Chapter 4
Biosystems and Bioinspired Systems

Abstract. Artificial genetic codes, neural networks and neural codes are presented as theoretical frames for evolutionary computation and biomimetic devices.

Models for genetic code evolution offer suggestions for chemical and biochemical inspired computations as for instance artificial chemistry or chemical programming.

Neural networks architecture issues require evolvability as outlined by growing neural nets or by protein based neural networks.

The significance of neural coding, symbolic connectionist hybrids, neural binding, temporal synchrony studies for unconventional computing and neurocognitive devices is highlighted.

Evolutionary circuits based on electrochemical filaments are proposed. The perspectives of evolvable circuits based on bio-molecules properties, are evaluated.

Case studies show how technological innovation should find the right moment to free the artificial system designer from the detailed experimental data of real systems.

4.1 Artificial Genetic Codes

4.1.1 Genetic Code Evolution

The main objective of the genetic code theoretical study in terms of PSM is to understand and make use of genetic code evolution scenarios as suggestions for new computing and information technologies. To re-apply this understanding in developing new ways of study or explanations of biological relevance the for real genetic code, may be considered as a long-term objective only.

Evolutionary computation studies and evolvable devices may make use of biological principles but do not attempt to model or to mimic detailed data or

processes from real genomes. Bio-inspired artificial design is not constrained by high fidelity to the original natural complex system. Examples include genetic algorithms calculus inspired by Darwinian evolution and genetics, artificial neural networks and artificial neural codes inspired by neuroscience but not restricted to this.

The expanding code scenario from single-bases nucleotides to doublets and then to triplets that is to codons offer interesting suggestions for evolvability studies and applications.

Several hypothetic scenarios have been advanced to explain the genetic code structure and its origin (Weberndorfer et al. 2003, Koonin and Novozhilov 2009). The main concepts on origin and evolution of the code are the stereochemical theory, the co-evolution theory and the adaptive theory.

The stereo-chemical theories suppose that the specificity of a codon for a particular amino acid is based on a direct interaction of amino acid and nucleotides. Amino acids might have been binding directly to the codons when the code was established and such binding imposed the code. The co-evolution theory explains the non-randomness of the code by the fact that the code system is an imprint of the pre-biotic pathways of amino-acid formation. According to this theory the genetic code evolution reflects the relationship among amino acids and their biosynthesis. In an early code used fewer codons and amino-acids and then expanded to include new amino-acids arising from biosynthesis coded for by new codons, with the resulting code assigning similar codons to amino acids that are related by biosynthesis.

Adaptive codes theories attempted to explain the observed patterns in genetic code and its evolution by postulating optimality of the code. Adaptation theories state that selection pressure resulted in the emergence of a code optimized for some measure, such as for minimization of the physicochemical effects of single mutational or translational errors.

A notable approach in the study of genetic code evolution is the Eigen's work on hypercycles systems of mutually autocatalytic components. It consider the question of under what conditions, the system can self-organize to a dynamic stability (Eigen and Schuster1979). Eigen approach was based on the view that the self-organization including the development of hypercycles is a process that can occur in a homogeneous system by intrinsic necessity.

Eigen and Schuster (1979) considered that the primitive code may use units of less than three bases. During its early evolution, the code would have increased both the number of codons and the coded aminoacids and the present code would reflect the pattern of this historical expansion (Wilhelm and Nikolajewa 2004, Patel 2005, Wu et al. 2005).

In partial contrast with Eigen approach, in the view of H. Kuhn, understanding the origin of living systems is a particular engineering problem: to find a sequence of physicochemical stages, beginning with prebiotically reasonable conditions and leading to self-organization of matter and to systems equipped with a life like genetic apparatus (Kuhn and Waser 1981, 1994, Kuhn H and Kuhn C 2003).

All these theories suggest that the genetic codes are information communication system that should reflect the physico-chemical properties of the amino acids. The different theories are not mutually exclusive and probably the code was shaped by a compromise of several scenarios (Ardell and Sella 2002).

4.1.2 Model for Code Evolution

4.1.2.1 Genetic Code

The genotype of cells is laid down in a linear sequence of four nucleotides: A-adenine, C-cytosine, U-uracil and G-guanine. The genetic information is transcribed in mRNA used as instructions for protein translation. Translation requires a mapping of the four nucleotides in 20 amino acids. Triplets of the four different RNA bases are read sequentially from mRNA. DNA is transcribed to mRNA that makes use of an RNA adaptor, tRNA to interpret nucleotides in amino acids. The four bases C, G, U and A, might form 64 different simple triplets patterns, the so-called codons. The 20 amino acids and the start and stop signals are coded redundantly by these 64 codons (Alberts et al. 1994).

