OPAALS

Contract N^{o} IST-034824

WP1:
Cell Biology, Autopoiesis and Biological Design Patterns

D1.3:
Biological Design Patterns of Autopoietic Behaviour in Digital Ecosystems

Project funded by the European Community under the "Information Society Technology" Programme.

Contract Number: IST-034824
Project Acronym: OPAALS

Deliverable No: D1.3
Due date: 31/05/2009
Delivery date: 3/09/2009

Short Description:
This report discusses interdisciplinary research leading to a mathematical framework for biologically inspired computing aimed at achieving the project's objectives in self-organising software behaviour and the first steps towards an autopoietic digital ecosystem model.

Authors: Paolo Dini and Gerard Briscoe (LSE), Ingeborg Van Leeuwen, Alastair Munro and Sonia Lain (UNIVDUN)
Partners contributed: Chrystopher Nehaniv and Attila Egri-Nagy (UH)
Made available to: Public distribution

Version	Date	Author, organisation
1	31/06/09	Paolo Dini, Gerard Briscoe (LSE), Ingeborg Van Leeuwen, Alastair Munro, Sonia Lain (UNIVDUN)
2	24/08/09	Paolo Dini, Gerard Briscoe (LSE), Ingeborg Van Leeuwen, Alastair Munro (UNIVDUN)

Quality Check:
1st Internal Reviewer: Ossi Nykänen, TUT
2nd Internal Reviewer: Maria Schilstra, UH
3rd Internal Reviewer: Sotiris Moschoyiannis, Surrey

Dependencies

Achievements*	Done: Developed broad and interdisciplinary research programme for bio-computing with a mathematical focus Demonstrated influence of marker on p53-mdm2 oscillations Began charting new concepts of autopoietic computing Not done: Behind in the discussion of the algebraic decomposition of automata
Workpackages	WP1
Partners	LSE, UNIVDUN, UH
Domains	Evolutionary and Interactive Computation
Targets	WP1, WP3, bio-computing researchers

Publications*	Van Leeuwen, I and Lain, S (2009). "Sirtuins and p53", *Advances in Cancer Research*. Schreckling, D and Dini, P (2009). "Distributed Online Evolution: An Algebraic Problem?" In IEEE 10th Congress on Evolutionary Computation, Trondheim, Norway, 18-21 May. Van Leeuwen, I (2009). "The p53 pathway: complexity, oscillations and therapeutic exploitation", Invited Lecture, EU-USA Strategic Workshop on Cancer Systems Biology, Rostock, Germany, 7-11 June. Gordon K E, Van Leeuwen I, Lan S, Chaplain M A J (2009). "Spatio-temporal modelling of the p53-mdm2 oscillatory system", *Mathematical Modelling of Natural Phenomena*, Vol 4, N 3.
PhD Students*	Gerard Briscoe (completed PhD in March 09)
Outstanding features*	New integrative mathematical framework for bio-computing New insights in p53-mdm2 behaviour New ideas on autopoiesis and computing
Disciplinary domains of authors*	Dini: aerospace engineering, physics, mathematics, social science Briscoe: computer science Van Leeuwen: mathematical biology Munro: cancer medicine, theoretical biology Lain: experimental biology, drug development

* Indicates information requested by reviewers

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit: http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Acknowledgments

The work reported in Chapter 3 has been partly funded through OPAALS but has also been supported by other sources (CRUK, The Ninewells Cancer Campaign, a bequest from the estate of Mrs D.B. Miller). Individuals contributing to the data presented include: Ian Sanders; Oliver Staples; Liz Sinclair and Jo Campbell.

We are grateful to Dr Attila Egri-Nagy (UH) for writing Section 2.1.

We are grateful to Prof Juan Parrondo from the Department of Atomic, Molecular and Nuclear Physics, Universidad Complutense de Madrid, for developing the original damped p53-mdm2 model, an extension of which is discussed in Section 3.3.

We are grateful to Daniel Schreckling of the University of Passau for writing sub-section 1.2.4.

Contents

Executive Summary

This report summarises the work performed in WP1 during Year 3 of the project. The research in WP1 focussed on three main activities: continued experimental characterisation of the p53-mdm2 regulatory cycle, algebraic foundations of bio-computing, and new autopoietic ideas in computer science. The experimental work has led to the demostration that the oscillations in the p53-mdm2 cycle that have been widely reported in the literature stemmed mainly from the interaction of the marker with the biochemical reactants, and that the character of the oscillations is actually damped and less pronounced. This experimental work feeds into a Lie group analysis of the ordinary differential equations derived from the p53-mdm2 biochemical rate equations. This analysis is taking place in the BIONETS project. In parallel, the same equations are used to derive a finite-state automaton whose algebraic structure is analysed with Krohn-Rhodes theory. This analysis is not presented in this report but will appear in the next, D1.4. The research aims to relate the dynamic and static symmetries, respectively, that can be obtained through these two analytical approaches. We believe that by following this approach we will be able to derive valuable insights about the dynamical and computational structure of self-organising systems in a manner that will be relatively easy to transfer to computer science.

The algebraic analysis of the biological automata has continued in collaboration with partners in both the OPAALS and the BIONETS projects, with less progress on the OPAALS side largely due to the additional work incurred in the preparation of the Interim review in April 09. New ideas on autopoietic computing are discussed in the final chapter of this report; this provides a very different perspective on computing in general that is currently being pursued in collaboration with the IITK partner. Progress relative to the final objectives of convergence of this work with WP3 has been made in the development of a broad interdisciplinary framework with a strong mathematical focus that spans multiple projects and that is discussed in Chapter 1. In short, this deliverable should be seen as a status report on a complex and multi-threaded research programme that is growing in depth and scope and that is attracting a growing number of researchers from multiple disciplines around the theme of bio-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 and self-healing 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 [35].

This report continues the construction of a theory of bio-computing that was begun in D1.1 [35] and in the DBE project before that. In parallel, different aspects of the same theory of bio-computing have been pursued also in the BIONETS project, in such a way as to render the two research threads (OPAALS and BIONETS) complementary.[1] Keeping the efforts complementary is important from the point of view of resource optimisation, since different skill sets are present in the two projects. Furthermore, in the two projects the application emphasis is somewhat different. In OPAALS the application areas of interest are:

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

In BIONETS the application areas of interest are:

- Symbiotic security
- Biologically-inspired service choreography (which could be RESTful)

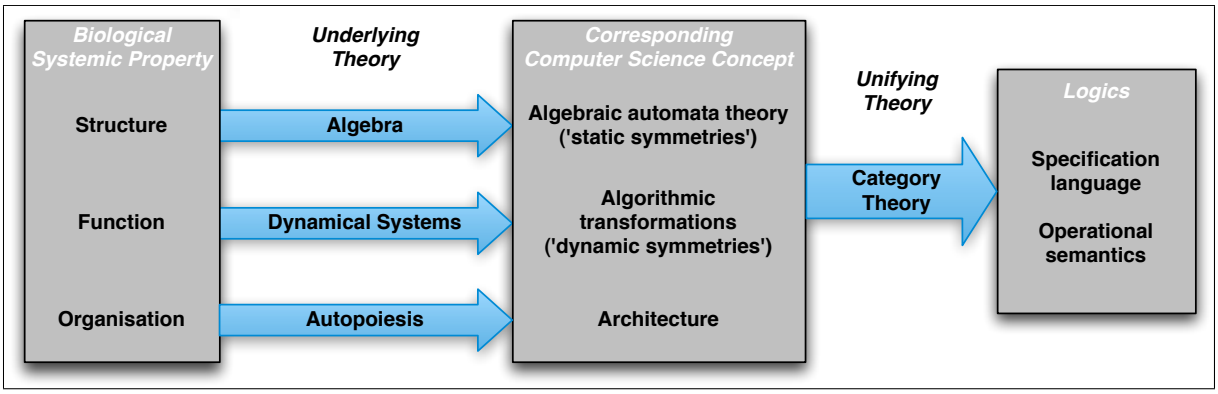

Figure 1: **High-level view of the theoretical research framework**

By way of preview, 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 chapter and in the rest of the report. 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

[1] As a consequence, the introductory chapters of this deliverable and of BIONETS deliverable D2.2.9 [36], which are being written at the same time, are almost identical.

of automata, and novel ideas in software architectures and biological design patterns inspired by autopoiesis. Category theory is then able to effect an additional mapping from any of the above algebraic structures to logic. Instantiation of this framework in modern distributed and web-oriented computing environments requires enriching the logics with operational semantics in order to develop suitable executable specification languages, and indicates that it may be possible to develop an interaction model based on the Representational State Transfer (REST) architectural style. It is important to emphasise that the term "structure" is quite overloaded in this report. It can refer to biological (physical) structure or to algebraic structure. Then, in Chapter 4, we use structure also in a software context, but as something separate from architecture and that will not become fully clear until a formalisation for interaction computing and its derivatives has been developed. Hopefully these various meanings will be clear from the context.

In this Introduction we now retrace the arguments and rationale that we have developed over the past 6 years in this area of research.

1.1 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 these two projects, 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.

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.

1.1.1 Evolution and self-organisation

The current research thread in gene expression or interaction computing being reported in OPAALS D1.3 and BIONETS D2.2.9 [36] began with a discussion of self-organisation through the minimisation of free energy, in DBE D18.1 [34]. 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 constant 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 constant flow of information through the system and a constant poking and prodding by the users that can be seen as analogous to a certain level of thermodynamic 'mixing'. If we abstract

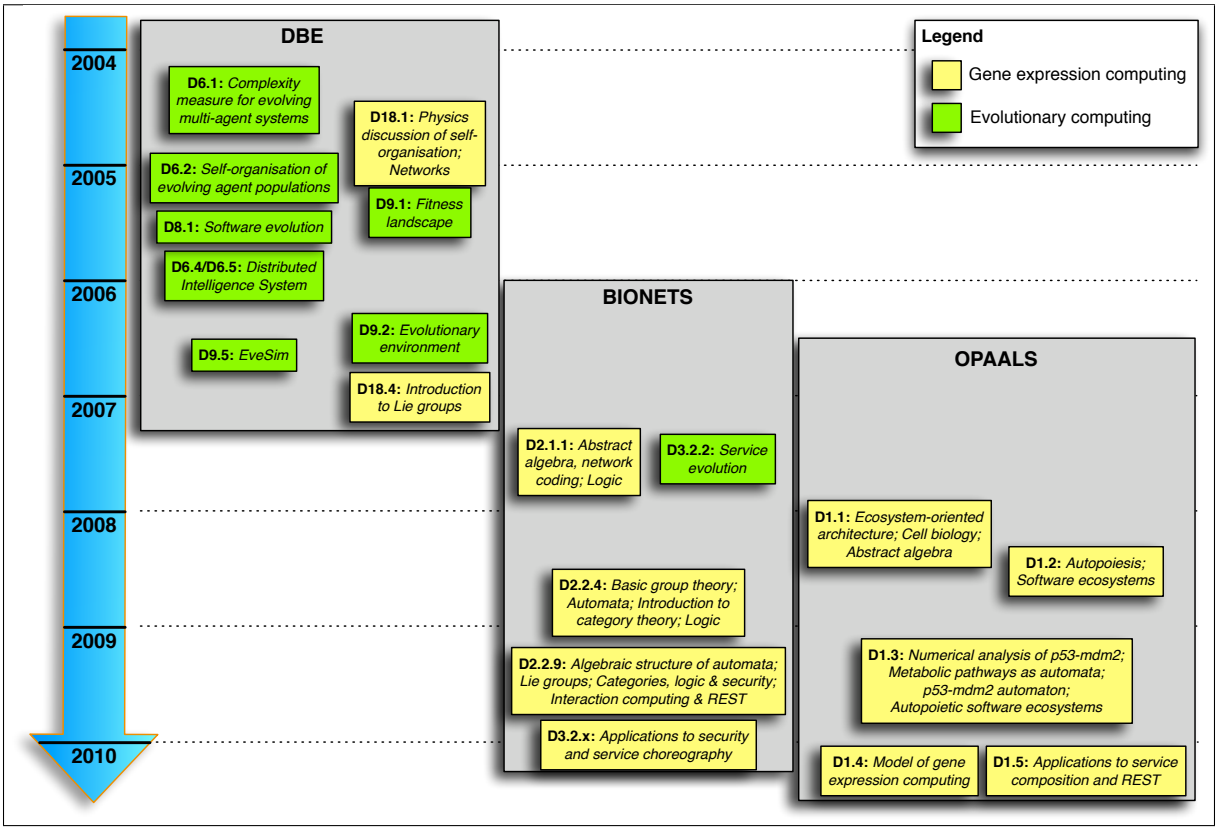

Figure 2: **History of relevant bio-computing reports across several projects**

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 (D9.1 [67]; D9.2 [81]; D8.1 [132]; D6.1 [17]; and D6.2 [18]). 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 (D6.4 and D6.5 [19, 20]), 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.[2] Gene expression computing refers to the nuts and bolts of cellular pathways and how they are able to contruct 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 Chapter 4. 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.

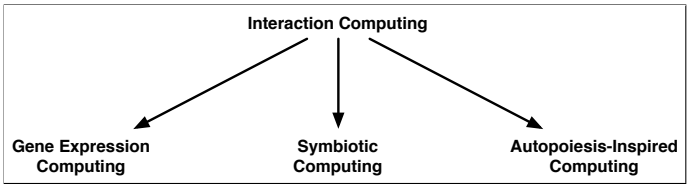

Figure 3: **Different possible models of computation derived from Interaction Computing**

This prioritisation of ontogeny over phylogeny implied that an in-depth investigation 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.

1.1.2 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 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

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

of its structure invariant form a group.[3] 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.

1.1.3 Lie groups

In DBE D18.4 [33] 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.[4]

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 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 (D2.1.1 in BIONETS [39] and D1.1 in OPAALS [35]).[5]

At about the same time we ran across the work of the Cuban HIV researchers Sanchez, Morgado, and Grau [134, 135, 133], 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

[3] Paraphrased from Stewart ([140]:xxvii).

[4] See Maeda [94] and Peter Hydon's work at http://personal.maths.surrey.ac.uk/st/P.Hydon/sym.htm.

[5] The algebra work in these two deliverables was done mainly by P Dini, who splits his time evenly between the two projects.

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.[6] 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, or 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 perturbations 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.

In their more recent work Sanchez, Morgado, and Grau [133] 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, 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 within BIONETS (D2.2.9 [36]) is

[6]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: $4^3 = 64$.

to extend the work begun in DBE D18.4 and hopefully perform a Lie group analysis of the P53-mdm2 regulatory pathway by the final report in that project (D3.2.7).

1.1.4 Functional completeness

There is one more topic that provides an important background to our research: functional completeness [72]. 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.[7] 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.

For example, Stokes's problem in fluid mechanics, or the velocity induced in a viscous fluid at rest by the instantaneous motion of an infinite plate parallel to the x axis immersed in the fluid, as it jumps from 0 to a constant velocity V_0 in the x direction (in other words the plate starts moving in its own plane). In this case the flow is clearly only tangential to the plate, and the velocity v decreases with perpendicular distance y from the plate. However, as time progresses the momentum of the moving plate is diffused in the y direction, meaning that areas of the fluid far away from the plate, which were originally at rest, will eventually 'feel' the motion of the plate and will also start moving, gradually approaching the constant velocity V_0 of the plate without ever reaching it. Therefore, this problem has 2 independent variables: y and t and one dependent variable $v(y,t)$; and is thus modelled by a partial differential equation (PDE), which turns out to be isomorphic to the Heat Diffusion Equation of classical physics. Interestingly, if one divides y by the square-root of t, the governing PDE turns into an ordinary differential equation (ODE). In other words, $v(y,t)$ becomes $v(y^*)$, a function of the single independent variable y^* obtained through the similarity transformation:

$$y^* = \frac{y}{\sqrt{t}}$$

Such groupings of variables can be derived systematically using Lie groups.

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 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

[7] E.g. see the famous Buckingham Pi Theorem [22] and generalisations thereof.

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).[8] 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. This is similar to the case of Stokes's flow: we don't have to use the similarity transformation, we can simply solve the PDE to solve the same physical problem; however, the similarity transformation simplifies the mathematics and improves physical understanding. Similarly, 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, some of which are addressed by this report and by D2.2.9 [36].

1.2 Background of Current Work

1.2.1 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.

In particular, our perspective views the stochastic nature of cell biochemistry mainly as a mechanism of dimensional reduction[9] 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

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

[9] 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 (ODE), successful abstraction and dimensional reduction can lead to significant theoretical insight and savings in CPU requirements.

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.[10] 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.

1.2.2 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 ([152, 95]). 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 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 [148]. 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 [113] 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.

[10] 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.

1.2.3 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 mixture of algebra and geometry and gives rise to what we are calling dynamic symmetries.

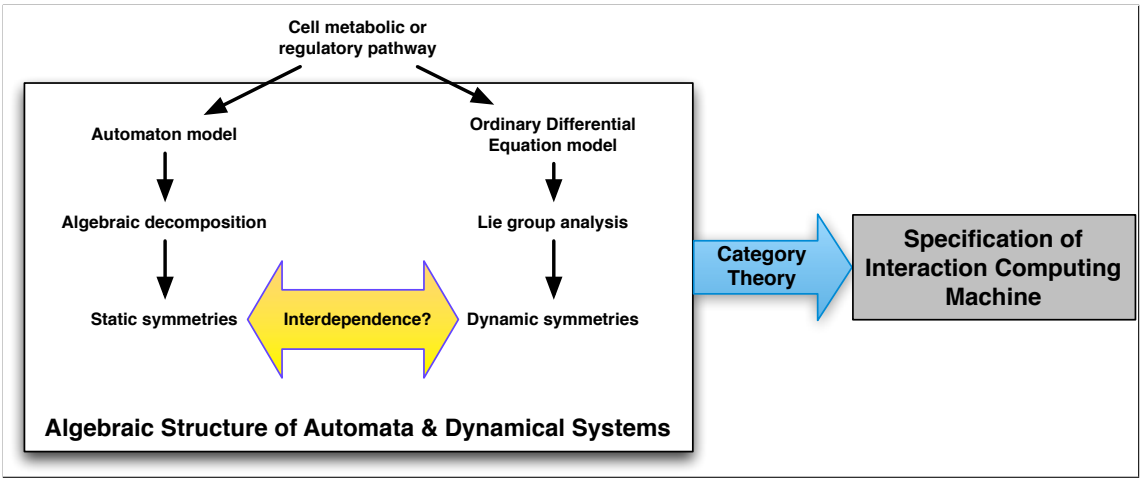

Figure 4: **Mathematical analysis workflow to uncover biological symmetries**

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 finite 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 [77], 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 ([73]: 8) 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 ([107, 47, 46, 45, 48]), 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.

