance on category theory is further motivated by Rosen [51, 52] who, following Rashevsky's ideas [50], first applied category theory to cell biology to develop a theory of "relational biology" as an alternative to the reductionist analytical methods still prevalent to this day. His main result was to prove that the cell metabolism repair function performed by the DNA is invertible into a DNA repair function performed by the cell metabolism. Hence the cell is 'self-sufficient' in terms of information, it contains all the information it needs to repair all of its parts. Of course we already knew that the cell is able to repair its DNA, but for our purposes it is very good to know that the same mathematical theory that can map automata to logic and dynamical systems is also able to capture important properties of the cell. Rosen's result has more recently been interpreted ([14]) as the mathematical analogue of Maturana and Varela's "operational closure" (or organisational closure) within the theory of autopoiesis [44]. In spite of the fact that Rosen's subsequent generalisation of this proof into a much more ambitious 'theory of Life' [54] has recently been criticised and has been the subject of a lively debate ([11, 13, 12, 41, 65]), Rosen should be credited with a simple but insightful observation:

> ... systems of the utmost structural diversity, with scarcely a molecule in common, are nevertheless recognizable as cells. This indicates that the essential features of cellular organization can be manifested by a profusion of systems of quite different structure. [53]

In other words, all cells, regardless of their structure, share a similar organisation. However, depending on their function, cells can have very different structure. This suggests that **Structure, Function,** and **Organisation** are equally fundamental concepts in biology.

In computer science, on the other hand, things are a bit different. In analogue computer systems the computation to be performed (Function) was strictly dependent on the electronic components utilised and their wiring (Structure). Digital computers,

by contrast, were developed as "general-purpose machines" through extensive use of abstraction/layering. In contrast to biology and analogue computers, there is very little interdependence between Structure and Function in digital computers – by design! However, Organisation does map well from biology to computer science, where it is called Architecture. An interesting example of the applicability of these concepts is provided by the "conscientious software" of Gabriel and Goldman [30], who identify software that performs some useful external function as "allopoietic", in symbiotic coexistence with software that keeps the system alive as "autopoietic". A related concept that is similar to operational closure and that is a current focus of our research is to wire different allopoietic components together in order to form an autopoietic whole. A more in-depth discussion of autopoiesis-inspired computing can be found in another paper being presented at this conference [9].

The complexity of the problem and of the theory that is emerging is making it difficult to keep the various analogies, metaphors, and models straight, partly because the concepts apply at very different levels of abstraction. Table 1 provides a possible mapping between how these three fundamental concepts apply in biology, mathematics and computer science.

Table 1. Examples of how the fundamental properties of biology might map to other domains

	Biology	**Mathematics**	**Computer Science**
Structure	Shape of nerve cell	Group structure of cellular pathways	sequential/ parallel/ concurrent
Function	Nerve signal conduction	Metabolic pathway	Algorithm Behaviour
Organisation	Operational closure	Group closure property	Autopoietic architecture

3.6 Gene Expression Computing, or Interaction Computing

In reference to Figure 5, proto-evolutionary mechanisms in the primordial soup bootstrapped resilient organisational forms such as hypercycles [28] and auto-catalytic cycles [37] from random physical interactions. After the membrane emerged as a structure that could delimit an 'inside' from an 'outside', these so-called molecular replicators eventually led to the emergence of the cell with its autopoietic properties (organisationally closed, recursively self-generating). As we argued above, cellular pathways today are still driven by the same interaction and entropic physical processes. Thus, if we wish to emulate, in software, principles from biology that can rightfully claim 'fundamental' status, in its most general form context-sensitivity must work both ways, which argues for a reciprocal and pervasive interaction model.

Our work is inspired by the observation that the computation performed by a biological ecosystem can be conceptualised as a theoretical limit characterised by the number of peers in a distributed P2P architecture approaching infinity, with the amount of traditional computation performed by each approaching zero. This analogy can also be extended to the 'computation' performed by the cell's cytoplasm. More

precisely, the computation performed by biological systems always involves at least two entities, each of which is performing a different, and often independent, algorithm which can only be advanced to its next state by the interaction itself. This is the kernel of the concept of interaction computing or gene expression computing. We wish to explore the implications of such a 'vanishing CPU' scenario because by providing a mathematical foundation to building nested and recursively interacting structures we believe that it underpins a model of emergent computation that will lead to new insights in biology and computer science, in equal measure.