There are some symmetry elements in genetic code. The symmetry supported the use of algebraic frames to characterize the genetic code. It has been suggested that the overall layout of the code can be accurately described in the algebra of group theory or of fields (Danckwerts and Neubert 1975, Findley et al. 1982, Jimenez-Montano et al. 1996, Jimenez-Montano 1999). These symmetries may be of help in explaining regularities and periodicities as observed in proteins sequences. They have been correlated to the possible evolution scenarios of the genetic code.

The relevant group to describe the symmetries of the bases {C, G, U, A} should be a group of order 4. There are only two possibilities for the group structure, the cyclic group C (4) and the group associated to the Galois field, GF (4). This is the so called Klein 4-group.

Several codes can be associated to the genetic code according to the order of importance for bases and of their positions in codons.

For triplets or codons the ranking: position 2>position 1>position 3 in establishing the amino-acid is acknowledged (Perlwitz et al. 1988).

One of the proposed nucleotide hierarchical ordering is: C>G>U>A. This hierarchy was established starting from the observation that C, in position 2 in codon, is anytime able to be source of a single amino acid. G is able to determine the amino acids in majority of cases, U only in some cases and A never. In other words, C base passes any time a single message while U, and A are credited with at least double message. G passes stronger messages than U or A, concerning the coded amino acid.

It is possible to associate to any base in codon a two-digit vector: (hydrogen bonds, chemical nature). The first digit refers to hydrogen bonds and the

second to the chemical nature. We will use first digit "1" for high number of hydrogen bonds that is for G and C and second digit "1" for chemical nature pyrimidines that is for C and U.

We will use first digit "0" for low number of hydrogen bonds that is for A and U and second digit "0" for chemical nature purines that is for A and G.

In this way we may describe the basis by the two-digit vectors:

C: (1, 1), G: (1, 0), U: (0, 1), A: (0, 0)

This corresponds to the hierarchy: C>G>U>A, and to the real numbers 3, 2, 1, and 0 associated to C, G, U, and A respectively. More exactly:

0=(0, 0), 1=(0, 1), 2=(1, 0), 3=(1, 1).

Of course, restricting the nucleotide characterization to only two properties: (hydrogen bonds, chemical nature) is a drastic simplification.

4.1.2.2 Expanding Genetic Code

The WE model (Sect. 2.2.2) highlights some particularities of genetic code Tables, the order of amino-acids availability and in this way, it may be of interest for evolutionary molecular devices and evolutionary computation.

Several scenarios for genetic code development from primitive bases will be considered in terms of WE model.

Table 4.1 represents the sum Y=Z⊕T, resulting as a solution of WE, as shown in Chapter 2. It is the sum for GF (4) or Klein 4 group relevant for genetic code.

The cyclic group C (4) offers a different solution. Z and T are identified with their two digit expressions for bases or equivalently with 0, 1, 2 and 3.

The development is supposed to start with all the four bases, C, G, U, and A.

This situation corresponds to the Table 4.1 for one dimensional Z and T.

Table 4.1 with GF (4) Klein-4 group table structure is well known in genetic code study.

Since the Table 4.1 contains 4 nucleotide of each type and the coding accomplished by nucleotides should allow conflict free access to parallel memory we will limit the Table 4.1 to vectors containing only distinct elements. They represent particular solutions of WE at specified levels in development.

An example is the vector Y(T) = y_0 = (C, G, U, A).

Table 4.1 Matrix of singlets

A	U	G	C
U	A	C	G
G	C	A	U
C	G	U	A

A specific folding operation allows rewriting y_0, as the Table 4.2.

There are only four elements and it is possible to represent y_0 as a 2x2 matrix like in Table 4.2.

Table 4.2 Single bases y_0

G	A
C	U

Other types of folding and other 2x2 matrices may be considered too. The folding with A and U interchanged was also studied. Various folding algorithms and their significance for genetic programming have been described by Banzhaf (1993).

It may be supposed that initially only one nucleotide is able to form doublets, then triplets and that this founding molecule is guanine G (Hartman, 1975).