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 [116]. 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. ([151]; quoted in [62])

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.

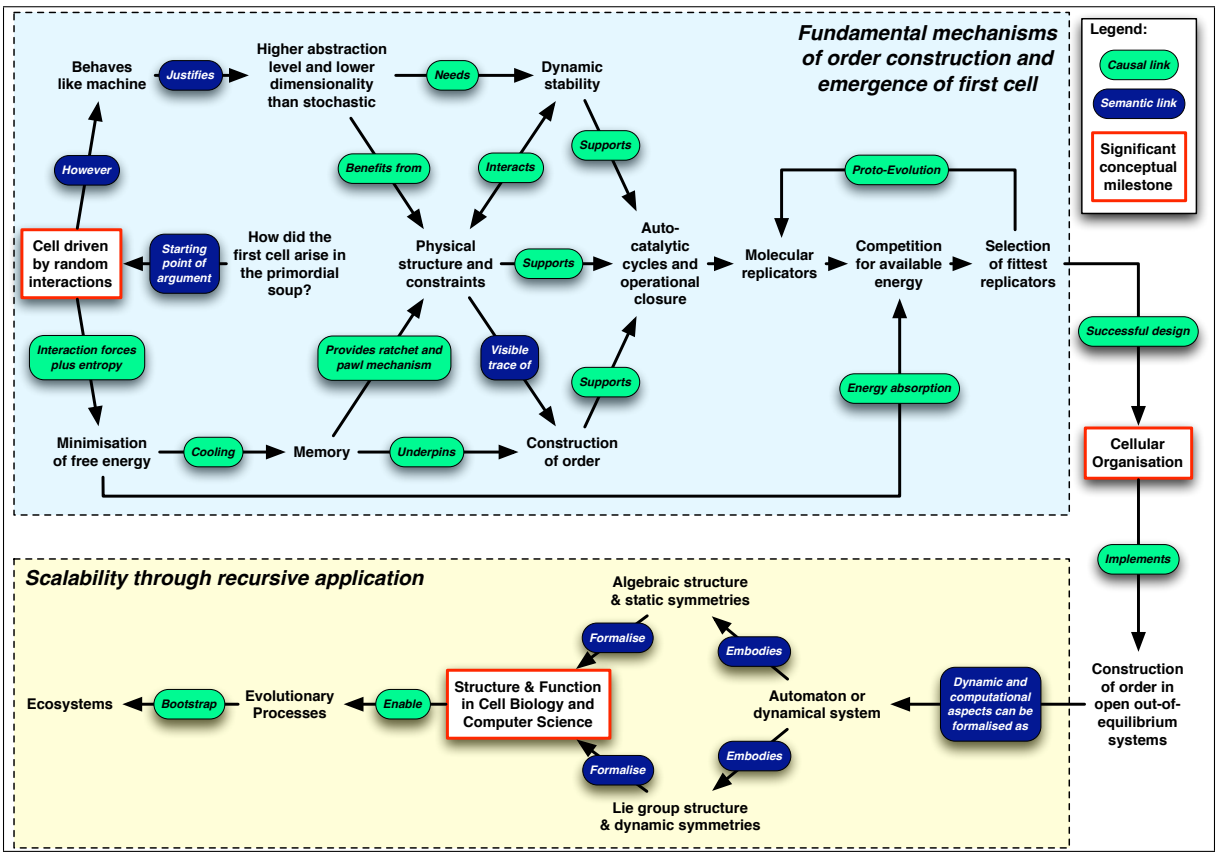

Figure 5: **Causal-semantic workflow summarising a part of the research rationale**

1.2.4 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 [12], Z notation [138], CCS [103], π-calculus [104], CSP [70], LOTOS [14], ACP_τ [8], 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 Chapter **??** we study how the two concepts of machine structure and its behaviour are strongly linked in categorical terms. 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 [3] 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 [41]), 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 deliverable forms the basis of developing 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 and has been reported in BIONETS deliverables D2.1.1 [39] and D2.2.4 [41], and related papers [40, 38, 136]. It continues in BIONETS deliverable D2.2.9 [36], elaborating concepts which map algebraic structure corresponding to automata into categories of behaviour. While this deliverable and D2.2.9 still explore the foundations of a large research agenda, the foundations collected in the listed deliverables appear to close a circle which will hopefully allow us to show more applied results in D1.4, D1.5 and BIONETS D3.2.7.

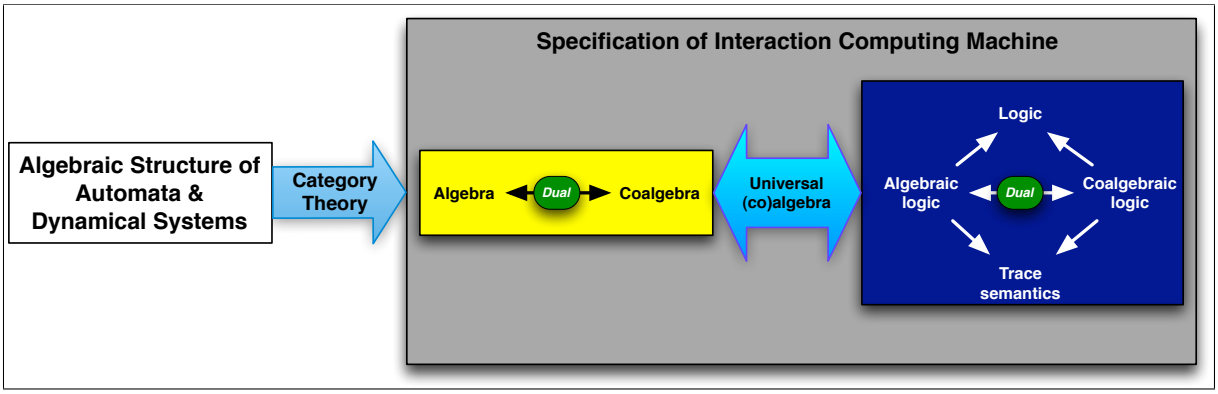

Figure 6: **Mapping of algebraic structures to logic structures through category theory**

1.2.5 Organisation in Biology and Computer Science

The reliance on category theory is further motivated by Rosen [127, 128] who, following Rashevsky's ideas [122], 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 ([31]) as the mathematical analogue of Maturana and Varela's "operational closure" (or organisational closure) within the theory of autopoiesis [97]. In spite of the fact that Rosen's subsequent generalisation of this proof into a much more ambitious 'theory of Life' [131] has recently been criticised and has been the subject of a lively debate ([26, 28, 27, 91, 92]), 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. [130]

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 [56], 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.

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.

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

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

1.2.6 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 [49] and autocatalytic cycles [75] 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 hopefully 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. The actual model of computation based on interacting automata will be developed in OPAALS deliverable D1.4. 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.

1.2.7 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 [54]) 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.

1.3 Structure of Current Reports

Having given a broad and historical overview of our research activities across 3 projects over the past 7 years, we can finally provide a overview of this report and of D2.2.9 [36], as shown in Figure 7. Chapters 2 and 3 of D1.3, Chapters 2 and 3 of D2.2.9 [36], and D1.4 are more focussed on the **Structure** and **Function** aspects of biology and their mapping to computer science. By contrast, Chapter 4 of D1.3, Chapter 4 of D2.2.9 [36] and D1.5 are more focussed on the **Organisation** aspect. We cannot guarantee that we will complete the mathematical foundations of gene expression or interaction computing in OPAALS and BIONETS. We will carry the theoretical work as far as we can, and in the final deliverables of these two projects we will discuss the applied areas where we expect this theory to have an impact, giving suitable proof-of-concept examples. As shown in the figure, we are hoping that this work will continue in a new project, one of whose aims is to achieve total convergence between Bio-Computing

and Computational Systems Biology, in order to motivate a very close collaboration between computer scientists, mathematicians, and cell biologists in the development of a common and unified theory of bio-computing for autopoietic digital ecosystems.

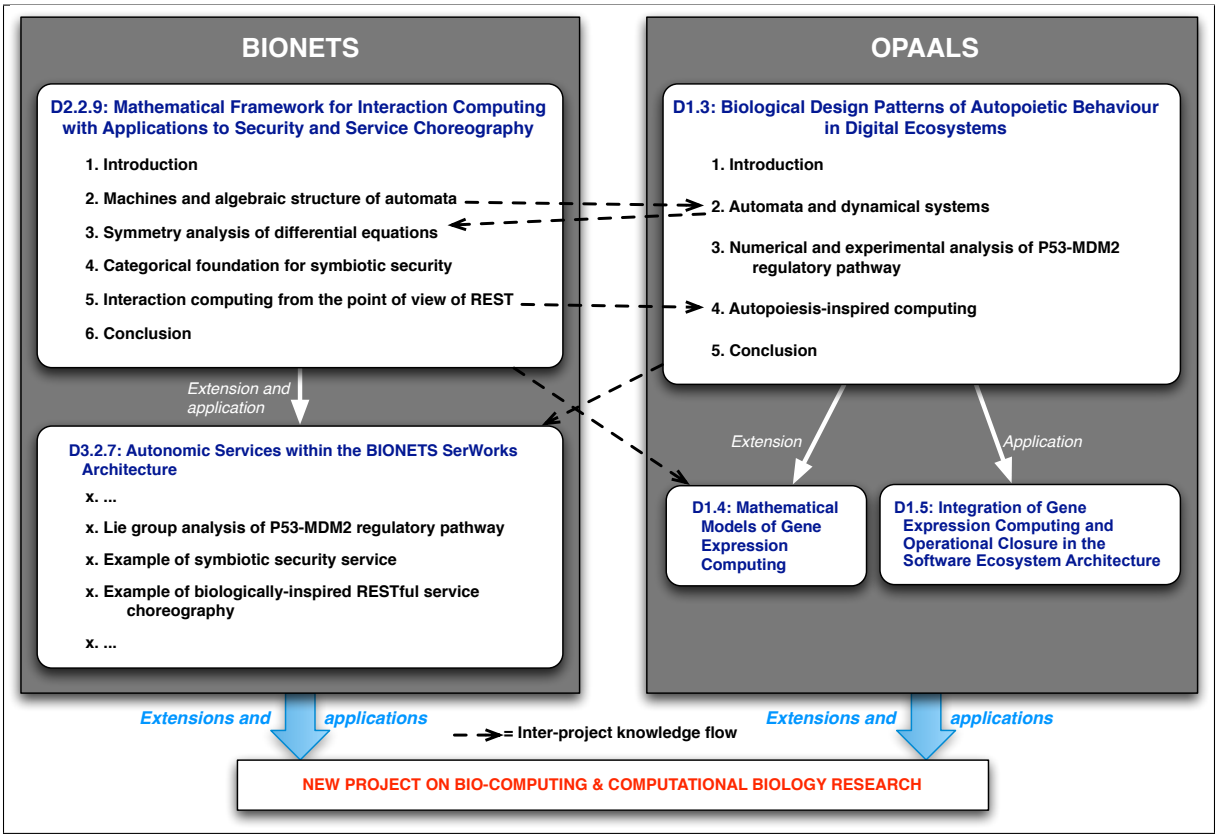

Figure 7: **Current and expected future research in bio-computing**

2 Automata and Dynamical Systems

2.1 Algebraic Decomposition of Automata

One of the most fundamental concepts of science and computation is the notion of change: from a state a system goes to another state (due to external manipulations or internal processes at various time-scales). Finite state automata capture the concept of change in a rigorous way. An automaton consists of a set of states, a set of input symbols, and a state transition function that describes the change of the current state when receiving an input symbol. Owing to its abstract nature, this definition is widely applicable in many different areas of science. Moreover, the questions about finite state automata can be formulated algebraically (the input symbols, considered as transformations of the state set, induce a transformation semigroup since they can be combined associatively); thus the tools of abstract algebra can be used in automata theory.

Our research concerns decompositions of automata, and goes along a line of algebraic methods that allow us to understand, manipulate and predict their behaviour. If there are many states and the transitions are determined by a network of complicated interactions between the components of the system (either visible or hidden), then we have a complex system. In the development of a theory that enables us to understand such complex systems and develop quantitative models and predictions that will eventually enable us to design them, i.e. achieve *system synthesis*, we must begin with the reductionist approach of *system analysis*. As we discussed extensively in OPAALS deliverable D1.2 [37] and in BIONETS deliverable D2.2.4 [41], Rashevsky's Relational Biology agenda, which emerged in direct contrast to the reductionist agenda still prevalent in biology today, remains a very important reference for OPAALS research. As we explained in the Introduction of this report (see Figure 1), however, our research has led us to identify different conceptual and systemic properties of biology, and of computer science, which require commensurably different theoretical approaches that will eventually be integrated into a single unified theory of bio-computing and computational biology. This chapter, which focuses more on the Structure and Function rather than the Organisation aspects of biological and software systems, relies more on the analytic/reductionist than the synthetic/holistic perspective. Therefore, in order to address the complexity of the systems under study, we need to decompose them.

Similar to the prime decomposition of integers the algebraic hierarchical decomposition of finite state automata yields a factorisation in the realm of computational structures: it identifies the basic building blocks and constructs a hierarchical (i.e. *partially ordered*) cascaded structure that can emulate the original computation. The cascaded (hierarchical) decomposition describes the structure of the original automaton in an easily comprehensible way, providing a *coordinate system* defined by the hierarchical levels. The mathematical theory behind this decomposition is the algebraic hierarchical decomposition theory of finite state automata, also called Krohn-Rhodes Theory [78, 43]. There are two types of basic building blocks: one for representing irreversible calculations (the flip-flop automaton with two states and with input symbols resetting to these states, that is basically the capability of storing 1 bit of information), and another type for representing reversible calculations (permutation groups are special kinds of transformation semigroups consisting of bijective transformations of their state set, thus they are invertible). The table below summarises parallels and differences between the prime decomposition of integers (e.g. $60 = 2^2 \times 3 \times 5$) and the algebraic hierarchical decomposition of automata (for the cascaded product see Figure 8).

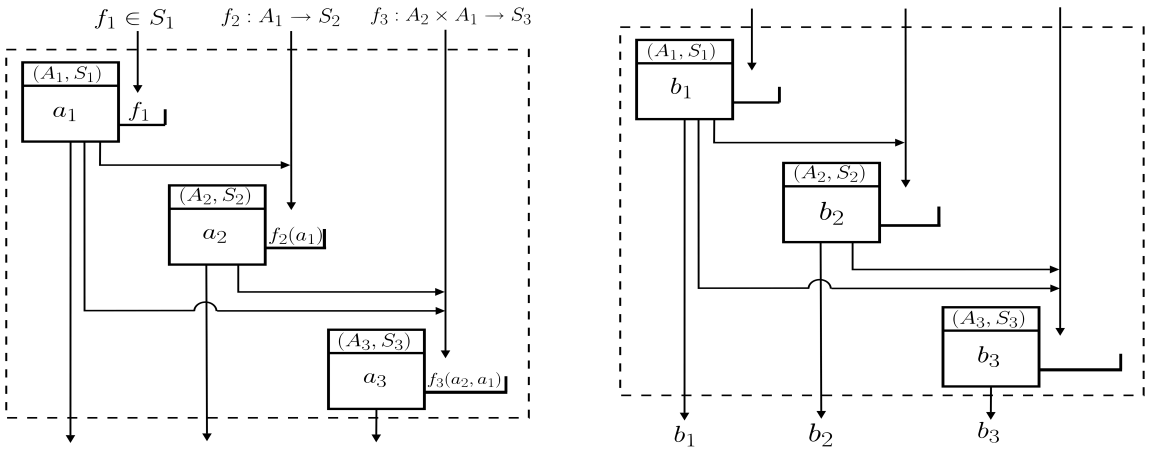

Figure 8: **Example of 3-level coordinate system composed using the cascaded/wreath product of component transformation semigroups:** (A_n, S_n)**,** $n \in \{1, 2, 3\}$**. The composed automaton is enclosed in dashed lines; both its input and output are 3-tuples. Left: For a state transition in the wreath product** $(A_3, S_3) \wr (A_2, S_2) \wr (A_1, S_1)$**, the transformation** (f_3, f_2, f_1) **is applied to state** (a_3, a_2, a_1) **yielding** $(b_3, b_2, b_1) = (a_3 \cdot f_3(a_2, a_1), a_2 \cdot f_2(a_1), a_1 \cdot f_1)$**. The "trays" denote the applications of functions** f_1**,** f_2**, and** f_3 **according to hierarchical dependence. Note that the applications of these functions happen simultaneously; their arguments are the previous states of other components, therefore there is no need to wait for the other components to calculate their new states. Right: (Without loss of generality), the new state** (b_1, b_2, b_3) **is the output of the automaton. Projection onto to initial coordinates is a homomorphism.**

	Natural Numbers	**Finite Automata**
Building Blocks	Primes	Flip-flop Automaton Permutation Automata
Composition	Multiplication	Wreath Product
Precision	Equality	Division, Emulation
Uniqueness	Unique	Different Decompositions

Though it is a fundamental theory of computer science, for forty years there was no computational implementation for the hierarchical decomposition of automata. Several different proofs were devised by mathematicians and computer scientists [77, 78, 110, 154, 155, 60, 50, 71, 43, 68, 51, 52, 124, 125], but there were no considerations for working computational implementations until 2005, when the first working software package for hierarchical decomposition of finite transformation semigroups was finally developed (the software development continues as a `GAP`[57] package [48]). The algebraic hierarchical decomposition of automata has immense possibilities for real world applications. These suggested applications were described several times [123, 109, 108, 111], but without a computational tool, they remained interesting promises only. The situation has finally changed, and though there are many issues to be solved regarding the efficiency and scalability of the algorithms, the research on applications has already begun. The capability of the developed software tools went far beyond the possibilities of pen and paper calculations. Now the powerful applications implied by the fundamental nature of the theory are within reach of a few years of research. As a proof of concept, initial attempts have been made in applying algebraic hierarchical decompositions in biological systems [44, 46, 45].