This explains why we are trying to develop an emergent model of computation by mapping the regulatory and metabolic biochemical pathways of the cell to interacting automata. Such a model of computation will both require and enable a shift from a reliance on human design as the only source of order in software towards a greater reliance on information and structures built into the environment. In fact, the complexity of the cell's interior suggests that in the cell 'interaction' can acquire significantly greater semantics than, for example, perfect collisions between point particles in an ideal gas. We then notice that the cell is itself surrounded by other cells with which it communicates, and all are embedded in a complex mixture of tissues and fluids that form organs. Organs, in turn, cooperate in the functioning of individuals, which interact to form biological ecosystems. Thus, interactions happen at all scales within the nested and recursively organised hierarchical structure of all biological systems.

3.7 Computational Medium and RESTful Architecture

Interaction signals in biological systems are mediated in physical space by the solid, liquid or gaseous media that fill it (with the exception of light, which does not need a medium). Software systems, by contrast, do not interact over continuous metric spaces, they interact over topological spaces, or networks. By 'network' we do not mean simply the IP layer or below, we mean the term in the most general possible sense, applicable as a medium of low-abstraction signals, of application layer protocols, or of semantic and knowledge networks. In order to provide a roadmap of applicability to instantiate the theoretical and mathematical results of the project into the software and web environments of the future we need to understand how distributed and networked systems can support the interaction or gene expression computing models and their recursive application.

Our starting point for the development of a run-time framework that is general enough to support the mathematical results and that is relevant to today's web computing environments is a RESTful architecture for the definition of a message-passing interaction model for distributed environments. REST (Representational State Transfer [29]) in general, and the REST over HTTP architecture of the web specifically, constitutes a language in which interaction can be considered a primitive element. The REST architectural style has been conceived to reflect the architecture of the web. Since the architecture of the web is constrained at the lowest levels to enable extensibility at higher levels, higher-order capabilities such as support for complex interactions that require transactional guarantees (e.g. in long-running service applications) and querying languages can be constructed on top of it.

4 Conclusion

The aim of this paper was to provide a broad research framework through which the rationale of more focussed research activities could be understood [62, 35, 23, 9]. Much work remains to be done. However we hope that the framework we have presented here will appear plausible enough to attract more computer scientists, mathematicians, and cell biologists in the development of a common and unified theory of bio-computing for autopoietic digital ecosystems.

Acknowledgements

The authors wish to thank Dr Sotiris Moschoyiannis of the University of Surrey for clarifying how the RESTful perspective could be connected to the concept of interaction computing presented here. The support for this work by the OPAALS (FP6-034824) and the BIONETS (FP6-027748) EU projects is gratefully acknowledged.

References

1. Andréka, H., Neméti, I., Sain, I.: Universal Algebraic Logic, 1st edn. Studies in Universal Logic. Springer, Heidelberg (to appear)
2. Bergstra, J.A., Klop, J.W.: ACP$_\tau$: a universal axiom system for process specification, pp. 447–463 (1989)
3. Bjorner, D., Jones, C.B. (eds.): The Vienna Development Method: The Meta-Language. LNCS, vol. 61. Springer, Heidelberg (1978)
4. Bolognesi, T., Brinksma, E.: Introduction to the ISO specification language LOTOS. Comput. Netw. ISDN Syst. 14(1), 25–59 (1987)
5. Briscoe, G.: D6.1-Entropy-Based Complexity Measure for the Evolution-Based Self-Organisation of Agent Populations. DBE Project (2004), http://files.opaals.org/DBE/deliverables
6. Briscoe, G., De Wilde, P.: D6.2-Self-Organisation of Evolving Service Populations. DBE Project (2005), http://files.opaals.org/DBE/deliverables
7. Briscoe, G., De Wilde, P.: D6.4-Intelligence, learning and neural networks in distributed agent systems. DBE Project (2005), http://files.opaals.org/DBE/deliverables
8. Briscoe, G., De Wilde, P.: D6.5-The effect of distributed intelligence in evolutionary dynamics. DBE Project (2006), http://files.opaals.org/DBE/deliverables
9. Briscoe, G., Dini, P.: Towards Autopoietic Computing. In: Proceedings of the 3rd OPAALS International Conference, Aracaju, Sergipe, Brazil, March 22-23 (2010)
10. Buckingham, E.: The principle of similitude. Nature 96, 396–397 (1915)
11. Chu, D., Ho, W.K.: A Category Theoretical Argument Against the Possibility of Artificial Life: Robert Rosens Central Proof Revisited. Artificial Life 12, 117–134 (2006)
12. Chu, D., Ho, W.K.: Computational Realizations of Living Systems. Artificial Life 13, 369–381 (2007)
13. Chu, D., Ho, W.K.: The Localization Hypothesis and Machines. Artificial Life 13, 299–302 (2007)