It results the situation shown in Table 4.3. The codification is accomplished by GC, GG, GA and GU. These doublets corresponds to the amino-acids: Ala, Gly, Asp, Glu and Val.

Table 4.3 Two-bases matrix

GG		GA
	G	
GC		GU

Table 4.4 is the matrix-like presentation of product of two identical vectors that is

$Y(T) = y_0 \times y_1$ with $y_0 = y_1$

This includes the elements in Table 4.2.

Table 4.4 Doublets $y_0 \times y_1$

GG		AG		GA		AA
	G				A	
CG		UG		CA		UA
GC		AC		GU		AU
	C				U	
CC		UC		CU		UU

At this stage some amino acids resulted as follows: Pro, Ser, Leu, Thr, Arg coded without any ambiguity by CC, UC, CU, AC and CG.

The remaining amino acids will be coded by triplets showing multiple codifications.

They are: (Phe and Leu), (Ile and Met), (Cis, Trp and Stop), (His and Gln), (Tyr and Stop), (Ser and Arg) and (Lys and Asn).

They are coded by UU, AU, UG, CA, UA, AG and by AA respectively.

In Table 4.4 the eight "strong" double-bases: CC, GC, CG, GG, GU, CU, UC and AC are intertwined with the eight "weak" double-bases: AA, UA, AU, UU, UG, AG, GA and CA. The mirror symmetry G↔A and U↔C relative to median Y-axis is obvious.

The new letters, A, C, U, G have been put adjacent to the first two ones to the left side or right side. In this way the significance order corresponds to the order of letter acquisition.

Table 4.5 is a product of three 2x2-matrices-like tables,

$Y(T) = y_2 \times y_0 \times y_1$ with $y_0 = y_1 = y_2$

The new letters have been put adjacent to the first two, to the right side.

A version of the genetic table with 64 codons is resulting. Many other artificial genetic code Tables may be obtained by changing the hierarchy for codons, the developmental rules, the initial set of bases, the position of concatenation, and so forth Notice also that there are two groups of order 4, the Klein group and the cyclic group.

The hypothesis concerning the right or left concatenation of solutions is related to other proposed evolution scheme. For instance it was suggested by Wu et al. (2005) that triplet codons gradually evolved from two types of ambiguous doublet codons, those in which the first two bases of each three-base window were read ("prefix" codons) and those is which the last two bases of each window were read ("suffix" codons).

The right or left concatenation of solutions is correlated also to the reverse recognition conjecture of Nikolajewa et al. (2006).

Table 4.5 contains the 64 codons grouped in 4 large quarters each with a common base in the center (C, G, U, and A), each formed by 16 codons. Each large quarter contains 4 new quarters for doublets with the central base in second position and finally 16 codons since each doublet is the center of a new quarter with the doublets in the first two positions. Table 4.5 is similar to the conventional genetic code Table (Alberts et al. 1994). Instead of representing the codons with a central nucleotide on a column the codons associated to a central nucleotide C, G, U or A may be found in a quarter. Similar Tables have been discussed by other authors (Jimenez et al. 1996, Benyo et al. 2004).

For presentation purposes it is easier to portray the codons in plane that is by Tables than by hyper-cubes. However the modes of presentations are equivalent.

According to the above analysis there are several stages in the development of the genetic code. To these corresponds stages of amino-acids availability in the order:

- 1^{st} stage: [Ala, Gly, Asp, Glu and Val]
- 2^{nd} stage: [Pro, Ser, Leu, Thr, Arg]

Table 4.5 Triplets. Codons-matrix, $y_2 \times y_0 \times y_1$

GGG		GGA		AGG		AGA	GAG		GAA		AAG		AAA
	GG				**AG**			**GA**				**AA**	
GGC		GGU		AGC		AGU	GAC		GAU		AAC		AAU
			G							**A**			
CGG		CGA		UGG		UGA	CAG		CAA		UAG		UAA
	CG				**UG**			**CA**				**UA**	
CGC		CGU		UGC		UGU	CAC		CAU		UAC		UAU
GCG		GCA		ACG		ACA	GUG		GUA		AUG		AUA
	GC				**AC**			**GU**				**AU**	
GCC		GCU		ACC		ACU	GUC		GUU		AUC		AAU
			C							**U**			
CCG		CCA		UCG		UCA	CUG		CUA		UUG		UUA
	CC				**UC**			**CU**				**UU**	
CCC		CCU		UCC		UCU	CUC		CUU		UUC		UUU

- 3^{rd} stage: [(Phe and Leu), (Ile and Met), (Cis, Trp and Stop), (His and Gln), (Tyr and Stop), (Ser and Arg) and (Lys and Asn)]

This temporal order is not so far from that resulting from co-evolution theory (Wong 1975).