2.2 Structure and Function

With the background provided thus far, we can now explain more precisely one of the core mathematical problems that we have identified and on which we are working in both the OPAALS and BIONETS projects. In its simplest form an automaton can be defined as follows:

> Automaton = [set of states] + [semigroup of transformations operating on this set]

It turns out that a dynamical system can be defined in a way that resembles this definition. A more extensive explanation or motivation for why we believe it is so important to study non-linear dynamical systems has already been provided in DBE deliverable D18.4 [33]. Briefly, given that cellular processes are strongly coupled with each other, their mathematical models derived from chemical rate equations are non-linear systems of coupled differential equations. Along with the arguments already provided in the Introduction, we feel that in order to reach a successful formalisation of self-organising biologically-inspired behaviour in software we need to understand the mathematics of self-organisation in biological systems – which is non-linear. Thus, the research we are pursuing in OPAALS and BIONETS aims to study this problem by combining the discrete and continuous perspectives on non-linear dynamical systems, and to relate the results of these two different mathematical analyses. The bulk of this work will happen in Year 4 of the project, so here we give a high-level overview.

As discussed more fully in D2.2.9 [36], a (continuous) dynamical system can be seen as a smooth manifold and a set of continuous transformations operating on this manifold. A 'manifold' is nothing more than a continuously parameterisable space for each of whose points a tangent space can be defined that looks locally like Eucledian $\mathbb{R}^n$ for some dimension n; for example, the surface of a sphere, or the surface of a torus, is a 2-dimensional manifold for which the tangent space at every point is $\mathbb{R}^2$ or a plane. The manifold of a dynamical system can be defined in different ways. Whereas in D2.2.9 [36] we explain the whole story, in this more introductory discussion we can say that the most familiar description relies on a visualisation of the solution as a curve (1-D manifold) in the so-called 'phase space', which is the conceptual analogue of the state space of an automaton and whose coordinates are the dependent variables of the system and their first derivatives. If the system is characterised by some conservation law, such as conservation of energy for a frictionless system, the solution will be constrained to a subspace of phase space of correspondingly smaller dimensionality. For example, the solution curve of a 2-degree-of-freedom system for which energy and angular momentum are conserved will lie on a 2-D surface embedded in a 4-D phase space. If the motion in both degrees of freedom is oscillatory, such 2-D manifold will look like a torus, or a distortion thereof.

When represented in this space, the solution is a curve parameterised by time, meaning that time increases along it but is not otherwise 'visible' since it is not an explicit dimension of phase space. In the case of a chaotic system the manifold is not smooth, it is a fractal. For example, it could have fractional dimension somewhere between 2 (a surface) and 3 (a solid), and the solution trajectory is therefore correspondingly messy. The only methods we have to tackle and solve some non-linear dynamical systems is through the method of Lie groups. Lie groups are continuous groups of transformations that operate on objects defined on the manifolds. In particular, because they leave the functional form of the ODEs invariant, they map solutions into other solutions, both of which lie in the same manifold. So we can write a similar 'equation':

> Dynamical system = [manifold in phase space] + [Lie group of continuous transformations operating on this manifold]

With this we notice that dynamical systems share a similar algebraic structure to automata, but that, in addition, they also have a continuous analytic structure [116]. This integration of algebra and analysis was the great achievement of Sophus Lie in developing his theory.

Now the surprising thing is that, whereas in an automaton the transformations are aligned with the time axis, in the case of dynamical systems the time evolution of the system along the solution curve is not actually the focus of attention of Lie groups-based analytical methods.Rather, the transformations of interest are 'perpendicular' to the direction in which time increases, i.e. they are transverse to the solution curve itself. This is because whereas the time dimension is measured along a solution curve, the transformations of a Lie group map a solution to an adjacent solution. Thus each point on the solution curve is mapped to the corresponding point on a nearby solution curve. The arrow that joins these two points is transverse to both. To align the mathematical structure of the two frameworks in the context of automata theory, nothing stops us from imposing a similar additional set of transformations upon the computation trajectories in order to find other, equivalent algorithms. For example, exploiting the symmetries of an automaton is in fact a well-known technique in model checking formalisms [30].

Thus, this approach opens the possibility of relating the static symmetries associated with the semigroup structure in the direction of computation with the dynamic symmetries associated with the geometrical structure of the manifold that hosts the solution trajectories in a different direction, even though in the case of an automaton such a manifold would not even be continuous, let alone smooth. Since every point on the solution manifold of a dynamical system represents an acceptable state of that system, it is analogous to the state of an automaton. Whereas the manifold of a dynamical system is embedded in a metric space that maps directly to physical space, or to a space that can easily be related to physical variables such as chemical concentrations, the state space of an automaton is abstract. However, it still has a topological structure dictated by the possible transformations and algorithms linking the various states. Therefore, although it seems difficult to relate the two kinds of algebraic structures associated with automata and with continuous dynamical systems because they speak very different mathematical 'languages', the two representations can be obtained from the *same* biological system. Hence it seems plausible to claim that at some abstract level these different kinds of mathematical structures ought to be related. In addition, Lie groups methods have also been applied to discrete dynamical systems [94].[11]

The fact that the first kind of symmetries is time-independent whereas the second is associated with dynamical behaviour suggests that the relationship between these two properties of cellular pathways may be a more abstract analogue of the fundamental relationship between structure and function in biology. Physical structure both constrains and facilitates – by separating and by bringing together. For example, the internal folds, cristae, of the mitochondrion provide a structural means whereby functional elements are brought into proximity with each other. This interdependence between Structure and Function has enabled evolutionary processes to prioritise different DNA structures by selecting their corresponding phenotypical functions and behaviour. The relevance of this relationship to all types of engineering and applied thinking motivates us to investigate how these two kinds of mathematical structure are related. The benefit of identifying and formalising such a relationship would be the ability to specify desired behavioural properties and derive the corresponding structural properties.

In order to be able to specify the desired behavioural properties of biologically-inspired software systems, we need to be able to identify the primitives of a specification language capable of expressing them. As we mentioned in the Introduction, category theory can help in this direction,

[11] See also http://personal.maths.surrey.ac.uk/st/P.Hydon/sym.htm.

in two steps. First, we wish to use category theory to develop a unifying mathematical framework for these phenomena, including the static and dynamic symmetries introduced above, through the natural duality between algebras and coalgebras. Second, the already well-established correspondence between coalgebras and temporal logics opens the possibility of developing 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. The groundwork for this activity is being laid in the BIONETS project, and will hopefully continue in the new project.

In the next chapter we summarise the analysis of several dynamical systems and of their effectiveness at capturing biological behaviour, in particular the p53-mdm2 regulatory pathway. Whereas the Lie group analysis of this pathway is being pursued in BIONETS, in the next OPAALS deliverable (D1.4) we will analyse the algebraic structure of the automaton derived from this pathway, hoping to reach some useful conclusions and insights as to the relationship between these two mathematical structures as they apply to the same biological system.

3 Numerical and experimental analysis of p53-mdm2 regulatory pathway

Oscillating systems in living organisms are mathematically tractable and, through analysis and understanding of their behaviour, might provide useful insights into the design of computer systems. It therefore seemed appropriate for us, as biologists within the OPAALS project, to look critically at oscillating systems in biology to see whether or not they might be used to inform the computer science aspects of the project.

3.1 Oscillation in cellular systems

Since autopoiesis is one of our key areas of interest, and since the archetypical autopoietic entity is the cell, we have looked systematically at cellular oscillating systems – a subject that has recently been reviewed in some detail [96]. At the cellular level, an oscillating system can be defined as one whose attributes rise and fall in a regular fashion over a sustained period. This definition is flexible with respect to time: the key requirement is that the sustained period is of relatively long in duration when compared to the period of the oscillation itself. Such systems have interesting implications for computer science: each oscillation could be regarded as the tick of a digital clock, a pulse that might be counted.

Probably the most obvious role for self-sustaining cellular oscillators in biology is as timekeepers, controlling events either on a shorter time frame, such as circadian rhythms [143], or on a longer time frame, such as embryonic development [10]. However cellular oscillators are not limited to timekeeping. Experimental as well as theoretical work has demonstrated that cellular oscillators might be capable of performing a wide variety of important functions [61]. These include: decisions concerning the fate of a cell [2]; control of calcium-dependent signalling pathways; facilitating cellular responses to changes in environment; and regulation of cellular energy production [11].

3.1.1 P53/mdm2 as an oscillating(?) system

We have chosen to investigate the behaviour of the p53/mdm2 system. Mainly because it is a critical element in determining the fate of a cell [84] – with obvious analogies to the success (or failure) of an enterprise – and because it is one with which we are somewhat familiar. We also believed, on the basis of previously published work [5, 82, 83, 59], that oscillatory behaviour was characteristic of the system and that, by further exploring these oscillations, we might usefully contribute to the mathematical and computer science aspects of the OPAALS project. P53 is a protein that is synthesised in response to stress or damage to a cell and has variable effects on cellular behaviour: cells may rest and repair the damage; they may die via a process of programmed cell death ("apoptosis"); they may continue to behave as normal but, later, prove incapable of division (premature replicative senescence). P53 is negatively regulated by mdm2 – mdm2 targets p53 for degradation and elimination via a process termed ubiquitination (conceptually this can be regarded as dumping p53 down the garbage chute). Mdm2 is named because its presence, in the mouse, is associated with double minute chromosomes: Mouse Double Minute. It is also called hdm2 – by convention, this term is used when human tissues, or cell lines of human origin, are involved.

As a first step we sought to justify some of the assumptions underlying the published work on sustained oscillations in the p53/mdm2 system. The single-cell experiment [83], upon which an elaborate mathematical superstructure has been built [82, 59, 144, 1, 6, 119, 115, 29, 25, 93], used fluorescent labelling to monitor simultaneous changes in expression of p53 and mdm2 following irradiation. A key assumption in any such experimental system is that the labelling does not

have any effect upon the function of the protein. Our own experimental work shows that, somewhat unexpectedly, the labelling of mdm2 renders it functionally inactive. This raises the possibility that the oscillations observed in the single-cell fluorescence experiment might be an artefact caused by the labelling procedure.

3.1.2 Experimental data

The Western blots shown in Figure 9 illustrate this point. A Western blot is a gel-based method for assessing the amounts of specific proteins present in an experimental system. The darker the band, the more protein is present. In this experiment a series of cell lines were irradiated with X-Rays – a classical method for stressing cells and inducing p53 expression. In normal cells any rise in p53 will be followed by a rise in its negative regulator mdm2(hdm2). We see this clearly in the pair of gels in the top half of Figure 9. In the absence of hdm2 there is abundant p53 (left-hand rectangle), in the presence of hdm2 (lower gel) there is partial suppression of p53 expression (right-hand rectangle). The hdm2 labelled with yellow fluorescent protein (hdm2-YFP) is not capable of suppressing p53 expression: in the lower pair of gels there is, if anything, more p53 expression within the blots enclosed by the right-hand rectangle. This suggests strongly that labelling hdm2 with yellow fluorescent protein might impair its ability to down-regulate p53 expression.

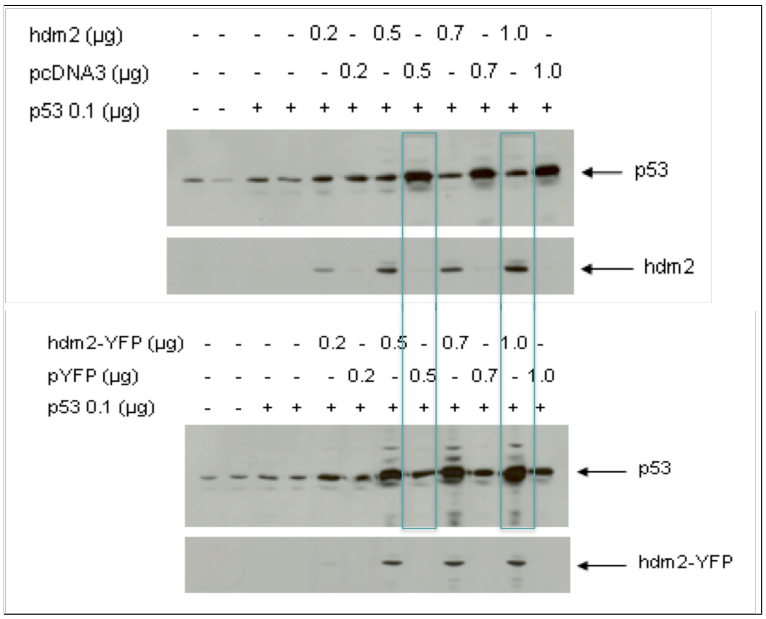

Figure 9: **In the lower gel there is abundant p53 expression (right-hand blue box) despite presence of mdm2. If the labelled mdm2 were functional we would expect to see inhibition of p53 – if anything, p53 is induced.**

It is possible that the sustained oscillations that have been observed in single-cell experiments may be due to artefacts induced by the abnormalities in mdm2 function introduced by the labelling procedure. This leaves us somewhat sceptical concerning the ability of regular sustained oscillations in the p53/mdm2 system, within individual cells, to act as a digital counting mechanism for determining the fate of a cell. Nevertheless we have good evidence from our own work and that of others [88] that the p53/mdm2 system does oscillate – but that these oscillations are damped. There are several possible explanations for such damped oscillations – the most parsimonious is that p53 and MDM2 operate in a simple feedback loop but with a discrete time delay [106]. Another explanation – less favoured by us for the reasons given – is that the damping at the aggregate level is caused by the presence of a mixed population of cells, some of which oscillate over a protracted period and some of which do not oscillate at all. A

relative increase in non-oscillating cells would then explain the apparent damping.

Having set out to use sustained oscillations in the p53/mdm2 system as the basis of our mathematical models we were forced into an unexpected conclusion: close scrutiny of some so-called "oscillating systems" shows that the biological observations upon which they are based may be flawed and, furthermore, the models used to demonstrate oscillations may only show oscillatory behaviour under certain restricted and somewhat artificial conditions. The literature at the interface between mathematics and biology is not as robust as a superficial reading would lead us to believe. Perhaps this is a reflection of a more general problem in interdisciplinary research: it is difficult to achieve a balance of expertise. Mathematical sophistication can be undermined by biological naïveté, and vice versa.

Our disappointment with our initial exploration of oscillating systems raises a fairly fundamental question: when we are using natural science to influence mathematics, algebra and computer science, are we seeking to inspire using a model that accurately reflects the natural world, a robust template; or are we simply generating possibilities, initiating chains of thought and encouraging creative approaches? If our aim is the latter then the extent to which our models fit the biological observations is less than critical – indeed could be entirely unimportant; if our aim is the former, then the goodness-of-fit is crucial. At this point it seems entirely reasonable to leave the question rhetorical, unanswered.

Despite our concerns about *sustained* oscillations in the p53/mdm2 system we are confident that the p53/mdm2 system exhibits *damped* oscillations in cell populations, as shown in Figure 10. This information has been used to inform the mathematical models that are discussed in the following section.

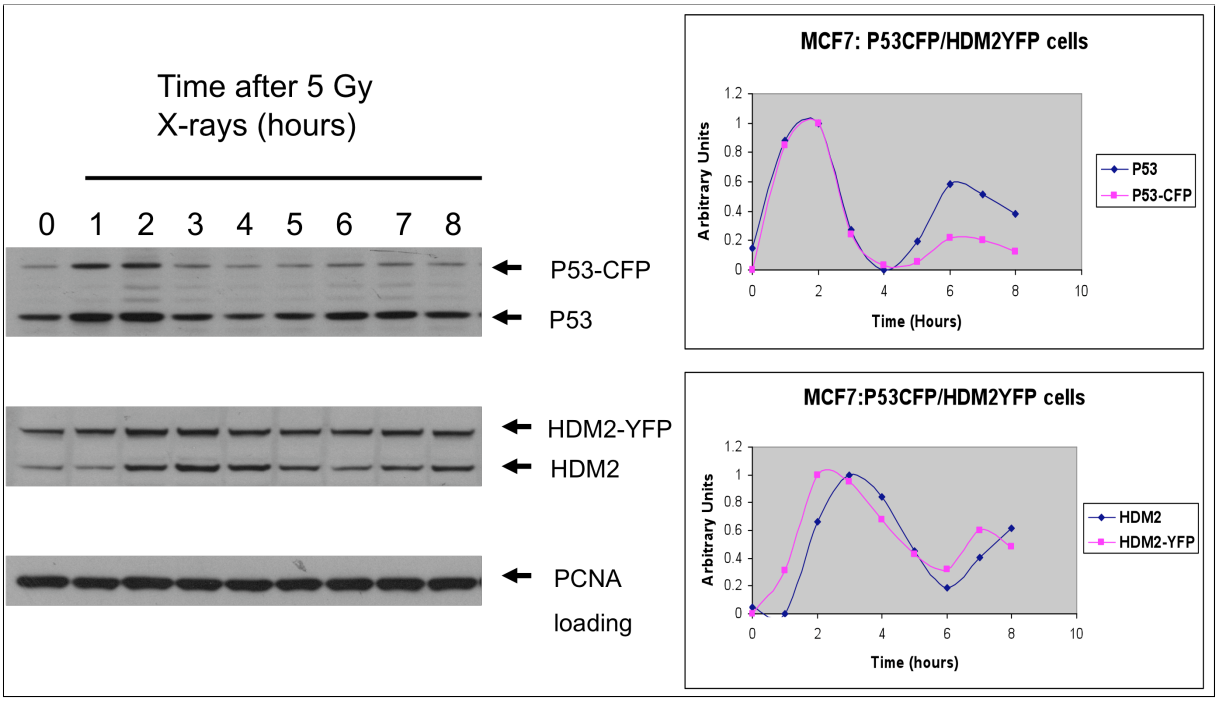

Figure 10: **Levels of p53 and mdm2 after irradiation of cells in suspension. There is a prominent peak for p53 at 2 hours post exposure followed by a second (lower) peak at 6 hours; for mdm2 the peaks are at 3 hours and 7 hours. Findings consistent with a delay between the rise in p53 and the induction of mdm2. Each observation is based on a pooled estimate from $> 100,000$ cells; the error bars (s.e.m.) are very narrow and are obscured by the data points themselves.**

3.2 P53 Modelling

During the cell cycle, key proteins undergo cyclic synthesis and degradation, the associated changes in their levels triggering progression through the cell-cycle phases. In recent years, similar oscillatory behaviour has been shown in several other regulatory networks. The NF-κB transcription factor, for instance, presents sustained nucleo-cytoplasmic fluctuations after stimulation with TNFα [4, 76, 112]. Similarly, the levels of Hes1 protein and mRNA oscillate in cell culture upon serum treatment [9, 69]. In the mouse segmentation clock, not only the levels of Hes1 but also of components of the Wnt and Notch pathways have been shown to oscillate *in vivo* [32, 150]. Given the biomedical importance of these pathways, the molecular basis and biological implications of their oscillations have attracted extensive attention from experimentalists and theoreticians alike. Although the signalling pathways above differ in their components and biological outcomes, the oscillations are believed to share a common underlying mechanism: a negative feedback loop (NFL) combined with a transcriptional delay [106, 117, 142]. Notably, such a mechanism is present in the p53–mdm2 network (Figure 11), as p53 promotes the synthesis of its main negative regulator, mdm2 (Figure 12). The possibility that p53 levels oscillate in response to stress has therefore been explored both experimentally and theoretically.