14. Cornish-Bowden, A., Cardenas, M.L.: Self-organization at the origin of life. Journal of Theoretical Biology 252, 411–418 (2008)
15. Dini, P.: D18.4-Report on self-organisation from a dynamical systems and computer science viewpoint. DBE Project (2007), http://files.opaals.org/DBE
16. Dini, P., Berdou, E.: D18.1-Report on DBE-Specific Use Cases. DBE Project (2004), http://files.opaals.org/DBE
17. Dini, P., Briscoe, G., Munro, A.J., Lain, S.: D1.1: Towards a Biological and Mathematical Framework for Interaction Computing. OPAALS Deliverable, European Commission (2008), http://files.opaals.org/OPAALS/Year_2_Deliverables/WP01/
18. Dini, P., Horvath, G., Schreckling, D., Pfeffer, H.: D2.2.9: Mathematical Framework for Interaction Computing with Applications to Security and Service Choreography. BIONETS Deliverable, European Commission (2009), http://www.bionets.eu
19. Dini, P., Schreckling, D.: More Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. In: 1st OPAALS Workshop, Rome, November 26-27 (2007)
20. Dini, P., Schreckling, D.: On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. In: Altman, E., Dini, P., Miorandi, D., Schreckling, D. (eds.) D2.1.1 Paradigms and Foundations of BIONETS research (2007)
21. Dini, P., Schreckling, D.: Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. In: 2nd International Conference on Digital Ecosystems and Technologies, IEEE-DEST 2008, February 26-29 (2008)
22. Dini, P., Schreckling, D., Yamamoto, L.: D2.2.4: Evolution and Gene Expression in BIONETS: A Mathematical and Experimental Framework. BIONETS Deliverable, European Commission (2008), http://www.bionets.eu
23. Egri-Nagy, A., Dini, P., Nehaniv, C.L., Schilstra, M.J.: Transformation Semigroups as Constructive Dynamical Spaces. In: Proceedings of the 3rd OPAALS International Conference, Aracaju, Sergipe, Brazil, March 22-23 (2010)
24. Egri-Nagy, A., Nehaniv, C.L.: Algebraic Properties of Automata Associated to Petri Nets and Applications to Computation in Biological Systems. BioSystems 94(1-2), 135–144 (2008)
25. Egri-Nagy, A., Nehaniv, C.L., Rhodes, J.L., Schilstra, M.J.: Automatic Analysis of Computation in Biochemical Reactions. BioSystems 94(1-2), 126–134 (2008)
26. Egri-Nagy, A., Nehaniv, C.L.: Hierarchical coordinate systems for understanding complexity and its evolution with applications to genetic regulatory networks. Artificial Life 14(3), 299–312 (2008) (Special Issue on the Evolution of Complexity)
27. Egri-Nagy, A., Nehaniv, C.L.: SgpDec - software package for hierarchical coordinatization of groups and semigroups, implemented in the GAP computer algebra system (2008), http://sgpdec.sf.net
28. Eigen, M., Schuster, P.: The Hypercycle. Naturwissenschaften 65(1) (1978)
29. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. UC Irvine PhD Dissertation (2000), http://www.ics.uci.edu/fielding/pubs/dissertation/top.htm
30. Gabriel, R.P., Goldman, R.: Conscientious software. In: OOPSLA'06, Portland, Oregon, October 22-26 (2006)
31. Golubitsky, M., Stewart, I.: Nonlinear Dynamics of Networks: The Groupoid Formalism. Bulletin of the American Mathematical Society 43, 305–364 (2006)
32. Heistracher, T., Kurz, T., Marcon, G., Masuch, C.: D9.1-Report on Fitness Landscape. DBE Project (2005), http://files.opaals.org/DBE/deliverables