A first stage groups the aminoacids Ala, Gly, Asp and Glu. The connection Ala, Val is not presented in co-evolution theory. The next stage involves Pro, Ser, Thr, Arg in both theories. The difference refers to Leu and is a consequence of the missing connection Ala,Val for the first stage.

According to the theory of Eigen and Winkler-Oswatitsch (1981) the first amino acids were Gly, Ala, Asp and Val a result confirmed by some classical experiments.

Kuhn and Waser (1994) selected as plausible steps in the evolution of genetic code Gly Ala,Val, Asp, Glu followed by a class containing Leu, Ile, Ser, Thr, Lys, then Arg, Gln, Asn, then Pro and so on.

The above results may be compared with consensual chronology of amino acids (Trifonov 2000). Trifonov presents the codon chronology as follows:

Gly, Ala, Val, Asp, Pro, Ser, Glu, Leu, Thr, Asn, Arg and so on.

4.1.3 Codons and Amino Acids

Each codon, codes for the introduction of a specific amino acid into a growing protein, a process that involves recognition of the anti-codon sequence.

There exists a relationship between the codons and the properties of the coded amino acids. To outline this relationship it is possible to associate to any base in codon a two-digit vector. The Table 4.6 contains the vectors associated to codons as well as the coded amino acids.

For the Table 4.6 the associated vector is in the significance order for bases is:

position2>position1>position3.

For example AUC will be replaced by 010011 corresponding in succession to U:(01) (position 2), then A:(00) (position 1) and then C:(11) (position 3). This is in fact the supposed evolutionary pathway for development.

Codons area as outlined by Table 4.6 is again in general agreement with some assertions of the co-evolution theory (Ronneburg et al. 2000). In the evolutionary map of the genetic code based on precursor product pairs the founding precursor is Ala coded by GC. Close to this there are Ser coded by UC and Gly coded by GG. This can be inferred from the Table 4.6 too. Relatively far from the Ala precursor in evolutionary map there are Asn and Lys coded by AA or Leu and Phe coded by UU, Tyr coded by UA, Met coded by AU. This is obvious from Table 4.6 too.

Table 4.6 may be of use in the context of stereo-chemical theory. This theory assumes that the physical and chemical properties of a given amino acid are related to the nature of the codons. If the codons are similar the amino acids will be similar and reverse.

Moreover similar amino acids might replace each other. This assumption is in agreement with those theories that place specific constraints on the assignment of codons to amino acids. For example a significant correlation was observed between hydrophobic ranking of the amino acids and the hydrophobic character of the anti codons.

4.1.4 Polypeptides

A graphical illustration of the polystochastic framework is presented here. The Table 4.7 reproduces the genetic code Table and includes free places for amino acids.

Some examples clarify the PSM framework for polypeptide synthesis.

Elements of the SKUP are emphasized by Table 4.7.

Denote the singlet conditioning level by m=0, the doublet level by m=1 and the triplet level by m=2.

The coding for amino acids may be done by doublets or by triplets.

S, K, U and P will be vectors denoted as follows: S = (s^1, s^2); K = (k^1, k^2); U = (u^1, u^2); P = (p^1, p^2). Upper index refers to levels while lower index will refers to time step.

We start with an m=1 example. It is known that CU coding for Leu may serve as start.

Let CU=k_0^1, UC=k_1^1, CC=k_2^1, GC=k_3^1 UG=k_4^1. The upper index refers to level while the lower index refers to the time step. The states and the conditions at the level m=1 are represented in Table 4.7 by high thickness border cells.

Correspondingly the states at the second level will be: s_0^1 = Leu, s_1^1 = Leu.Ser, s_2^1 = Leu.Ser.Pro, s_3^1 = Leu.Ser.Pro.Ala, s_4^1 = Leu.Ser.Pro.Ala.