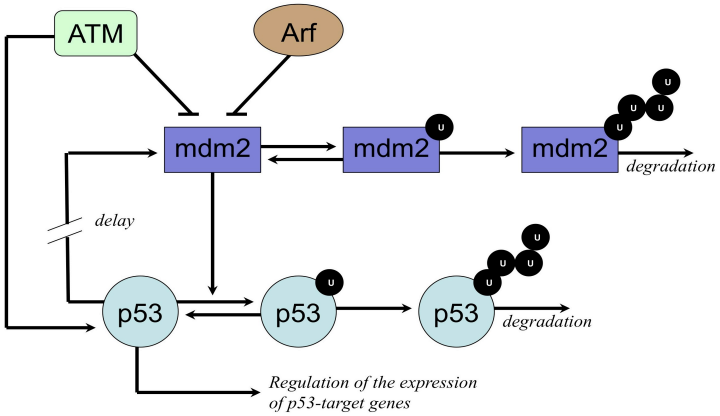

Figure 11: **Simple schematic of the p53 network. The p53 tumour suppressor – the *guardian of the genome* – acts as a transcription factor, regulating the expression of over a hundred target genes [53, 84, 147]. The mdm2 protein, a RING finger-dependent ubiquitin protein ligase, binds to p53 and targets it for proteasomal degradation [66]. Both p53 and mdm2 are highly regulated proteins [79].**

The p53 pathway has been subject to extensive modelling efforts, the first model being published by Bar-Or and co-workers in 2000. The existing models can be classified into two categories based on their purpose. On the one hand, relatively simple models describing the role of p53 in cancer have been developed as part of multiscale models for the disease [23, 42, 89, 126]. Ribba *et al.* [126], for instance, used p53 as a switch adopting values 0 and 1 in the absence and presence of DNA damage, respectively, whereas in the cellular automaton model developed by Alarcón and co-workers [23] cells possess different automaton features depending on whether their p53 status is wild-type or mutant. On the other hand, various kinetic models have been built to investigate whether the system can give rise to damped oscillations in p53 and mdm2 in cell populations and undamped pulses in individual cells [1, 6, 15, 29, 59, 83, 93, 102, 106, 88, 119, 120, 149, 157]. The vast majority assume that a cell consists of a single compartment in which the proteins of interest are abundant and evenly distributed; hence they undertake a continuum, deterministic approach, using either ordinary or delay differential equations (ODEs or DDEs) to describe the changes in protein levels. Alternative modelling approaches include stochastic, spatial and multiple compartment models. Proctor *et al* [119], for instance, developed stochastic models to account

for the intercellular variation in the number of macromolecules. In contrast, Ciliberto *et al* [29] formulated a deterministic, compartmental ODE model that distinguishes between nuclear and cytoplasmic mdm2. Finally, Gordon *et al* [63] relaxed the intracellular homogeneity assumption further and used partial differential equations (PDEs) to describe intracellular protein density patterns.

To prevent the negative feedback loop linking p53 and mdm2 (Figure 12) from immediately inhibiting itself and, in particular, to enable oscillations to occur, a delay in the negative feedback loop is required [141]. Such a delay can be modelled in different ways. In some cases the delay has been included implicitly in mathematical models – Lev Bar-Or *et al* [88], for instance, included an additional unknown component in the p53–mdm2 pathway, whereas Ciliberto *et al* [29] combined positive and negative feedbacks. The alternative is to incorporate time delays explicitly. Srividya *et al* [139] have shown that discrete delay terms can help reduce the number of variables and parameters required to describe a molecular system by replacing one or more intermediate reactions. Concerning spatial effects, [106] demonstrated that the waiting times associated with transcription and translation can be fused into a single time delay without altering the dynamical properties of the system. Hence, several models [6, 63, 106, 115] calculate the rate of mdm2 synthesis at time t as a function of the amount of p53 present in the system at time $t - \tau$.

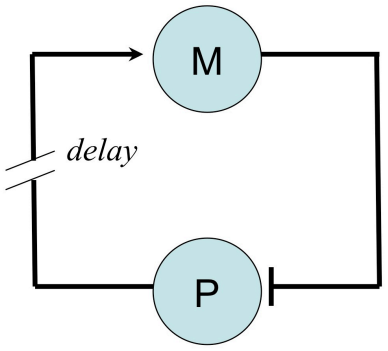

Figure 12: **Negative feedback loop with transcriptional delay linking p53 and mdm2.**

Below we discuss in detail three existing theoretical approaches to the p53 pathway (i.e. [88, 106, 29]), while in the next section we present a new model. To facilitate comparison of the models, we have unified the notation as follows: $[P(t)]$ and $[M(t)]$ are the total intracellular concentrations of p53 and mdm2 at time t; $\mathcal{S}(t)$ represents a transient stress stimulus (e.g. DNA damage); $s_\star$ are *de novo* synthesis rates; $k_\star$ are production rates (e.g. phosphorylation); $j_\star$ are reverse reactions (e.g. desphosphorylation); $d_\star$ are degradation rates; $\sigma_\star$ are transport rates; $K_\star$ are saturation coefficients and $c_\star$ are additional constants.

3.2.1 Model I: Lev Bar-Or et al (2000)

The first model for the p53-mdm2 network, formulated by Lev Bar-Or *et al* [88], describes the interaction between p53 and mdm2. The time–lag between p53 activation and p53-mediated induction of mdm2 synthesis is modelled by the presence of a hypothetical intermediary, X. The proposed kinetic equations are [88]:

$$\begin{aligned} \mathrm{d}[P]/\mathrm{d}t &= s_p - \left(d_p + \widehat{d}_{pm}(t)[M]\right)[P], \\ \mathrm{d}[M]/\mathrm{d}t &= s_{m0} + \frac{s_{mx}[X]^n}{[X]^n + K_{mx}^n} - d_m[M], \\ \mathrm{d}[X]/\mathrm{d}t &= \frac{s_s \mathcal{S}(t)[P]}{1 + c_x[M][P]} - d_x[X], \end{aligned} \tag{1}$$

with initial conditions $[P(0)] = P_0$, $[M(0)] = M_0$, and $[X(0)] = 0$. The stress stimulus $\mathcal{S}$ is assumed to influence the behaviour of the system in two ways, namely by promoting p53's transcriptional activity and by downregulating mdm2-mediated degradation of p53. The former mechanism is modelled by making the synthesis of X an increasing function of $\mathcal{S}$, while the latter is incorporated by making $\widehat{d}_{pm}$ a decreasing function of $\mathcal{S}$. Numerical analyses and simulations revealed that, for certain parameter values, the system can show damped oscillations in which mdm2 and p53 levels peak out of phase.

3.2.2 Model II: Monk (2003)

In contrast to [88], the model proposed by Monk [106] not ony characterises the dynamics of the mdm2 and p53 proteins, but also the changes in the level of mdm2 mRNA. Morever, it explicitly accounts for a transcriptional delay as follows:

$$\begin{aligned} \mathrm{d}[P]/\mathrm{d}t &= s_p - \left(d_{p0} + \frac{d_{pm2}[M]^2}{[M]^2 + K_{pm}^2}\right)[P], \\ \mathrm{d}[R_M]/\mathrm{d}t &= s_{rm0} + \frac{s_{rm1}[P(t-\tau)]^n}{[P(t-\tau)]^n + K_{rm}^n} - d_{rm}[R_M], \\ \mathrm{d}[M]/\mathrm{d}t &= s_{rt}[R_M] - d_m[M], \end{aligned} \tag{2}$$

where both p53's transcriptional activity and mdm2's ubiquitin-ligase activity are assumed to be saturating functions. The system demonstrates oscillatory behaviour for certain parameter values, the period of the oscillations depending on the transcriptional delay and the protein and mRNA half-lives.

3.2.3 Model III: Ciliberto et al (2005)

Compared with the approaches above, Ciliberto *et al* [29] incorporated substantially more biological detail in their model, to arrive at a relatively more sophisticated description of the p53–mdm2 network that accounts for two subcellular compartments, namely the nucleus and the cytoplasm. As it is assumed that mdm2 has to be phosphorylated in order to enter the nucleus, the model includes three molecular forms of mdm2: nuclear mdm2, and both unphosphorylated and phosphorylated cytoplasmic mdm2. Moreover, ubiquitination is modelled as a multistep process (Figure 11), involving three molecular forms of p53 (i.e., non-ubiquitinated, mono-ubiquitinated and poly-ubiquitinated protein). The dynamics of the six molecular components is expressed by:

$$\begin{aligned} \mathrm{d}[P]/\mathrm{d}t &= s_p - d_p[P] - d_{pu}[P_{UU}], \\ \mathrm{d}[P_U]/\mathrm{d}t &= k_u([P] - [P_U] - [P_{UU}])[M_N] + j_u[P_{UU}] - (j_u + d_{p1} + k_u[M_N])[P_U], \\ \mathrm{d}[P_{UU}]/\mathrm{d}t &= k_u[M_N][P_U] - (j_u + d_{pu} + d_{p1})[P_{UU}], \\ \mathrm{d}[M_N]/\mathrm{d}t &= (\sigma_{cn}[M_{PC}] - \sigma_{nc}[M_N])v - \left(d_{m0} + \frac{d_{m1}\mathcal{S}(t)}{\mathcal{S}(t) + K_m}\right)[M_N], \\ \mathrm{d}[M_C]/\mathrm{d}t &= s_{m0} + \frac{s_{m1}[P]^n}{[P]^n + K_{pm}^n} + k_{pc}[M_{PC}] - \left(d_{m0} + \frac{k_{pc}}{[P] + K_{pc}}\right)[M_C], \\ \mathrm{d}[M_{PC}]/\mathrm{d}t &= \frac{k_{pc}[M_C]}{[P] + K_{pc}} + \sigma_{nc}[M_N] - (j_{pc} + \sigma_{cn} + d_{m0})[M_{PC}], \end{aligned} \tag{3}$$

The state variables above represent the concentrations of total p53, P; monoubiquitinated p53, P_U; poly-ubiquitinated p53, P_{UU}; nuclear mdm2, M_N; unphosphorylated cytoplasmic mdm2, M_C; and phosphorylated cytoplasmic mdm2, M_{PC}. The parameter v denotes the nuclear-cytoplasmic volume ratio. Unlike in the previous models, the value of the stress function $\mathcal{S}$ depends on the level of p53, as it is assumed that p53 plays a role in DNA repair:

$$\mathrm{d}\mathcal{S}/\mathrm{d}t = \widehat{k}_s(I(t)) - \frac{d_s\mathcal{S}(t)[P]}{\mathcal{S}(t) + K_s},$$

where $\widehat{k}_s$ is the DNA damage production rate as a function of time and dose of irradiation. Based on numerical analyses and simulations, the authors suggest that the discrete pulses observed by Lahav *et al* [83] might be the result of a combination of positive and negative feedbacks. In the model,

> the positive feedback originates from two opposing negative effects: nuclear mdm2 induces p53 degradation, while p53 inhibits nuclear entry of mdm2, by inhibiting phosphorylation of mdm2 in the cytoplasm. The negative feedback loop is the well-known fact that p53 induces the synthesis of mdm2 [29].

Eqns (3) also predict that the level of mdm2 can increase in the absence of mdm2 synthesis, which indicates a mass balance problem. This can be solved by rewriting the expressions for the changes in $[M_N]$ and $[M_{PC}]$:

$$\begin{aligned} \mathrm{d}[M_N]/\mathrm{d}t &= v\sigma_{cn}[M_{PC}] - \sigma_{nc}[M_N] - \left(d_{m0} + \frac{d_{m1}\mathcal{S}(t)}{\mathcal{S}(t)+K_m}\right)[M_N], \\ \mathrm{d}[M_{PC}]/\mathrm{d}t &= \frac{k_{pc}[M_C]}{[P]+K_{pc}} + \frac{\sigma_{nc}[M_N]}{v} - (j_{pc}+\sigma_{cn}+d_{m0})[M_{PC}], \end{aligned} \tag{4}$$

An analogous modification has been introduced in [1], which presents a four ODE model inspired by [29]. The new system of ODEs based on (4) does not oscillate for the parameter values used in [29].

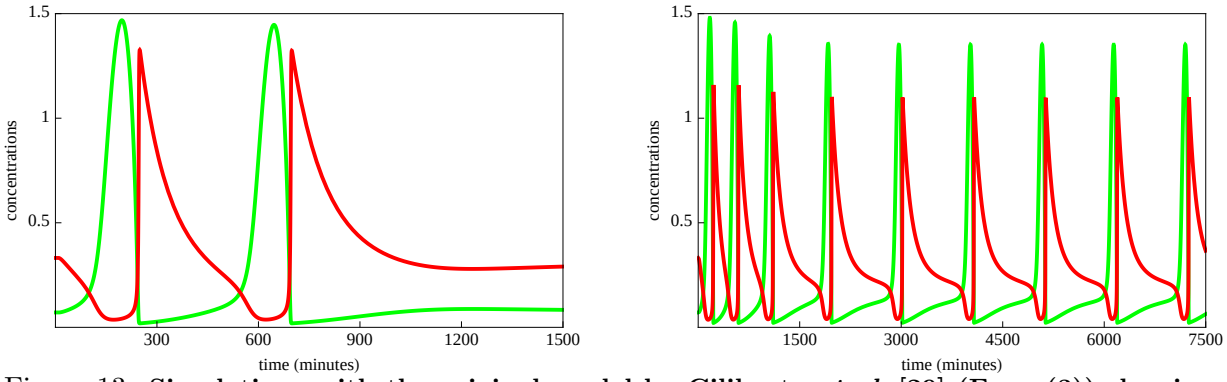

Figure 13: **Simulations with the original model by Ciliberto *et al.* [29] (Eqns (3)) showing two pulses (*left*) and sustained oscillations (*right*) in the levels of total p53, $[P]$ (green), and nuclear mdm2, $[M_N]$ (red). For the parameter values, see the original paper [29]. In the left and right panels, the value of the cytoplasm:nucleus ratio, v, is 15 and 11, respectively.**

3.3 Formulation of the New Model

To show that there is yet another possible biological explanation for the occurrence of oscillations in p53 levels in response to stress, we will now introduce a new model for the p53 pathway. The structure of our simple network is shown in Figure 14. It describes the interactions between four molecular components: P_I, the p53 tumour suppressor, M, p53's main negative regulator, mdm2, C, the p53–mdm2 complex and, P_A, an *active* form of p53 that is resistant against mdm2-mediated degradation. The model accounts for the following phenomena: (1) basal p53 synthesis; (2) basal (i.e. mdm2-independent) p53 degradation; (3) mdm2 synthesis; (4) basal mdm2 degradation; (5) p53–mdm2 complex assembly; (6) p53–mdm2 complex dissociation; (7) mdm2-mediated p53 ubiquitination and subsequent elimination; (8) stress-induced p53 activation; (9) p53 inactivation; and (10) basal degradation of active p53. According to the

reaction scheme shown in Figure 14, the changes in the concentrations of the four molecular components are given by:

$$\begin{aligned} \mathrm{d}[P_I]/\mathrm{d}t &= r_1 - r_2 - r_5 + r_6 - r_8 + r_9, \\ \mathrm{d}[M]/\mathrm{d}t &= r_3 - r_4 - r_5 + r_6 + r_7, \\ \mathrm{d}[C]/\mathrm{d}t &= r_5 - r_6 - r_7, \\ \mathrm{d}[P_A]/\mathrm{d}t &= r_8 - r_9 - r_{10}. \end{aligned} \tag{5}$$

where $r_i(t)$, for $i = 1,\ldots,10$, is the rate of reaction i at time t and $[X]$ denotes the concentration of molecular component X, for $X = P_I$, P_A M, and C.