33. Hoare, C.A.R.: Communicating sequential processes. ACM Commun. 21(8), 666–677 (1978)
34. Horvath, G.: Functions and Polynomials over Finite Groups from the Computational Perspective. The University of Hertfordshire, PhD Dissertation (2008)
35. Horvath, G., Dini, P.: Lie Group Analysis of p53-mdm3 Pathway. In: Proceedings of the 3rd OPAALS International Conference, Aracaju, Sergipe, Brazil, March 22-23 (2010)
36. Kalman, R.E., Falb, P.L., Arbib, M.A.: Topics in Mathematical System Theory. McGraw-Hill, New York (1969)
37. Kauffman, S.: The Origins of Order: Self-Organisation and Selection in Evolution. Oxford University Press, Oxford (1993)
38. Krohn, K., Rhodes, J.: Algebraic Theory of Machines. I. Prime Decomposition Theorem for Finite Semigroups and Machines. Transactions of the American Mathematical Society 116, 450–464 (1965)
39. Kurz, T., Marcon, G., Okada, H., Heistracher, T., Passani, A.: D9.2-Report on Evolutionary and Distributed Fitness Environment. DBE Project (2006), http://files.opaals.org/DBE/deliverables
40. Lahti, J., Huusko, J., Miorandi, D., Bassbouss, L., Pfeffer, H., Dini, P., Horvath, G., Elaluf-Calderwood, S., Schreckling, D., Yamamoto, L.: D3.2.7: Autonomic Services within the BIONETS SerWorks Architecture. BIONETS Deliverable, European Commission (2009), http://www.bionets.eu
41. Louie, A.H.: A Living System Must Have Noncomputable Models. Artificial Life 13, 293–297 (2007)
42. Maeda, S.: The similarity method for difference equations. IMA Journal of Applied Mathematics 38, 129–134 (1987)
43. Manrubia, S.C., Mikhailov, A.S., Zanette, D.H.: Emergence of Dynamical Order. World Scientific, Singapore (2004)
44. Maturana, H., Varela, F.: Autopoiesis and Cognition, the Realization of the Living. D. Reidel Publishing Company, Boston (1980)
45. Milner, R.: A Calculus of Communication Systems. LNCS, vol. 92. Springer, Heidelberg (1980)
46. Milner, R., Parrow, J., Walker, D.: A Calculus of Mobile Processes, I. Inf. Comput. 100(1), 1–40 (1992)
47. Nehaniv, C.L., Rhodes, J.L.: The Evolution and Understanding of Hierarchical Complexity in Biology from an Algebraic Perspective. Artificial Life 6, 45–67 (2000)
48. Nicolis, G., Prigogine, I.: Self-Organization in Nonequilibrium Systems. Wiley, New York (1977)
49. Olver, P.: Applications of Lie Groups to Differential Equations. Springer, Heidelberg (1986)
50. Rashevsky, N.: Mathematical Biophysics and Physico-Mathematical Foundations of Biology, vol. II. Dover, New York (1960)
51. Rosen, R.: A Relational Theory of Biological Systems. Bulletin of Mathematical Biophysics 20, 245–260 (1958)
52. Rosen, R.: The Representation of Biological Systems from the Standpoint of the Theory of Categories. Bulletin of Mathematical Biophysics 20, 317–341 (1958)
53. Rosen, R.: Some relational cell models: The metabolism-repair systems. In: Rosen, R. (ed.) Foundations of Mathematical Biology. Cellular Systems, vol. II, Academic Press, London (1972)
54. Rosen, R.: Life Itself. Columbia University Press, New York (1991)

55. Rowe, J.E., Mitavskiy, B.: D8.1 - report on evolution of high-level software components (April 2005)
56. Sanchez, R., Grau, R., Morgado, E.: A novel Lie algebra of the genetic code over the Galois field of four DNA bases. Mathematical Biosciences 202, 156–174 (2006)
57. Sanchez, R., Morgado, E., Grau, R.: The genetic code boolean lattice. Communications in Mathematical and Computational Chemistry 52, 29–46 (2004)
58. Sanchez, R., Morgado, E., Grau, R.: Gene algebra from a genetic code algebraic structure. Journal of Mathematical Biology 51, 431–457 (2005)
59. Schreckling, D., Dini, P.: Distributed Online Evolution: An Algebraic Problem? In: IEEE 10th Congress on Evolutionary Computation, Trondheim, Norway, May 18-21 (2009)
60. Spivey, J.M.: The Z notation:a reference manual. Prentice-Hall, Inc., Upper Saddle River (1989)
61. Stewart, I.: Galois Theory, 2nd edn. Chapman and Hall, London (1989)
62. Van Leeuwen, I., Munro, A.J., Sanders, I., Staples, O., Lain, S.: Numerical and Experimental Analysis of the p53-mdm2 Regulatory Pathway. In: Proceedings of the 3rd OPAALS International Conference, Aracaju, Sergipe, Brazil, March 22-23 (2010)
63. Voss, R.F., Clarke, J.: 1/f noise in music: Music from 1/f noise. Journal of the Acoustical Society of America 63(1) (1978)
64. Weinstein, A.: Groupoids: unifying internal and external symmetry. Notices of the American Mathematical Society 43, 744–752 (1996)
65. Wolkenhauer, O.: Interpreting Rosen. Artificial Life 13, 291–292 (2007)
66. Wu, C.W.: Synchronization in Coupled Chaotic Circuits and Systems. World Scientific, Singapore (2002)