Table 4.6 Triplets vectors and amino acids

GGG 101010 Gly		GGA 101000 Gly		AGG 100010 Arg		AGA 100000 Arg		GAG 001010 Glu		GAA 001000 Glu		AAG 000010 Lys		AAA 000000 Lys
	GG				**AG**				**GA**				**AA**	
GGC 101011 Gly		GGU 101001 Gly		AGC 100011 Ser		AGU 100001 Ser		GAC 010011 Asp		GAU 010001 Asp		AAC 000011 Asn		AAU 000001 Asn
			G								**A**			
CGG 101110 Arg		CGA 101100 Arg		UGG 100110 Trp		UGA 100100 Stop		CAG 001110 Gln		CAA 001100 Gln		UAG 000110 Stop		UAA 000100 Stop
	CG				**UG**				**CA**				**UA**	
CGC 101111 Arg		CGU 101101 Arg		UGC 100111 Cys		UGU 100101 Cys		CAC 001111 His		CAU 001101 His		UAC 000111 Tyr		UAU 000101 Tyr
GCG 111010 Ala		GCA 111000 Ala		ACG 110010 Thr		ACA 110000 Thr		GUG 011010 Val		GUA 011000 Val		AUG 010010 Met		AUA 010000 Ile
	GC				**AC**				**GU**				**AU**	
GCC 111011 Ala		GCU 111001 Ala		ACC 110011 Thr		ACU 110001 Thr		GUC 011011 Val		GUU 011000 Val		AUC 010011 Ile		AUU 000101 Ile
			C								**U**			
CCG 111110 Pro		CCA 111100 Pro		UCG 111001 Ser		UCA 110100 Ser		CUG 011110 Leu		CUA 011100 Leu		UUG 010110 Leu		UUA 010100 Leu
	CC				**UC**				**CU**				**UU**	
CCC 111111 Pro		CCU 111101 Pro		UCC 110111 Ser		UCU 110101 Ser		CUC 011111 Leu		CUU 011101 Leu		UUC 010111 Phe		UUU 010101 Phe

The operator U, associated to this one level process is: $\mathrm{u}^1\ (s_0^1, k_1^1) = s_1^1$.

Here s_1^1 is the amino acid associated to the codon k_1^1 coupled to previous chain of amino acids. Observe that: k_0^1, k_1^1, k_2^1, k_3^1, k_4^1 is a trajectory in the K space, while: s_0^1, s_1^1, s_2^1, s_3^1, s_4^1 is a trajectory in S-space.

Consider now examples of level m=2 of coding evolution. The states and the conditions at this level are indicated in Table 4.7 by medium thickness border cells. Suppose that the construction starts at AUG. The codon AUG codes for methionine and serves as an initiation site. This is the initial condition k_0^2 = AUG. The associated state is methionine that is s_0^2 = Met.

Then the trajectory may evolve towards the condition k_1^2 = UAC.

This corresponds to the amino acid Tyr. The new state s_1^2 is the succession Met.Tyr.

Then the new condition may be k_2^2=UAG. This is a terminal codon. Hence $s_2^2 = s_1^2$.

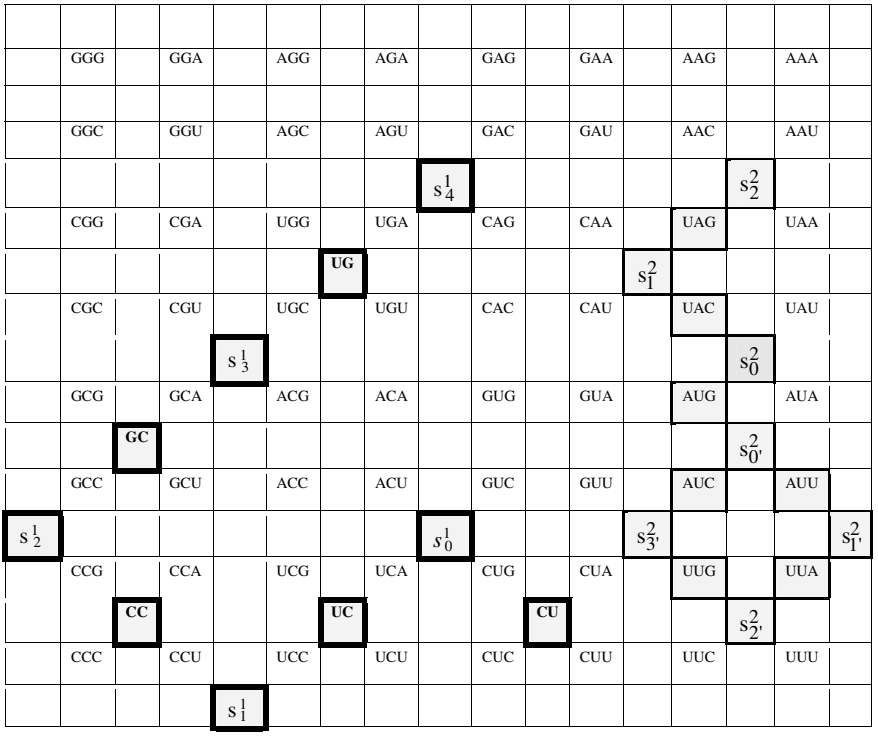
Observe that s_1^2 depends on s_0^2 and on k_1^2, s_2^2 depends on s_1^2 and on k_2^2.