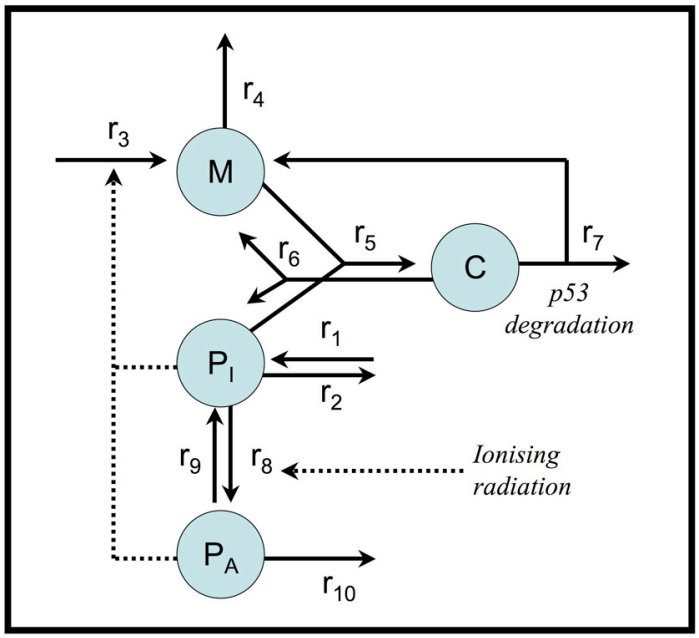

Figure 14: **Schematic of the original p53–mdm2 interaction model.** M **= mdm2 protein;** P_I **=** ***inactive*** **p53 protein,** C **= mdm2–p53 complex;** P_A **=** ***active*** **p53 protein.**

3.3.1 Simplifying assumptions

For simplicity, we assume that the basal p53 synthesis rate, r_1, remains constant in time and that basal degradation rates, r_2, r_4 and r_{10}, are proportional to the corresponding substrate concentrations. The mdm2 protein, a RING finger-dependent ubiquitin protein ligase, is known to bind to p53 and target it for proteasomal degradation [66]. In the model, the binding of mdm2 to p53 is assumed to be reversible with assembly and dissociation rates r_5 and r_6, respectively. Furthermore, experimental evidence has shown that mdm2's downregulation of p53 is inhibited under DNA damage [137]. We have incorporated this observation into the model by assuming that ionising radiation induces the phosphorylation of p53, which prevents mdm2 binding. That is, r_8 is an increasing function of the level of radiation exposure. Finally, reaction 3 represents the negative feedback loop in which p53 transactivates expression of the *MDM2* gene [153]. As we expect this expression rate to reach a maximum value when there is negligible delay between the binding of two successive p53 molecules to the *MDM2* promoter region, we assume that r_3 is a saturating function of the total level of p53. Given the assumptions above, the reaction rates

can be calculated as follows:

$$
\begin{aligned}
r_1(t) &= s_p, \\
r_2(t) &= d_p[P_I(t)], \\
r_3(t) &= s_{m0} + \frac{s_{m1}[P_I(t)] + s_{m2}[P_A(t)]}{[P_I(t)] + [P_A(t)] + K_m}, \\
r_4(t) &= d_m[M(t)], \\
r_5(t) &= k_c[P_I(t)][M(t)], \\
r_6(t) &= j_c[C(t)], \\
r_7(t) &= k_u[C(t)], \\
r_8(t) &= k_a\mathcal{S}(t)[P_I(t)], \\
r_9(t) &= j_a[P_A(t)], \\
r_{10}(t) &= d_p[P_A(t)].
\end{aligned}
$$

Substitution of the expressions above into Eqns (5) yields:

$$
\begin{aligned}
\mathrm{d}[P_I]/\mathrm{d}t &= s_p + j_a[P_A] - (d_p + k_a\mathcal{S}(t))[P_I] - k_c[P_I][M], \\
\mathrm{d}[M]/\mathrm{d}t &= s_{m0} + \frac{s_{m1}[P_I] + s_{m2}[P_A]}{[P_I] + [P_A] + K_m} + k_u[C] - (d_m + k_c[P_I])[M], \\
\mathrm{d}[C]/\mathrm{d}t &= k_c[P_I][M] - (j_c + k_u)[C], \\
\mathrm{d}[P_A]/\mathrm{d}t &= k_a\mathcal{S}(t)[P_I] - (j_a + d_p)[P_A].
\end{aligned}
$$

3.3.2 Model IV: dimensionless equations

The state variables in Figure 14 can be scaled as

$P_i = k_u[P_I]/s_p$, $M = k_u[M]/s_p$, $C = k_u[C]/s_p$, $P_a = k_u[P_A]/s_p$ and $\tau = k_u t$.

In terms of these new variables, the model equations reduce to:

$$
\begin{aligned}
\mathrm{d}P_i/\mathrm{d}\tau &= 1 + \beta_a P_a - (\beta_p + \alpha_a\mathcal{S}(\tau))P_i - \alpha_c P_i M, && (6) \\
\mathrm{d}M/\mathrm{d}\tau &= \alpha_{m0} + \frac{\alpha_{m1}P_i + \alpha_{m2}P_a}{P_i + P_a + \kappa_m} + C - (\beta_m + \alpha_c P_i)M, && (7) \\
\mathrm{d}C/\mathrm{d}\tau &= \alpha_c P_i M - (1 + \beta_c)C, && (8) \\
\mathrm{d}P_a/\mathrm{d}\tau &= \alpha_a\mathcal{S}(\tau)P_i - (\beta_a + \beta_p)P_a. && (9)
\end{aligned}
$$

where the dimensionless parameters are defined as: $\alpha_a = k_a/k_u$, $\beta_a = j_a/k_u$, $\alpha_c = k_c s_p/k_u^2$, $\beta_c = j_c/k_u$, $\alpha_{m0} = s_{m0}/s_p$, $\alpha_{m1} = s_{m1}/s_p$, $\alpha_{m2} = s_{m2}/s_p$, $\beta_m = d_m/k_u$, $\beta_p = d_p/k_u$ and $\kappa_m = k_u K_m/s_p$. In the simulations depicted in Figure 15, two modes of stress have been considered, namely a discrete pulse insult at time zero and long-term exposure to a constant stressful stimulus. The former is expressed as:

$$
\mathcal{S}(t) = \mathrm{e}^{-c_s t} \qquad \text{and} \qquad \mathcal{S}(\tau) = \mathrm{e}^{-\gamma\tau}, \tag{10}
$$

in dimensional and dimensionless form, respectively (the dimensionless stress coefficient is $\gamma = c_s/k_u$). In contrast, the latter is modelled as:

$$
\begin{aligned}
\mathcal{S}(\tau) &= \mathcal{S}_0 && \text{for} \quad 0 \le \tau \le \tau_i && (11) \\
&= \mathcal{S}_0\mathrm{e}^{-\gamma(\tau - \tau_i)} && \text{for} \quad \tau > \tau_i.
\end{aligned}
$$

Notably, the response of the system to the two kinds of insult is very different. Exposure to a single, pulse insult at time zero results in damped oscillations in the levels of p53 and mdm2 around their steady-state values in the absence of stress. In contrast, exposure to a long-term signal causes the system to move to a new steady-state, fluctuating transiently. This new steady-state has a higher p53 level, which depends on the strength of the signal. When the stimulus ends, the system returns to its original steady-state, displaying a second round of damped oscillations.

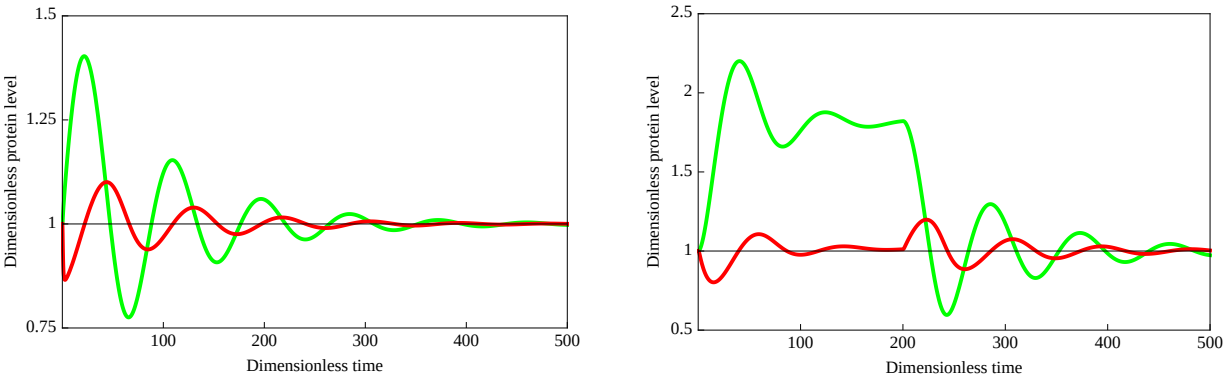

Figure 15: **Simulations with the dimensionless system (6–9) showing damped oscillations in the levels of total p53, $P + C$ (green), and total mdm2, $M + C$ (red). The values have been normalised with respect to the corresponding concentrations in the absence of stress (i.e., $P_0 + C_0$ and $M_0 + C_0$, respectively). The parameter values used in the simulations are provided in Table 2. One dimensionless time unit corresponds to 2.5 minutes. *Left panel*: response to a single pulse insult at time $\tau = 0$ (Eqn (10) with $\gamma = 2.5$). *Right panel*: response to a constant, long-term insult (Eqn (11) with $\gamma = 2.5$, $\tau_i = 200$ and $\mathcal{S}_0 = 0.05$).**

Table 2: Parameter values used in the model simulations shown in Figure 15

$[P_I(0)] = 12.25$ nM	$k_a = 20$ min^{-1}	$P_{i0} = 3.5$	$\alpha_a = 50$
$[P_A(0)] = 0$ nM	$k_c = 4$ min^{-1} nM^{-1}	$P_{a0} = 0$	$\alpha_c = 35$
$[C(0)] = 3.48$ nM	$k_u = 0.4$ min^{-1}	$C_0 = 0.993$	
$[M(0)] = 0.028$ nM	$j_a = 0.2$ min^{-1}	$M_0 = 8 \times 10^{-3}$	$\beta_a = 0.5$
$s_{m0} = 2 \times 10^{-3}$ nM/min	$j_c = 2 \times 10^{-3}$ min^{-1}	$\alpha_{m0} = 0.00143$	$\beta_c = 5 \times 10^{-3}$
$s_{m1} = 0.15$ nM/min	$s_p = 1.4$ nM/min	$\alpha_{m1} = 0.107$	
$s_{m2} = 0.2$ nM/min	$d_p = 2 \times 10^{-4}$ min^{-1}	$\alpha_{m2} = 0.143$	$\beta_p = 5 \times 10^{-4}$
$d_m = 0.4$ min^{-1}		$\beta_m = 1$	
$K_m = 100$ nM		$\kappa_m = 28.57$	

3.4 Discussion

In the above section we have visited four alternative mathematical descriptions of the p53 network. Their specific features are highlighted in Table 3. These models, which represent only a small sample of the models available in the literature, were chosen to illustrate the use of differential equations in systems biology and, in particular, to show how very different mechanisms can succeed in explaining the same data. The modelling efforts addressed here were motivated mainly by two experimental studies. First, Lev Bar-Or *et al* [88] exposed mouse fibroblasts NIH 3T3 cells expressing wild-type p53 and mdm2 to 5 Gy of irradiation and then measured the protein levels at several time points after exposure. Their Western Blots showed two peaks in the level of p53, each followed by a peak in mdm2 approximately one hour later. Second, Lahav and co-workers [83] created a cell line expressing p53 and mdm2 tagged with

fluorescent proteins to study the dynamics of these molecules in single cells. In response to γ-radiation, they observed digital pulses in both p53 and mdm2 levels (for a critical discussion on this approach, see Section 3.1). Unifying the two experimental observations, Ma *et al* [93] suggested that

> the damped oscillations previously observed in cell populations can be explained as the aggregate behaviour of single cells.

Comparing the four models above, models I [88] and II [106] are the most similar, as they both produce an oscillatory behaviour based on the mdm2–p53 NFL (Figure 12) alone and both account for an intermediary component linking of the level of p53 to the rate of mdm2 synthesis. The main difference is that model II also includes an explicit transcriptional delay, which implies a shift of the mathematical approach from ODEs to DDEs. According to model III [29], however, the mechanism underlying the oscillations is more complex and involves both negative and positive feedbacks. In response to ionising radiation, DNA damage

> increases abruptly as does [...] the rate of [...] degradation of $[M_N]$. As $[M_N]$ decreases, $[P]$ increases, which causes an initial drop in $[M_{PC}]$ and a steady increase in $[M_C]$. When a sufficient amount of mdm2 accumulates in the cytoplasm, it initiates a change of regime: phosphorylated mdm2 enters the nucleus, causing increased degradation of p53, which relieves the inhibition of mdm2 phosphorylation in the cytoplasm, allowing more mdm2 to enter the nucleus. The positive feedback loop causes the abrupt drop in $[P]$ and rise in $[M_N]$. The drop in $[P]$ cuts off the synthesis of $[M_C]$, and consequently $[M_C]$ and $[M_N]$ drop. The system is back to the original state, $[P]$ starts to accumulate again due to the low level of $[M_N]$ and a new oscillation starts [29].

While model III incorporates substantially more biologically-relevant information than models I and II, its main weaknesses are the strong dependence of the behaviour of the system on the relative size of the nucleus (see the simulations in Figure 3) and the mass balance problems discussed above (see corrected Eqns (4)).

Model III predicts a decrease in nuclear mdm2 in response to stress (Figure 13), and a subsequent reduction in mdm2-mediated p53 degradation. This is also the case in the context of model IV (Figure 15). The reason behind the drop in mdm2 is different, though. According to model III, the degradation rate of nuclear mdm2 is an increasing function of the level of DNA damage. In contrast, under model IV, p53-binding protects mdm2 from proteasomal degradation and, therefore, any decrease in $[P_I]$ translates naturally in an increased mdm2 elimination rate. Hence, when radiation promotes the transformation of P_I into P_A, mdm2 has less chance to bind to P_I and is thus at a higher risk of being rapidly degraded. This model prediction highlights the importance of accounting for the dynamics of the mdm2–p53 complex, as suggested by Proctor *et al* [119]

> since the regulation of p53 is dependent on its interaction with mdm2, we would expect that the oscillatory behaviour of the system would be strongly affected by the binding affinity of mdm2 to p53. Therefore any mechanistic model of the sytem should include the mdm2–p53 complex.

The reduction in mdm2 levels in response to stress in models III and IV plays a role equivalent to the transcriptional delay in models I and II: it enables the levels of mdm2 and p53 to peak out of phase, thereby allowing oscillations to occur.

Table 3: Summary of the mathematical models described in this chapter.

	FEATURES	**MECHANISM(S) FOR OSCILLATIONS**
MODEL I Lev Bar-Or *et al* [88]	• 3 ODEs + 1 expression for the stress function • Includes an unknown intermediary	• Implicit delay, in the form of an unknown intermediary, in the mdm2–p52 negative feedback loop
MODEL II Monk [106]	• 2 ODEs + 1 DDE • Includes mdm2 mRNA dynamics	• Implicit time delay, in the form of a known intermediary (i.e, mdm2 mRNA), in the mdm2-p53 negative feedback loop • Explicit time delay for gene transcription
MODEL III Ciliberto *et al* [29]	• 6 ODEs + 1 ODE for the stress function • Distinguishes between nuclear and cytoplasmic mdm2 • Accounts for 3 molecular forms of p53 and 2 forms of mdm2 • DNA damage enhances mdm2 degradation • p53 promotes DNA repair	• Combination of positive and negative feed back loops between mdm2 and p53 • Implicit time delay (mdm2 has to be phosphorylated and then shuttled into the nucleus before it can degrade p53)
MODEL IV New model	• 4 ODEs + 1 expression for the stress function • Characterises of the dynamics of the p53-mdm2 complex • Accounts for a mdm2-resistant form of p53	• mdm2-p53 negative feedback loop • p53 binding protects mdm2 from proteosomal degradation

The purpose of the experimental and numerical work is to gain better insight into the dynamics of the chosen cellular regulatory pathway and of the ODE models that represent it most faithfully. The next step is to derive the corresponding automaton and extract the algebraic structure of its semigroup (in OPAALS), whilst simultaneously performing the Lie group analysis of the ODE system (BIONETS). Once these more powerful analytical tools are in place, we can re-examine the models discussed in this chapter, in order to correlate algebraic structure to observed or numerically calculated behaviour.

4 Autopoiesis-Inspired Computing

Natural systems provide unique examples of computation, in a form very different from contemporary computer architectures. Biology also demonstrates capabilities such as adaptation, self-repair and self-organisation that is becoming increasingly desirable for our technology [7]. Autopoietic systems (*auto* = self and *poiesis* = generating or producing) as a theoretical construct on the nature of living systems centre on two main notions: that of the circular organisation of metabolism and a redefinition of the systemic concepts of structure and organisation. This theoretical construct has found an important place in theoretical biology, but it could also be used as a foundation for a new type of Autopoiesis-Inspired Computing. We will first briefly explain our approach, followed by a minimal summary of the necessary *autopoietic theory*, before discussing the development of autopoietic computation [85].

In reference to Section 1.2.5, in the discussion that follows we focus on Structure and Organisation, but not on Function, because the theory of autopoiesis focuses on the former two concepts. Function is however implicit in Structure in biology, and our theory of bio-computing is attempting to replicate this interdependence. Function is also implicit in computing since any algorithm can be considered a function. So in autopoiesis-inspired computing we focus on the roles of Structure and Organisation at first, and will integrate also Function as formalisations become available from the other research threads.

Autopoiesis is a biological phenomenon, which requires (mathematical) formalisation, and then an understanding in terms of theoretical computing, to understand it in computational terms. Therefore, our efforts in experimental biology, mathematical modelling, and theoretical computing form a crucible of understanding from which to define a model for Autopoiesis-Inspired Computing, each building upon the next, as shown in Figure 16.

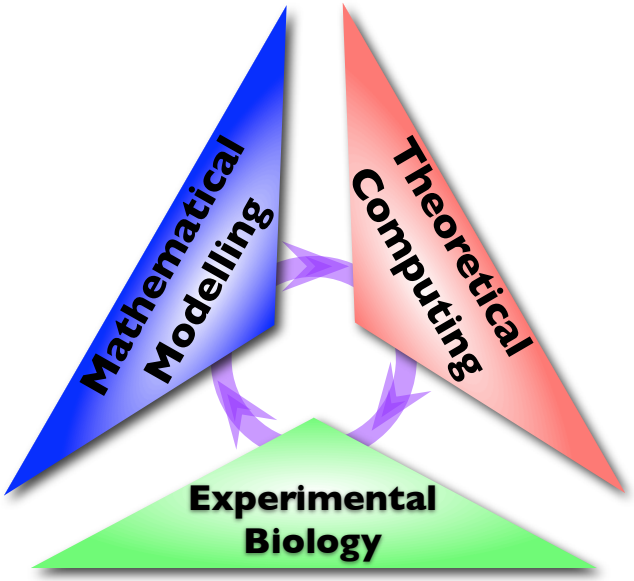

Figure 16: **Defining Autopoiesis-Inspired Computing: Computational biomimicry, as with all biomimicry, starts with identifying useful behaviour from a biological system. Followed by observation to understand the mechanisms or principles by which it operates, and therefore allowing for an abstract understanding of the behaviour. This can then be mimicked in a non-biological system and its performance and effectiveness evaluated [65].**

4.1 Autopoiesis

Autopoiesis explores the consequences of the operation of a system when it possesses a circular organisation that separates it from its surroundings. Autopoiesis asserts that the core of biological phenomena arises from circular organisation, and not from information processing, reproduction, the generation of *the* correct response to an outside stimulus or optimising metabolic fluxes by minimising energy use [85]. The essential turnover of components in autopoietic systems, as well as the destruction and creation of whole classes of molecules during ontogeny[12], means that these systems cannot be characterised within the scope of traditional Dynamical Systems Theory [87]. Also, as their structure can change, without changing the organisation, autopoietic systems cannot be described with a *fixed*-state space [74]. This is why in our research we are attempting to apply generalisations of Lie groups (e.g. Lie groupoids) to non-linear dynamical systems theory. Groupoids are like groups but with a partial function rather than a bijection from the group to itself, meaning that the group operation is not defined for all the elements. This results in the group properties being limited to a subset of the system, which we believe will capture the local character (in parameter space) of most biological symmetries. Similarly, we regard finite-state machines as an idealised approximation that is meant to capture the computational aspects of the cell over a time scale that is small relative to the duration of the cell cycle. Under these conditions, the local dynamics (in time) can be captured by treating the cell as a closed system and neglecting the constant, but relatively small, flux of new material flowing through it as an open system. With the above provisos, we can begin to build a conceptual framework of the cell from the point of view of its organisation, which will translate into software architecture once a mathematical formalisation has been developed by the other threads of WP1 research.

The notion of circular organisation, the central aspect of living systems, is the axiom in autopoiesis that:

> An autopoietic system (machine) is organised (defined as a unity) as a network of processes of production (transformation and destruction) of components which: (i) through their interactions and transformations continuously regenerate and realise the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in space in which they (the components) exist by specifying the topological domain of its realisation as such a network. [100, 145, 98, 105]

This is a somewhat abstract and somewhat self-referential definition, which is, therefore, nicely represented by the artistic interpretation of autopoiesis in Figure 17. In an autopoietic system the result of any given process is the production of components that eventually would be transformed by other processes in the network into the components of the first process. This property, termed *operational closure*, is an organisational property that perfectly coexists with the fact that living systems are, from a physical point of view, energetically and materially open systems [85]. The molecules that enter the system determine the organisation of a system, which generates pathways whose operation produces molecular structures that determine the physical system and the organisation of the system [55]. So, an autopoietic system does not have inputs or outputs in the traditional sense, instead it constitutes a web of interdependent (i.e. non-linearly coupled) molecular processes that maintain autopoietic organisation. As the internal dynamics of an autopoietic system is self-determined, there is no need to refer any operational (or organisational) aspect to the outside. Therefore, the external environment with

[12]Ontogeny (also ontogenesis or morphogenesis) (ontos present participle of *to be*, genesis *creation*) describes the origin and the development of an organism from the fertilised egg to its mature form [64].

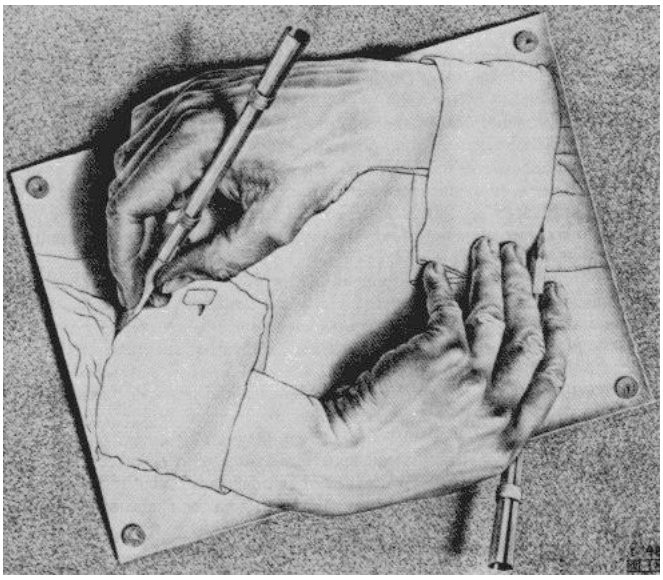

Figure 17: **Drawing Hands: An artistic representation of the process of autopoiesis. The left hand draws the right hand, and the right hand draws the left hand. From two dimensions, and through an *exchange of complementary information*, emerges a third dimension. Also, while the two complementary hands can draw each other, one hand cannot draw itself.**

which it interacts does not inform, instruct or otherwise define directly the internal dynamics, instead it indirectly perturbs the dynamics of the system.

The second clause (ii) demands that an autopoietic system have sufficiently complex dynamics to self-produce the boundaries that separate the system from the *non-system*. This apparently trivial clause has profound implications as it touches upon the dilemma of autonomy and serves to exclude out some pure formal systems [85]. So, autopoietic systems are not simple relational devices, but must conform to an important topological property, in that their boundary (in the space where their components exist) is actively produced by the network of processes. This property of autopoietic systems couples a purely relational property (operational closure) with a topological property, and demands that an autopoietic system be an autonomous unity, topographically and functionally segregated from its background [85]. In the realm of molecules, the coupling of these two conditions necessarily implies that the minimal metabolism compatible with autopoietic organisation must be rather more complex than the spatial coupling of a direct chemical reaction with its reverse reaction, and lends credence to our starting hypothesis that a profoundly non-linear mathematical structure must underpin the phenomenon.

In understanding autopoietic systems, it is important to distinguish between processes and components. Components interact through processes to generate other components. With this distinction, it is possible to define the organisation of a system as the pattern or configuration of processes between components that define the type of system. The structure, therefore, is the specific embodiment (implementation) of these processes into specific material (physical) entities. According to this definition, organisation is a subset of structure. Therefore, an autopoietic system is an entity that, with a variable and dynamic structure, maintains an invariant circular organisation. In this respect, autopoietic systems are rather different from man-made machines where only variables are maintained unchanged [85].

4.1.1 Formalisations

Autopoiesis, as originally described [99, 100], lacks any mathematical framework. Many attempts have been made to provide one and simulate autopoiesis. The first tessellation[13] computer models [145, 156], redone in Swarm[14] [101], were a direct translation of a minimal autopoietic system into a small bi-dimensional lattice. Indicational Calculus [21] was used [146] to model autonomous systems, but progress with this approach has been limited. Other mathematical formulations have included the use of differential equations to model feedback [90]. Another attempt, a pure algebraic approach, used the theory of categories to understand systems operating with *operational closure* [86].

However, all these formalisations lack one aspect or another [114], not least because many do not manage the non-Turing computability aspect prevalent in autopoietic systems, as discussed further below. So, none of these quantitative, or semi-quantitative models have generated clear, satisfactory results.

4.1.2 Structural coupling

Autopoietic systems do not simply behave or react passively in an environment that is provided. A central aspect of autopoiesis is the mechanism of structural coupling by which the living system and its medium determine, in a mutual way and resulting from a historic process, some of their properties. Figure 18 depicts the basis of the mechanism of structural coupling. The autopoietic system, represented by a circle, and defined by its structure and its organisation (hatched area), initially confronts a medium without organised *objects* (at t_0). As recurrent interactions (represented by the arrows) between the medium and the system are stabilised, at t_1, an *object* (represented by the triangle) begins to be configured. The *object* is made of two complementary parts. One part exists in the medium, and the other exists as a change in the autopoietic system structure. Finally, at t_2 the *object* is totally configured. So, the change in structure, but not in organisation as the hatched area remains unchanged, could be very important. For computing the important aspect is the existence of *objects* defined by spatio-temporal correlations, thus the change in the autopoietic system structure also contains these spatio-temporal correlations. Such spatio-temporal correlations take the form of complementary changes in interacting entities, which are called 'congruences'. So congruences are a consequence of structural coupling, meaning that the temporal changes in the structure of a system can manage future changes in the environment, because the spatio-temporal objects in the environment are congruent with the structure of the autopoietic system.

In effect, as the organisation of a system is maintained invariant, its structure can change in many dimensions that do not affect the organisation, for example its operational closure. However, this change is not random, being neither an accommodation nor adaptation to outside features (classical adaptationism), nor the result of the deployment of internal plans embodied in the structure of the autopoietic system (vitalism) [85]. Instead, the changes produced by structural coupling require the existence of recurrent interactions as well as a necessary level of plasticity[15] in the autopoietic system and its medium. During the ontogeny of the system (or the phylogeny[16] of the lineage) a congruence between the system and its medium is selected

[13] A tessellation or tiling of the plane is a collection of plane figures that fills the plane with no overlaps and no gaps [16].

[14] Swarm is a platform for agent-based models (www.swarm.org).

[15] The ability to change the structure.

[16] Phylogenetics is the study of evolutionary relatedness among various groups of organisms (e.g., species, populations).

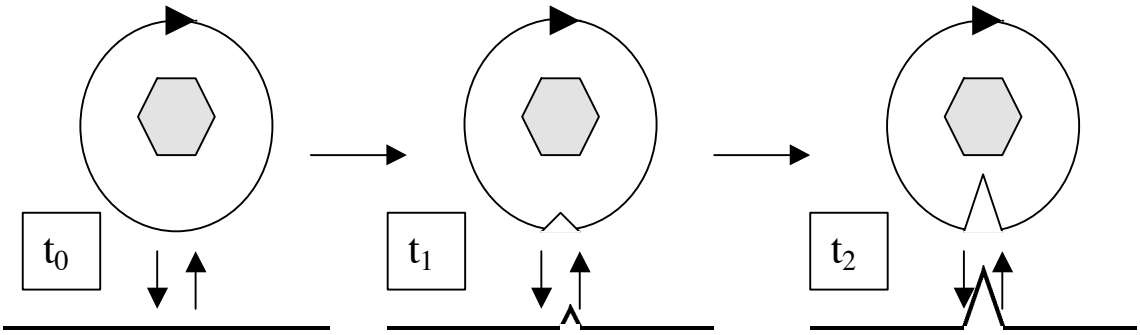

Figure 18: **Structural Coupling [85]: The autopoietic system, represented by a circle, defined by its structure and its organisation, initially confronts a medium without organised *objects*. As recurrent interactions between the medium and the system are stabilised, at t_1, an *object* begins to be configured. The *object* is made of two complementary parts. One part exists in the medium, and the other exists as a change in the autopoietic system structure.**

or stabilised, so the medium gradually becomes the environment and, for external observers unaware of the buildup of the relationship, the organism appears to become adapted to some characteristics of the medium. Therefore, autopoietic systems do not adapt to an earlier defined ecological niche, which is the crucial difference to the standard notion of evolution by adaptation, but create their environment by the systematic production of congruences [85].

These congruences produced by the structural coupling of the autopoietic system and its medium create meaning for the autopoietic system involved in the structural coupling, but not for external observers who do not live in the same environment as the system [80]. The structural change inside the autopoietic system is caused by the recurrent external trigger (perturbation) as well as its own internal, circular dynamics. So, a given external perturbation will not induce an internal structural change that can be viewed as its representation or internal model. The relation between the internal structural change and the external trigger is one of correlation or congruence instead of identity or isomorphism. The external perturbation does not induce a one-to-one model in the autopoietic entity, instead a congruence is constructed [85].

This concept could be the basis for a new type of autopoiesis-inspired form of computation, which would not be program-based [87]. As autopoietic systems do not have inputs or outputs, only a circular dynamic which is perturbed but not defined by external agents, it is not possible to encode outside concepts into autopoietic states, nor to control a trajectory of states (like Turing machines). So, an external observer can only define a computation for an autopoietic system as the particular ontogeny for that system. During the ontogeny of a system, a relation between it and its medium is selected or stabilised. This relation has meaning for the autopoietic system, which is structurally coupled to its medium, but not for external observers. Therefore, external observers (eventually users), if they wish to use autopoietic systems to perform computations, must find a procedure to attach the meaning of particular moments and properties to the ontogeny of the system [85].

4.1.3 Computability

Autopoietic systems are intrinsically different from Turing machines, the structure of which is shown in Figure 19. So, as they are not Turing-computable they also cannot be simulated by Turing machines, because the Turing-Church thesis[17] cannot be applied because the self-

[17]The Church-Turing thesis is a combined hypothesis about effectively calculable (computable) functions by recursion, by mechanical device equivalent to a Turing machine.

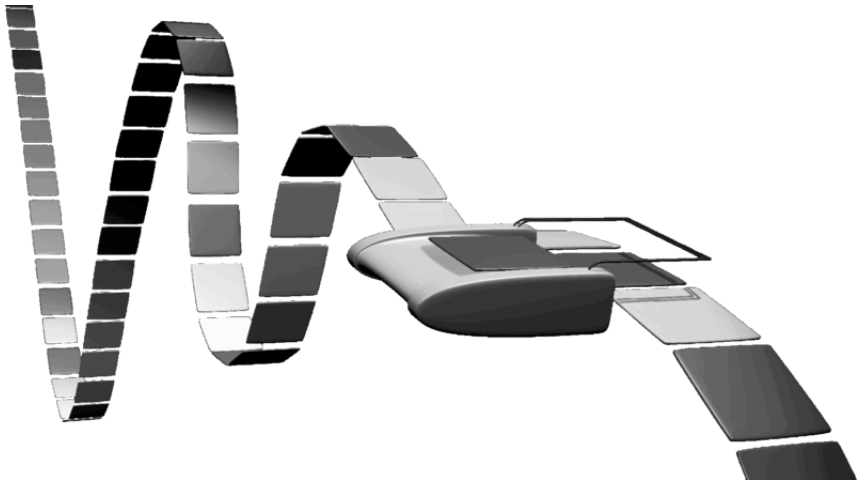

Figure 19: **Turing Machine: A basic abstract symbol-manipulating device which, despite its simplicity, can be adapted to simulate the logic of any computer algorithm. The Turing machine mathematically models a machine that mechanically operates on a tape for which symbols are written which it can read and write one at a time using a tape head. Operation is fully determined by a finite set of pre-determined elementary instructions contained within the Turing machine.**

referential nature of circularity demands an infinite number of states, which is non-Turing-computable [129, 131, 86]. The non-computability of autopoietic systems [74, 13] indicates that some intrinsic and fundamental part of their behaviour escapes our standard analysis based on phase states and/or evolution equations.

The non-computability of autopoietic systems by Turing machines has many important theoretical consequences [87]. First, it limits the validity of mimesis (i.e. simulation) as a means to understand living systems, showing that the phenomenology that arises from the circularity of metabolism cannot be simulated with current computer architectures, those based on the Von-Neumann implementation of Turing machines. Using different approaches this result has been hinted at on at least two occasions in the last decade [74, 13]. Using formal arguments, the impossibility of designing a living system without a real metabolism has been argued for [13]. A development for the concept of living systems, called *Component-systems*, has also been developed, for which it has been shown that equations of state, equations of motion or evolution equations cannot be applied [74]. This non-computability of autopoietic systems would not seem to be supported by past simulation results involving tessellation automatas [145]. However, new versions of this simulation show that the original report of computational autopoiesis was flawed, as it used a non-documented feature involving chain-based bond inhibition [101]. So, the closure exhibited by tessellation automatas is not a consequence of the *network* of simulated processes, but rather an artifact of coding procedures. So, the failure of closure in these computational models was not a conceptual failure of autopoiesis, but a reflection of the non-computability of autopoietic systems [87].

The non-computability of autopoietic systems could initially appear an overly strong result, but even in the restricted field of pure mathematics it has been possible to prove the existence of simple, but non-computable functions like the busy beaver[18] problem [121]. So, Turing non-

[18]In computability theory, a busy beaver (from the colloquial expression for *industrious person*) is a Turing machine which, when started on an empty tape, runs as long as possible, but eventually halts. This machine attains the limits on the amount of time and space that a halting Turing machine of the same class can consume. The busy beaver function quantifies those limits and is an example of a non-computable function. In fact, it can be shown to grow faster than any computable function. [121].

computability is a property that does not require the complexities of circular organisation to be apparent, as it is already demonstrable in simpler systems or problems [87]. The inapplicability of the Turing-Church thesis for autopoietic systems also opens some important new questions. The first being to analyse whether an autopoietic system can implement a Turing machine. The second, is considering whether some Turing non-computable problems, like the busy beaver [121], can be computed by autopoietic systems. Tackling these problems will require the use of *category theory* to represent autopoietic systems and to understand and manipulate the operational closure of metabolism.

4.2 Autopoietic Computation

To use autopoietic systems as computational tools, it is necessary to redefine the process of computation that we usually identify with the operation of Turing machines, which compute by performing symbol manipulation [85]. The symbol processing algorithm embodied in a Turing machine does not concern itself with the semantic content of those symbols, but only deals with the syntactical rules of symbol transformation. The semantics is left to the user who must map the string symbols with content. Autopoietic systems are significantly different to Turing devices, with their structure being variable, and hence they lack a true phase-space[19], because of intrinsic in-determination in biological systems from changes in the phase space during ontogenesis (morphogenesis). Also, they do not have direct inputs or outputs, only a circular dynamic that is perturbed but not defined by external agents. The autonomy of autopoietic systems implies that it is not possible to encode outside concepts, concepts pertaining to the requirements of users, into autopoietic states, nor to control a trajectory of states. Instead, users could only define a computation for an autopoietic system as the particular ontogeny for that system [85].

So, we require a new definition of *computing*, different from symbol processing. The unfamiliar computing aspects of autopoietic systems arise from the internal reference frame of the controller. The control is done by the logic of the maintenance of circular organisation [24] in the presence of structural coupling. The basic mechanism by which autopoietic systems can compute is the historic change in its structure triggered by recurrent temporal correlations [87]. This change is the consequence of the recurrent interactions between the autopoietic system and its medium. So, because of this relationship, every autopoietic system transforms the original medium of its deployment into an *environment* capable of computation.

Autopoietic processing would be rather different to reading a binary sequence from a uni-dimensional Turing tape. It would require a *mature* autopoietic system, where the necessary temporal correlations had been constructed, such that the structure of a system would manifest correlations to the environment. So, to compute with an autopoietic system, a historical link must be established between the autopoietic system and the medium (or space) in which we wish to perform computations. Therefore, an autopoietic system must be introduced in such a medium and a congruent lineage must be established via structural coupling [87], which would change the medium into an environment for this specific autopoietic system.

As time passes the structural changes induced by structural coupling would become more and more ingrained in the structure of the autopoietic system, which would capture more and more temporal (and spatial) correlations from its environment. This stage would be equivalent to the programming of a Turing machine, and once the autopoietic system is full of induced correlations it could be used for computation. So, ingrained in an autopoietic system, because of

[19] A phase space is a space in which all possible states of a system are represented, with each possible state of the system corresponding to one unique point in the phase space.

the structural coupling, would be the congruent dynamic of the environment. So, this procedure has the important advantage that programming is endogenously produced, by living inside the medium and forming a stable lineage [87]. However, it may not be possible to specify exactly the type of computations we desire. The intrinsic autonomy of autopoietic systems makes it impossible to force a system that has created its own relation with the environment, to capture the temporal correlations that are important for us, as our relation (perspective) to the environment is different to that of the system [85].

A solution to this conundrum can be achieved by the simple expedient of brute force given the age [58] of *multi-core processors*[20] has resulted in unused and underutilised cores being commonplace in modern personal computers [118], which would lend themselves well to the paradigm. Instead of establishing a single lineage, we could simultaneously use many different initial autopoietic systems, ideally with rather different structures. Each lineage would transform the single medium (i.e. the space recognised by the observer/autopoietic-programmer) into its environment, and so a wider range of temporal correlations could be established and some of them would be useful in performing computations (user-desired processing) [85].