The operator u associated to this one level process is: $\mathrm{u}^2\ (s_0^2, k_1^2) = s_1^2$.

Here s_1^2 is the amino acid associated to the codon k_1^2 coupled to previous chain of amino acids.

Possibility as $\mathrm{p}\ (k_1^2 \mid s_0^2)$ depends on the fact that it is a genetic transition (purine to purine or pyrimidine to pyrimidine) or a genetic transversion

Table 4.7 Codons and amino acids schema



(purine to pyrimidine or vice-versa). Recall that A, G are purines while C, U are pyrimidines. Examples of possibilities are the similarities as defined in Sect. 2.3.1.

Periodicities may arise as for instance in the example at level m=2: $k^2_{0'}$=AUC.

Then $k^2_{1'}$=AUU, $k^2_{2'}$=UUA, $k^2_{3'}$=UUG.

The conditions and states trajectories are outlined by medium thickness border cells. More large excursions in the Table 4.7 may be considered and "junk" steps may be very frequent. Simulation of process as exemplified serves to illustrate the proposed code evolution scenario.

4.1.5 Basic Framework Evaluation

Life involves a semantically closed organization between symbolic records and dynamical constraints (Pattee 1995). Symbols, as discrete functional switching-states, are seen in all evolvable systems in form of genetic codes, and at the core of all neural systems in the form of informational mechanisms that switch behavior DNA molecules represent the symbolic aspects

here, that is, the genome. This corresponds to the conditions K, in SKUP framework. The dynamic material aspects are represented by the phenotype that is by proteins, organisms and eventually by the environment. This corresponds to states S. The genome generates different dynamical systems that promotes their stability and survive and in that way serves as seeds of a generally evolvable system.

The genome may be interpreted as a possible solution of the wave model.

It is an apparently "time-less" model since the time T is defined on a finite group and has a cyclic character. As discussed by H. Kuhn this temporal cycling is crucial for genetic code emergence and evolution. Dynamical model with usual real time, characterizing the kinetic equations completes the evolvable system description.

The closure between symbolic that is digital and real aspect of the closure are clearly illustrated in the PSM framework. The operator U may be associated to tRNAs. The tRNAs performs decoding activities. It incorporate two codes: one to read the info from mRNA and a second code that determines the amino acid with which the tRNA is loaded. This outlines the role of operator U in the transition from discrete symbol to real material aspects. U correlates informational and chemical data.

The enzymes as RNA-replicase may be associated to possibilities P. It performs encoding activities. On the primitive genetic code, the tRNAs and RNA replicases could have been involved but other closure possibilities exists.

PSM framework illustrates a minimal closed organism with translation (Webernsdorfer et al. 2003). It has a genome that carries genes for a protein replicase and tRNAs, a translation apparatus, and system loading tRNAs with amino acids.

P and U correspond to the so-called upward and downward causation respectively (Pattee 2000). For the life and artificial life, AL, situations, the semantic genetic control can be viewed as up-down causation, while the dynamics of organism growth controlling the expression of the genes can be viewed as down-up causation. The closure concept is an essential relation of these causations. Semantic closure is limited to two levels, denoted here by K and S.

The interplay between the WE in the so-called sequence space and the more or less similar real valued equations of thermodynamics and chemical kinetics represents the specificity of living systems. Obviously the closure mediated by the operators U and possibilities P is compatible with both co-evolution and with stereo-chemical theories (Webernsdorfer et al. 2003).