4.3 Summary

We believe that the main contribution of the notion of autopoietic systems, in the endeavour of Autopoiesis-Inspired Computing, lies in constructing the notion of temporally correlated structural coupling in computing. Structural coupling is a mechanism by which a lineage of autopoietic systems can change their structure (i.e. components and processes), which is congruent with the recurrent perturbations that arise in the medium. Albeit autopoietic computing possibly requires hardware that we do not currently produce, we believe we will be able to implement these ideas once more concrete formalisations are available.

[20] A multi-core processor is an integrated circuit to which two or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks [**?**].

5 Conclusion

This report has presented a broad framework for the integration of the disciplines underpinning bio-computing, and a multi-threaded research workflow to traverse it.

The challenge of bio-computing as summarised in this report has attracted nine partners across 3 projects and is attracting more partners for future projects, as shown in the table below.

Project	Institutions
DBE	LSE, Imperial College, University of Birmingham, SUAS
OPAALS	LSE, UNIVDUN, UH, IITK, Surrey, Naica
BIONETS	LSE, University of Passau, Fraunhofer-Fokus
New project initiatives	LSE, University of Passau, Fraunhofer-Fokus, UNIVDUN, UH, IITK Surrey, SUAS, Trinity College Dublin, University of Lecce, Aerospace Corporation, Karolinska Institutet, ...

Because these institutions are spread over several disciplines (computer science, biology, mathematics, physics), it has taken a long time to develop a problem statement that everyone can subscribe to, even though not everyone understands all of its parts and implications. In the spirit of OPAALS as a Network of Excellence we have sought to leverage the synergy between bio-computing (which aims to develop new models of computation inspired by biology) and computational systems biology (which aims to develop new modelling frameworks for cell biology inspired by computer science), and the results in terms of community building and language construction are starting to be felt, as we think can be seen especially from the two most recent reports, this one and D2.2.9.

No hard conclusions can be drawn yet regarding the mathematical structure that underpins bio-computing. This is the part of the research that is more concerned with the formalisation of the interplay between **Structure** and **Function**. However, we hope that the framework that is emerging is starting to look plausible. This work will be extended, hopefully reaching useable conclusions, in D1.4, as well as in the BIONETS series D2.2.9 and D3.2.7.

The emerging thread of autopoiesis-inspired computing, which is architectural in flavour and more concerned with biological **Organisation**, is starting to offer some interesting ideas and possibilities. These are making the need for a mathematical formalisation of interaction computing and its derivatives increasingly urgent. A model of interaction computing is under construction by the UH, UNIVDUN and LSE partners (leading to D1.4), who will collaborate closely during the course of Year 4 with IITK, Surrey, Naica, and LSE to achieve a formalisation of autopoiesis-inspired computing in D1.5.

The underlying and unifying theory of categories is being synthesised within the BIONETS project, in communication with UH, to provide a translation from the algebraic structures of automata and differential equations to logics, with the objective of developing a specification language with operational semantics that can support the specification of generative software environments such as a biologically-inspired software factory. Applications of interest that are being pursued are symbiotic security services (in BIONETS) and dynamic instantiation of business workflows from declarative SBVR statements (in OPAALS). Finally, we are also investigating the suitability of a RESTful interaction model and architecture as a formalisation framework that can implement the bio-computing principles and models we are developing over web environments.

The need to define areas of application of the theory is motivated by Objective 7 of the OPAALS Description of Work (Table 2.1, p. 5): "Context-dependent automatic code structure generation from domain-specific natural language models". During the course of the last 3 years this objective has grown in scope and increased in definition. As mentioned, we now see the possibiliby of connecting interaction computing ideas also to RESTful environments. At the same time, whereas we don't think we can achieve the automatic generation of software services from business model specifications parsed from natural language descriptions, generation of business processes from SBVR declarative specifications is a slightly less ambitious goal about which we may be able to say something useful by the end of the project. The integration with the work of SUAS to generate web applications from SBVR models would then become more feasible, perhaps for a future project.

Having said that, the more important point is that the need for a mathematical framework that underpins these desirable properties of software has become increasingly apparent. Therefore, we view the output of WP1 as one that is mainly theoretical and mathematical, since such a mathematical framework is clearly necessary before we can achieve a formalisation that supports autopoiesis-inspired computing in all these areas of application.

The next concern is that if, as discussed in Chapter 4, autopoietic systems are not Turing-computable, how can we hope to arrive at these applications? Whereas this might seem like a show-stopper, in fact if we look at biology we can see that autopoietic systems (e.g. the cell) are able to function perfectly well. This motivates us to think, on the one hand, that the theory of what is computable could be extended or generalised; and, on the other hand, that our reliance on other disciplines such as dynamical systems and algebra does not necessarily make the attainment of the bio-computing vision contingent on a strictly computer science-based theory of computability – as we understand it today. We believe that the analytical tools at our disposal are such as to make significant process in understanding how the cell 'computes' possible within the life of the project, leaving issues of extending the theory of computation perhaps to a future research effort.

Figure 20 summarises the mathematical framework for bio-computing as we understand it now. This framework should be understood as a workflow that takes place within the iterative process depicted in Figure 16, hence the feedbacks are not shown explicitly. Because the objective is to develop a model rather than a metaphor of biological processes, the epistemology the research effort departs from is necessarily rooted in physics and mathematics. Hence, the corresponding methodology is to begin with as simple a problem statement as possible and to add complexity as the mathematical framework emerges and is verified (or falsified, see D12.10). It is for this reason that autopoiesis does not appear at the top of this diagram: it is much too complex a phenomenon to be a viable starting point for a mathematical theory. Autopoiesis can be regarded as an infinite recursion on the concept, or the phenomenon, of self-organisation. Accordingly, therefore, our approach since the DBE project has been to prioritise an understanding of self-organisation at the level of gene expression before attempting generalisation of bio-computing towards autopoiesis and, eventually, evolution.

The research, however, has opened new doors and new possibilities we had not foreseen. Namely, the association of autopoiesis with Organisation and ultimately with Architecture[21] has made it possible to engage with software architecture and software engineering thinking much earlier than we thought possible. The node in the diagram labelled "Architecture of self-organisation", therefore, becomes a possible accumulation and integration point between the different research streams. This will be reported in deliverable D1.5 at the end of the project.

[21] We owe the latter association to Prof T V Prabhakar of IITK.

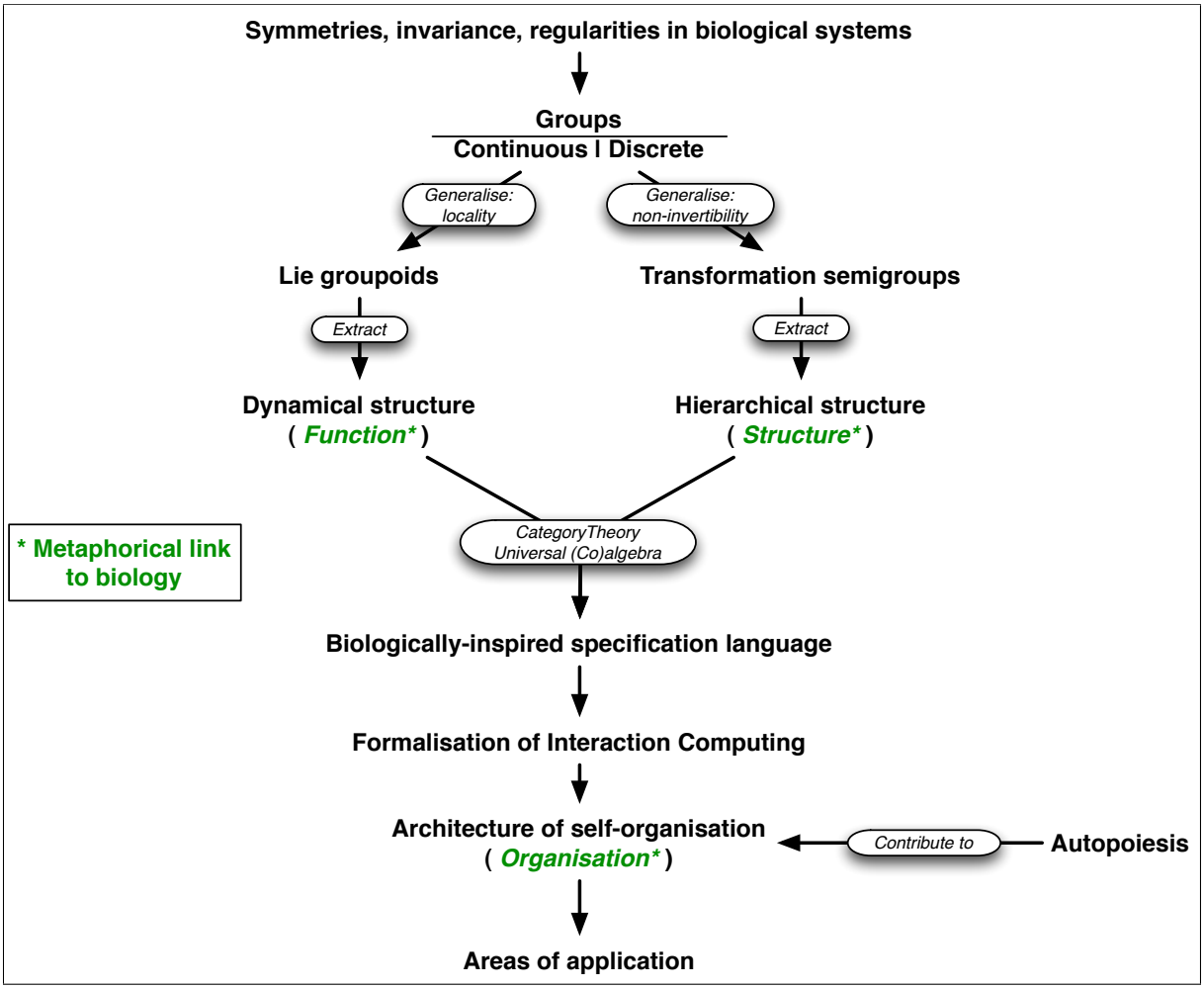

Figure 20: **Current status of the bio-computing mathematical framework (feedbacks not shown). This framework should be understood as a workflow that takes place within the iterative process depicted in Figure 16**

References

[1] W Abou-Jaude, D A Ouattara, and M Kaufman. From structure to dynamics: Frequency tuning in the p53-mdm2 network. *J Theor Biol*, 258:561–577, 2009. I. Logical approach.

[2] S Agrawal, C Archer, and D V Schaffer. Computational models of the notch network elucidate mechanisms of context-dependent signaling, *PLoS Comput Biol*, 5(5):e1000390. 2009.

[3] H. Andréka, István Németi, and Ildikó Sain. *Universal Algebraic Logic*. Studies in Universal Logic. Springer, 1st edition, to appear.

[4] J M Ankers, D G Spiller, M R H White, and C V Harper. Spatio-temporal protein dynamics in single living cells, *Curr Opinion Biotechnol* 19: 375–380. 2009.

[5] E Batchelor, A Loewer, and G Lahav. The ups and downs of p53: understanding protein dynamics in single cells, *Nat Rev Cancer* 9(5): 371–377. 2009.

[6] E Batchelor, C S Mock, I Bhan, A Loewer, and G Lahav. Recurrent initiation: A mechanism for triggering p53 pulses in response to dna damage, *Mol Cell* 30: 277–289. 2008.

[7] P.J. Bentley. Systemic computation: A model of interacting systems with natural characteristics. *International journal of parallel, emergent and distributed systems*, 22(2):103–121, 2007.

[8] J. A. Bergstra and J. W. Klop. ACPτ: a universal axiom system for process specification. pages 447–463, 1989.

[9] S Bernard, B Cajavec, L Pujo-Menjouet, M C Mackey, and H Herzel. Modelling transcriptional feedback loops: The role of gro/tle1 in hes1 oscillations, *Phil Trans R Soc A* 364: 1155–1170. 2006.

[10] Y Bessho and R Kageyama. Oscillations, clocks and segmentation, *Current opinion in genetics and development* 13(4):379-384. 2003.

[11] M Bier, B Teusink, B N Kholodenko, and HV Westerhoff. Control analysis of glycolytic oscillations, *Biophys Chem*, 62(1-3):15-24. 1996.

[12] Dines Bjørner and Cliff B. Jones, editors. *The Vienna Development Method: The Meta-Language*, volume 61 of *Lecture Notes in Computer Science*. Springer, 1978.

[13] MA Boden. Is metabolism necessary? *The British Journal for the Philosophy of Science*, 50(2):231–248, 1999.

[14] Tommaso Bolognesi and Ed Brinksma. Introduction to the ISO specification language LOTOS. *Comput. Netw. ISDN Syst.*, 14(1):25–59, 1987.

[15] S Bottani and B Grammaticos. Analysis of a minimal model for p53 oscillations, *J Theor Biol* 249: 235–245. 2007.

[16] A. Bowyer. Computing dirichlet tessellations*. *The Computer Journal*, 24(2):162–166, 1981.

[17] G Briscoe. *D6.1-Entropy-Based Complexity Measure for the Evolution-Based Self-Organisation of Agent Populations*. DBE Project, http://files.opaals.org/DBE/deliverables, 2004.

[18] G Briscoe and P De Wilde. *D6.2-Self-Organisation of Evolving Service Populations*. DBE Project, http://files.opaals.org/DBE/deliverables, 2005.

[19] G Briscoe and P De Wilde. *D6.4-Intelligence, learning and neural networks in distributed agent systems*. DBE Project, http://files.opaals.org/DBE/deliverables, 2005.

[20] G Briscoe and P De Wilde. *D6.5-The effect of distributed intelligence in evolutionary dynamics*. DBE Project, http://files.opaals.org/DBE/deliverables, 2006.

[21] G.S. Brown. *Laws of form*. Allen & Unwin, London, 1969.

[22] E Buckingham. The principle of similitude. *Nature*, 96:396–397, 1915.

[23] H M Byrne, I M M van Leeuwen, M R Owen, T A Alarcon, and P K Maini. Multiscale modelling of solid tumour growh, in *Selected Topics on Cancer Modelling: Genesis, Evolution, Immune Competition and Therapy* (n bellomo, maj chaplain and e de angelis, editors). 2008. Series: Modelling and Simulation in Science, Engineering and Technology, Birkhauser, Boston.

[24] J.L. Casti. The simply complex:" biologizing" control theory: How to make a control system come alive, the big problems of control theory. *Complexity*, 7(4):10–12, 2002.

[25] V Chickarmane, A Nadim, A Ray, and HM Sauro. A p53 oscillator model of dna break repair control, available from: http://arxiv.org/abs/q-bio.mn/0510002. 2006.

[26] D Chu and W K Ho. A Category Theoretical Argument Against the Possibility of Artificial Life: Robert Rosens Central Proof Revisited. *Artificial Life*, 12:117–134, 2006.

[27] D Chu and W K Ho. Computational Realizations of Living Systems. *Artificial Life*, 13:369381, 2007.

[28] D Chu and W K Ho. The Localization Hypothesis and Machines. *Artificial Life*, 13:299302, 2007.

[29] A Ciliberto, B Novak, and J J Tyson. Steady states and oscillations in the p53/mdm2 network, *Cell Cycle* 4: 488–493. 2005.

[30] E M Clarke, O Grumberg, and D Peled. *Model Checking*. MIT Press, Cambridge, MA, 1999.

[31] A Cornish-Bowden and M L Cardenas. Self-organization at the origin of life. *Journal of Theoretical Biology*, 252:411–418, 2008.

[32] M L Dequeant, E Glynn, K Gaudenz, M Wahl, J Chen, A Mushegian, and O Pourquie. A complex oscillating network of signaling genes underlies the mouse segmentation clock, *Science* 314: 1595–1598. 2006.

[33] P Dini. *D18.4-Report on self-organisation from a dynamical systems and computer science viewpoint*. DBE Project, www.digital-ecosystem.org, Deliverables: SP5, 2007.

[34] P Dini and E Berdou. *D18.1-Report on DBE-Specific Use Cases*. DBE Project, http:files.opaals.org/DBE, 2004.

[35] P Dini, G Briscoe, A J Munro, and S Lain. *D1.1: Towards a Biological and Mathematical Framework for Interaction Computing*. OPAALS Deliverable, European Commission, http://files.opaals.org/OPAALS/ Year_2_Deliverables/WP01/, 2008.

[36] P Dini, G Horvath, D Schreckling, and H Pfeiffer. *D2.2.9: Mathematical Framework for Interaction Computing with Applications to Security and Service Choreography*. BIONETS Deliverable, European Commission, http://www.bionets.eu, 2009.

[37] P Dini, A J Munro, M Iqani, F Zeller, S Moschoyiannis, J Gabaldon, and O Nykanen. *D1.2: Foundations of the Theory of Associative Autopoietic Digital Ecosystems: Part 1*. OPAALS Deliverable, European Commission, http://files.opaals.org/OPAALS/Year_2_Deliverables/WP01/, 2008.

[38] P Dini and D Schreckling. More Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. In *1st OPAALS Workshop, Rome, 26-27 November*, 2007.

[39] P Dini and D Schreckling. On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. In E Altman, P Dini, D Miorandi, and D Schreckling, editors, *D2.1.1 Paradigms and Foundations of BIONETS research*, http:www.bionets.eu, 2007.

[40] P Dini and D Schreckling. 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, 26-29 February*, 2008.

[41] P Dini, D Schreckling, and L Yamamoto. *D2.2.4: Evolution and Gene Expression in BIONETS: A Mathematical and Experimental Framework*. BIONETS Deliverable, European Commission, http://www.bionets.eu, 2008.

[42] D D Dionysiou and G S Stamatakos. Applying a4d multiscale in vivo tumour growth model to the exploration of radiotherapy scheduling: The effects of weekend treatment gaps and p53 gene status on the response of fast growing solid tumors, *Cancer Informatics* 2: 113–121. 2006.

[43] Pál Dömösi and Chrystopher L. Nehaniv. *Algebraic Theory of Finite Automata Networks: An Introduction*. SIAM Series on Discrete Mathematics and Applications, 2005.

[44] A. Egri-Nagy and C. L. Nehaniv. Hierarchical decomposition – coordinate systems for understanding complexity. In *Proceedings of the Evolution of Complexity Workshop at Artificial Life X*, 2006. Journal version published in *Artificial Life* 14(3):299–312, 2008, Special issue on "Evolution of Complexity".

[45] A Egri-Nagy and C L Nehaniv. Algebraic properties of automata associated to petri nets and applications to computation in biological systems. *BioSystems*, 94(1-2):135–144, 2008.

[46] A. Egri-Nagy, C. L. Nehaniv, J. L. Rhodes, and M. J. Schilstra. Automatic analysis of computation in biochemical reactions. *BioSystems*, 94(1-2):126–134, 2008.

[47] Attila Egri-Nagy and Chrystopher L. Nehaniv. 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),.

[48] Attila Egri-Nagy and Chrystopher L. Nehaniv. `SgpDec` – software package for hierarchical coordinatization of groups and semigroups, implemented in the `GAP` computer algebra system. (`http://sgpdec.sf.net`)., 2008.

[49] M Eigen and P Schuster. The Hypercycle. *Naturwissenschaften*, 65(1), 1978.

[50] Samuel Eilenberg. *Automata, Languages and Machines*, volume B. Academic Press, 1976.

[51] Gillian Z. Elston and Chrystopher L. Nehaniv. Holonomy embedding of arbitrary stable semigroups. *International Journal of Algebra and Computation*, 12(6):791–810, 2002.

[52] Zoltán Ésik. A proof of the Krohn-Rhodes decomposition theorem. *Theoretical Computer Science*, 234:287–300, 2000.

[53] G Farmer, J Bargonetti, H Zhu, P Friedman, R Prywes, and C Prives. Wild-type p53 activates transcription in vivo, *Nature* 358: 83–84. 1992.

[54] R Fielding. Architectural Styles and the Design of Network-based Software Architectures. 2000.

[55] G.R. Fleischaker. Origins of life: an operational definition. *Origins of Life and Evolution of Biospheres*, 20(2):127–137, 1990.

[56] R P Gabriel and R Goldman. Conscientious software. In *OOPSLA06, Portland, Oregon, 22-26 October*, 2006.

[57] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4*, 2006. (`http://www.gap-system.org`).

[58] D. Geer. Chip makers turn to multicore processors. *IEEE Computer*, 38(5):11–13, 2005.

[59] N Geva-Zatorsky, N Rosenfeld, S Itzkovitz, R Milo, A Sigal, E Dekel, T Yarnitzky, Y Liron, P Polak, G Lahav, and U Alon. Oscillations and variability in the p53 system, *Mol Systems Biol.* 2006.

[60] Abraham Ginzburg. *Algebraic Theory of Automata.* Academic Press, 1968.

[61] A Goldbeter. Computational approaches to cellular rhythms, *Nature*, 420(6912):238-45. 2002.

[62] M Golubitsky and I Stewart. Nonlinear Dynamics of Networks: The Groupoid Formalism. *Bulletin of the American Mathematical Society*, 43:305–364, 2006.

[63] K E Gordon, I M M van Leeuwen, S Lain, and M A J Chaplain. Spatio-temporal modelling of the p53-mdm2 oscillatory system, *Math Model Nat Phenom*, in press. 2009.

[64] S.J. Gould. *Ontogeny and phylogeny.* Belknap press, 1977.

[65] Carl Hastrich. Biomimicry: A tool for innovation. Technical report, The Biomimicry Institute, 2007. Available from: `http://www.biomimicryinstitute.org/about-us/biomimicry-a-tool-for-innovation.html`.

[66] Y Haupt, R Maya, A Kazaz, and M Oren. Mdm2 promotes the rapid degradation of p53, *Nature* 387: 296–299. 1997.

[67] T Heistracher, T Kurz, G Marcon, and C Masuch. *D9.1-Report on Fitness Landscape.* DBE Project, http://files.opaals.org/DBE/deliverables, 2005.

[68] Karsten Henckell, Susan Lazarus, and John L. Rhodes. Prime decomposition theorem for arbitrary semigroups. *Journal of Pure and Applied Algebra*, 55:121–172, 1988.

[69] H Hirata, S Yoshiura, T Ohtsuka, Y Bessho, T Harada, K Yoshikawa, and R Kageyama. Oscillatory expression of the bhlh factor hes1 regulated by a negative feedback loop, *Science* 298: 840–843. 2002.

[70] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[71] W. M. L. Holcombe. *Algebraic Automata Theory*. Cambridge University Press, 1982.

[72] G Horvath. *Functions and Polynomials over Finite Groups from the Computational Perspective*. The University of Hertfordshire, PhD Dissertation, 2008.

[73] R E Kalman, P L Falb, and M A Arbib. *Topics in Mathematical System Theory*. McGraw-Hill, New York, 1969.

[74] G. Kampis and G. Kampis. *Self-modifying systems in biology and cognitive science*. Pergamon Press Oxford, 1991.

[75] S Kauffman. *The Origins of Order: Self-Organisation and Selection in Evolution*. Oxford University Press, Oxford, 1993.

[76] S Krishna, M H Jensen, and K Sneppen. Minimal model of spiky oscillations in nf-kappab, *Proc Natl Acad Sci USA* 103:(15)0840–1084. 2006.

[77] K Krohn and J Rhodes. Algebraic Theory of Machines. I. Prime Decomposition Theorem for Finite Semigroups and Machines. *Transactions of the American Mathematical Society*, 116:450–464, 1965.

[78] Kenneth Krohn, John L. Rhodes, and Bret R. Tilson. The prime decomposition theorem of the algebraic theory of machines. In Michael A. Arbib, editor, *Algebraic Theory of Machines, Languages, and Semigroups*, chapter 5, pages 81–125. Academic Press, 1968.

[79] J P Kruse and W Gu. Modes of p53 regulation, *Cell* 137: 609–622. 2009.

[80] K. Kull. Jakob von uexkull: An introduction. *Semiotica*, 2001(134):1–59, 2001.

[81] T Kurz, G Marcon, H Okada, T Heistracher, and A Passani. *D9.2-Report on Evolutionary and Distributed Fitness Environment*. DBE Project, http://files.opaals.org/DBE/deliverables, 2006.

[82] G Lahav. Oscillations by the p53-mdm2 feedback loop, *Adv Exp Med Biol* 641:28-38. 2008.

[83] G Lahav, N Rosenfield, A Sigal, N Geva-Zatorsky, A J Levine, M B Elowitz, and U Alon. Dynamics of the p53-mdm2 feedback loop in individual cells, *Nat Gen* 36: 147–150. 2004.

[84] D P Lane. p53, guardian of the genome, *Nature* 358:15–16. 1992.

[85] J.C. Letelier, G. Marín, and J. Mpodozis. Computing With Autopoietic Systems. *Soft Computing and Industry Applications. Springer, London*, 2002.

[86] J.C. Letelier, G. Marın, and J. Mpodozis. Autopoietic and (m, r) systems. *Journal of theoretical biology*, 222(2):261–272, 2003.

[87] J.C. Letelier, G. Marin, J. Mpodozis, and J. Soto-Andrade. Anticipatory Computing with Autopoietic and (M R) System. *Soft Computing Systems: Design, Management and Applications*, page 205, 2002.

[88] R Lev Bar-Or, R Maya, L A Segel, U Alon, A J Levine, and M Oren. Generation of oscillations by the p53-mdm2 feedback loop: A theoretical and experimental study, *Proc Natl Acad Sci USA* 97: 11250–11255. 2000.

[89] H A Levine, M W Smiley, A L Tucker, and M Nilsen-Hamilton. A mathematical model for the onset of avascular tumor growth in response to the loss of p53 function, *Cancer Informatics* 2: 163-188. 2006.

[90] A. Limone. L'autopoiese dans les organisations. *Doctorat de Troisieme cycle. Universit*, 1977.

[91] A H Louie. A Living System Must Have Noncomputable Models. *Artificial Life*, 13:293297, 2007.

[92] A H Louie. Interpreting Rosen. *Artificial Life*, 13:291292, 2007.

[93] L Ma, J Wagner, J J Rice, W Hu, A J Levine, and G A Stolovitzky. A plausible model for the digital response of p53 to dna damage, *Proc Natl Acad Sci USA* 102: 14266–14271. 2005.

[94] S Maeda. The similarity method for difference equations. *IMA Journal of Applied Mathematics*, 38:129–134, 1987.

[95] S C Manrubia, A S Mikhailov, and D H Zanette. *Emergence of Dynamical Order*. World Scientific, Singapore, 2004.

[96] M Maroto and N A M Monk. *Cellular oscillatory mechanisms*. Springer, Austin, Texas, 2008.

[97] H Maturana and F Varela. *Autopoiesis and Cognition. the Realization of the Living*. D. Reidel Publishing Company, Boston, 1980.

[98] H.R. Maturana and F. Varela. Autopoietic systems. *Urbana, Biological Computer Laboratory University of Illinois*, 1975.

[99] H.R. Maturana and F.J. Varela. De mdquinas y seros vivos. *Santiago, Chile: Editorial Universitaria. Translated as Autopoiesis and Cognition: The Realization of the Living. Dordrecht, Holland: D. Reidel (1980)*, 1972.

[100] H.R. Maturana and F.J. Varela. *Autopoiesis and cognition: The realization of the living*. Springer, 1980.

[101] B. McMullin and F.J. Varela. Rediscovering computational autopoiesis. In *Fourth European Conference on Artificial Life*, pages 38–47, 1997.

[102] G I Mihalas, Z Simon, G Balea, and E Popa. Possible oscillatory behaviour in p53-mdm2 interaction computer simulation, *J Biol Syst* 8: 21–29. 2000.

[103] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[104] Robin Milner, Joachim Parrow, and David Walker. A Calculus of Mobile Processes, I. *Inf. Comput.*, 100(1):1–40, 1992.

[105] J. Mingers. *Self-producing systems: Implications and applications of autopoiesis*. Springer, 1995.

[106] N A M Monk. Oscillatory expression of hes1, p53, and nf-kappab driven by transcriptional time delays, *Curr Biol* 13(16):1409–1413. 2003.

[107] C L Nehaniv and J L Rhodes. The Evolution and Understanding of Hierarchical Complexity in Biology from an Algebraic Perspective. *Artificial Life*, 6:45–67, 2000.

[108] Chrystopher L. Nehaniv. Algebraic engineering of understanding: Global hierarchical coordinates on computation for the manipulation of data, knowledge, and process. In *Proc. 18th Annual International Computer Software and Applications Conference (COMPSAC 94)*, pages 418–425. IEEE Computer Society Press, 1994.

[109] Chrystopher L. Nehaniv. Algebra and formal models of understanding. In Masami Ito, editor, *Semigroups, Formal Languages and Computer Systems*, volume 960, pages 145–154. Kyoto Research Institute for Mathematics Sciences, RIMS Kokyuroku, August 1996.

[110] Chrystopher L. Nehaniv. A simple, direct proof of the Krohn-Rhodes theorem. In Y. Kobayashi and R. Matsuda, editors, *Proceedings of the 20th Symposium on Semigroups, Languages, and Their Related Areas*, pages 29–33, Mito, Japan, November 1996.

[111] Chrystopher L. Nehaniv and John L. Rhodes. The evolution and understanding of hierarchical complexity in biology from an algebraic perspective. *Artificial Life*, 6(1):45–67, 2000.

[112] D E Nelson, A E Ihekwaba, M Elliott, J R Johnson, C A Gibney, B E Foreman, G Nelson, V See, C A Horton, D G Spiler, S W Edwards, H P McDowell, J F Unitt, E Sullivan, R Grimley, N Benson, D Broomhead, D B Kell, and M R White. Oscillations in nf-kappab signaling control de dynamics of gene expression, *Science* 306:704–708. 2004.

[113] G Nicolis and I Prigogine. *Self-Organization in Nonequilibrium Systems*. Wiley, New York, 1977.

[114] T. Nomura. Category theoretical distinction between autopoiesis and (M, R) systems. *Lecture Notes in Computer Science*, 4648:465, 2007.

[115] B A Ogunnaike. Elucidating the digital control mechanism for dna damage repair with the p53-mdm2 system: Single cell data analysis and ensemble modelling, *J R Soc Interface* 3: 175–184. 2006.

[116] P Olver. *Applications of Lie Groups to Differential Equations*. Springer, 1986.

[117] S Pigolotti, S Krishna, and M H Jensen. Oscillation patterns in negative feedback loops, *Proc Natl Acad Sci USA* 104: 6533–6537. 2007.

[118] B Posey. Multi-core processors: Their implication for windows. 2007. http://searchwindowsserver.techtarget.com/tip.

[119] C J Proctor and D A Gray. Explaining oscillations and variability in the p53-mdm2 system, *BMC Syst Biol* 2(1): 75. 2008.

[120] K Puszynski, B Hat, and T Lipniacki. Oscillations and bistability in the stochastic model of p53 regulation, *J Theor Biol* 254: 452–465. 2008.

[121] T. Rado. On non-computable functions. *Bell System Technical Journal*, 41(3):877–884, 1962.

[122] N Rashevsky. *Mathematical Biophysics and Physico-Mathematical Foundations of Biology Vol II*. Dover, New York, 1960.

[123] John L. Rhodes. *Applications of Automata Theory and Algebra via the Mathematical Theory of Complexity to Biology, Physics, Psychology, Philosophy, and Games*. World Scientific Press (in press), 2009. Foreword by Morris W. Hirsch, edited by Chrystopher L. Nehaniv (Original version: University of California at Berkeley, Mathematics Library, 1971).

[124] John L. Rhodes and Pascal Weil. Decomposition techniques for finite semigroups using categories, I. *Journal of Pure and Applied Algebra*, 62:269–284, 1989.

[125] John L. Rhodes and Pascal Weil. Decomposition techniques for finite semigroups using categories, II. *Journal of Pure and Applied Algebra*, 62:285–312, 1989.

[126] B Ribba, T Colin, and S Schnell. A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies, *Theor Biol Med Model* 3: 7. 2006.

[127] R Rosen. A Relational Theory Of Biological Systems. *Bulletin of Mathematical Biophysics*, 20:245–260, 1958.

[128] R Rosen. The Representation Of Biological Systems From The Standpoint Of The Theory Of Categories. *Bulletin of Mathematical Biophysics*, 20:317–341, 1958.

[129] R. Rosen. Abstract biological systems as sequential machines: Iii. some algebraic aspects. *Bulletin of Mathematical Biology*, 28(2):141–148, 1966.

[130] R Rosen. Some relational cell models: The metabolism-repair systems. In R Rosen, editor, *Foundations of Mathematical Biology, Vol II: Cellular Systems*. Academic Press, 1972.

[131] R Rosen. *Life Itself*. Columbia University Press, New York, 1991.

[132] J E Rowe and B Mitavskiy. D8.1 - report on evolution of high-level software components. Version 1.0, April 2005.

[133] R Sanchez, R Grau, and E Morgado. A novel Lie algebra of the genetic code over the Galois field of four DNA bases. *Mathematical Biosciences*, 202:156–174, 2006.

[134] R Sanchez, E Morgado, and R Grau. The genetic code boolean lattice. *Communications in Mathematical and Computational Chemistry*, 52:29–46, 2004.

[135] R Sanchez, E Morgado, and R Grau. Gene algebra from a genetic code algebraic structure. *Journal of Mathematical Biology*, 51:431–457, 2005.

[136] D Schreckling and P Dini. Distributed Online Evolution: An Algebraic Problem? In *IEEE 10th Congress on Evolutionary Computation, Trondheim, Norway, 18-21 May*, 2009.

[137] S Y Shieh, M Ikeda, Y Taya, and C Prives. Dna damage-induced phosphorylation of p53 alleviates inhibition by mdm2, *Cell* 91: 325–334. 1997.

[138] J. M. Spivey. *The Z notation: a reference manual*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[139] J Srividya, M S Gopinathan, and S Schnells. The effects of time delays in a phosphorylation-dephosphorylation pathway, *Biophys Chem* 125: 286–297. 2007.

[140] I Stewart. *Galois Theory (2nd Ed.)*. Chapman and Hall, London, 1989.

[141] G Tiana, M H Jensen, and K Sneppen. Time delay as a key to apoptosis induction in the p53 network, *Eur Phys J B*, 29: 135–140. 2002.

[142] G Tiana, S Krishna, S Pigolotti, M H Jensen, and K Sneppen. Oscillations and temporal signalling in cells, *Phys Biol*, 4: R1–r17. 2007.

[143] M Tigges, T T Marquez-Lago, J Stelling, and M Fussenegger. A tunable synthetic mammalian oscillator, *Nature*, 457(7227):309-312. 2009.

[144] J J Tyson. Monitoring p53's pulse, *Nat Genet*, 36(2):113-114. 2004.

[145] FG Varela, HR Maturana, and R. Uribe. Autopoiesis: the organization of living systems, its characterization and a model. *Currents in modern biology*, 5(4):187, 1974.

[146] F.J. Varela. *Principles of biological autonomy*. North Holland New york, 1979.

[147] B Vogelstein, D P Lane, and A J Levine. Surfing the p53 network, *Nature*, 408: 307–310. 2007.

[148] R F Voss and J Clarke. 1/f noise in music: Music from 1/f noise. *Journal of the Acoustical Society of America*, 63(1), 1978.

[149] J Wagner, L Ma, J J Rice, W Hu, A J Levine, and G A Stolovitzky. P53-mdm2 loop controlled by a balance of its feedback strength and effective dampening using atm and delayed feedback, *IEE Proc Syst Biol* 152: 109–118. 2005.

[150] C Wawra, M Kuhl, and H A Kestler. Extended analyses of the wnt/beta-catenin pathway: robustness and oscillatory behaviour, *FEBS Lett*, 581:4043–4048. 2007.

[151] A Weinstein. Groupoids: unifying internal and external symmetry. *Notices of the American Mathematical Society*, 43:744–752, 1996.

[152] C W Wu. *Synchronization in Coupled Chaotic Circuits and Systems*. World Scientific, Singapore, 2002.

[153] A Zauberman, D Flusberg, Y Haupt, Y Barak, and M Oren. A functional p53-response intronic promoter is contained within the human mdm2 gene, *Nucleic Acids Res*, 23: 2584–2592. 1995.

[154] H. Paul Zeiger. Cascade synthesis of finite state machines. *Information and Control*, 10:419–433, 1967. plus erratum.

[155] H. Paul Zeiger. Cascade decomposition using covers. In Michael A. Arbib, editor, *Algebraic Theory of Machines, Languages, and Semigroups*, chapter 4, pages 55–80. Academic Press, 1968.

[156] M. Zeleny. Autopoiesis: A theory of living organization. *New York*, 1980.

[157] T Zhang, P Brazhnik, and Tyson J J. Exploring mechanisms of dna-damage response: P53 pulses and their possible relevance to apoptosis, *Cell Cycle*, 6: 85–94. 2007.