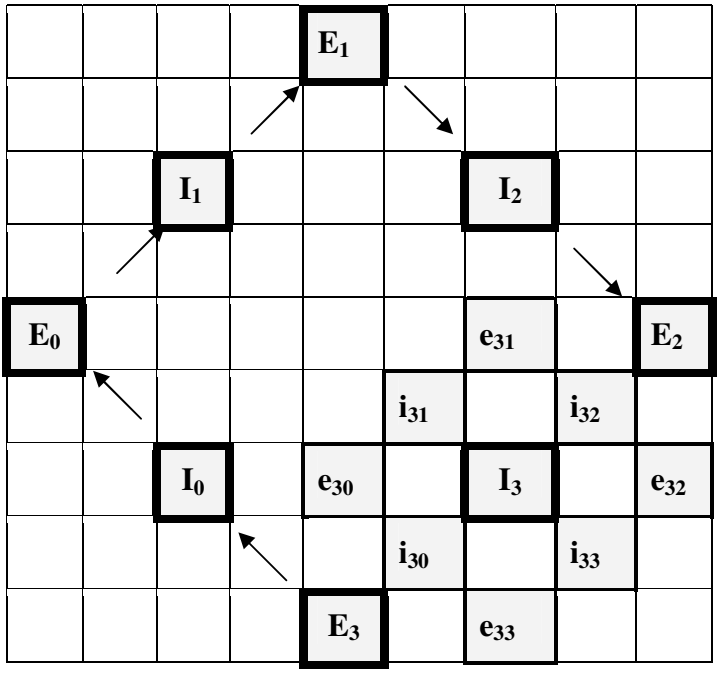
Table 4.8 illustrates the SKUP schema associated to hypercycles.

The relation of Table 4.8 to WE is similar to that outlined for Table 4.7.

In this case the conditions K are associated to RNA while the states S to enzymes.

The hypercycle is a self reproducing macromolecular in which RNAs and enzymes cooperate. There are RNA matrices (I_i), the i-th RNA codes the i-th enzyme E_i. The enzymes cyclically increase RNA's replication rates, namely,

Table 4.8 Schema for hypercycles



E_0 increases replication rate of I_1, E_1 increases replication rate of I_2, E_2 increases replication rate of I_3, and E_3 increases replication rate of I_0. The hypercycle is represented in Table 4.8 by high thickness border cells.

The mentioned macromolecules cooperate to provide primitive translation abilities, so the information, coded in RNA-sequences, is translated into enzymes analogous to the usual translation processes in biosystems.

The cyclic organization of the hypercycle ensures its structure stability. For effective competition, the different hypercycles should be placed in separate compartments.

Table 4.8 shows that some RNA may induce the reproduction of other metabolites in cyclic manner. Supposing that I_3 is in this situation, e_{30} increases replication rate of i_{31}, e_{31} increases replication rate of i_{32}, e_{32} increases replication rate of i_{33}, and e_{33} increases replication rate of i_{30}. The secondary cycle is represented in Table 4.8, by high thickness border cells.

The number of RNAs in each cycle may vary.

The wave model WE, characterizes the genetic bio-chemical reactor in a discrete space. It includes the "convection" or "drift" term $V \otimes \frac{\partial Y}{\partial Z}$ and the "kinetic" term Q⊗Y.

Observe that just one wave equation replaces the entire system of differential equations for quasispecies (Eigen and Schuster 1979).

This WE is adequate for highly non-linear processes modeling. The time T is a more natural expression for time to record qualitative developments than the usual linear time. The cyclic and diversified characters of environment, as described by Kuhn are accounted for by T and Z introduced here. The different values of T correspond to the developmental or pattern recognition stages.

Q takes into account the mutations and selections. The velocity V takes into account the "convection". It could happen that the convection contribution is more significant than that of mutations for evolution.

4.1.6 Perspectives

4.1.6.1 Three Realms Framework

Research within evolutionary computation has identified properties of biological coding that may be significant to evolutionary algorithms (Rocha 1997, Kargupta 2001, Suzuki and Sawai 2002). Applying computation results back to biology suggests that the genetic basis of life may enhance the power of natural mechanisms as selection as a search algorithm.

This approach offers a partial answer to the present need to elaborate common mathematical frames for evolvable systems.

Fig. 4.1 shows the categorical framework for the three levels of the refined central dogma of biology. According to the central dogma proteins are not made directly from genes but require an intermediary, and this intermediary is RNA.

Here S denotes the proteins level. For computing purposes, K1 and K2 are the two conditions levels. K1 is associated to RNA K2-is in this case the meta-level representing the DNA.

The strategies are defined at the level K2 since the information is transmitted from DNA to protein through RNA.

U10: K1 $\rightarrow$ S describes the translation. U21: K2 $\rightarrow$ K1 describes the transcription.

Implicit in the central dogma view is the idea of a unique mapping from gene to protein in which RNA plays only a mediatory role.

Some refinements of central dogma refer to the possibilities P01 and P12. P01: S $\rightarrow$ K1, and P12: K1 $\rightarrow$ K2 effects may be associated to the regulation processes.

For the operon model the DNA encodes two classes of proteins, structural and regulatory. It refers to a splitting of S in two non-interacting realms S1 and S2. Structural proteins play a functional role in the cell's metabolism. Regulatory proteins interact with DNA to control the rates of transcription of other genes. This links the proteins S to K2 realm (Jacob and Monod 1961).

Geard and Wiles (2003) evaluated the possibility of splitting of K1 in two functional modules corresponding to small RNA and standard RNA. The small RNA molecules may provide a kind of meta-level of evolution allowing for the evolution of new and complex functions by modulating the control

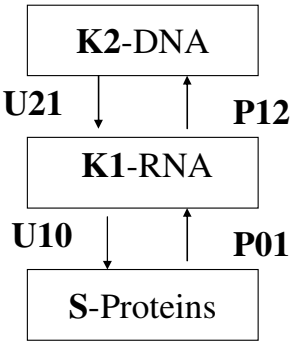


Fig. 4.1 Refined central dogma

architecture of a stable proteome. In this case the DNA should offer the meta-information.

Fig. 4.2 shows a theoretical model in which S-Proteins are supposed to play a regulatory role for both K1-RNA and K2-DNA levels.

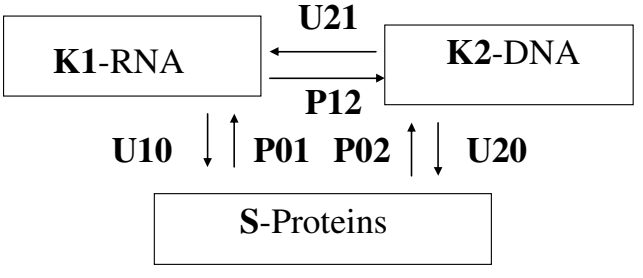


Fig. 4.2 Regulatory models

The categorical framework describes hypothetical interactions as follows:

U10: RNA → Proteins, U20: DNA → Proteins, P01: Proteins → RNA, P02: Proteins → DNA. The DNA is a code for RNA level.

The framework shown in Fig. 4.2 is an integrative closure and should correspond to a degree of evolvability and autonomy of the system.

4.1.6.2 Higher Order Genetic Code Hypothesis

It was observed that the genome may contain more information than it has been anticipated. This redundant information suggested investigating if there is a higher level genetic code that directs evolution (Caporale 1984).

An examination of the rate of codon substitution during gene evolution reveals that not all so-called silent mutations, that is, the mutations to another codon that signifies the same amino acid, behave as if they are neutral.

There appear to be constraints of codons selection so that in a given context two codons, although thought to be synonymous are not in reality equivalent. Another issue is the evidence that the so-called neutral third position in codons may also carry a message may be as important as specifying the amino acid.

Rejecting purely probabilistic mechanism of genetic variation is not a refutation but rather a higher understanding of the Darwinian theory of selection. Genomes that evolve efficient biochemical systems to navigate through the space of possible future genomes would be favored by natural selection and would allow adapting more quickly when confronted by environmental challenges.

Several other facts allowed the hypothesis of a fourth realm, controlling the significant modifications of codes, or the transitions between several codes.

A four realms framework may be considered for code evolution.

Fig. 4.3 illustrates this higher order genetic code hypothesis.

The signification of the functors U and possibilities P is explicit in Fig. 4.3.

U10: K1-RNA→S-Proteins, U21: K2-DNA→K1-RNA, U32: K3-Metacode →K2-DNA, U30: K3-Metacode→S-Proteins, P01: S-Proteins→K1-RNA, P12: K1-RNA→K2-DNA, P23:K2-DNA→K3-Metacode and P03: S-Proteins→ K3-Metacode.

Observe that: U30=U10oU21oU32 and P03=P01oP12oP23.

The DNA is a code for RNA level.

The new realm, K3-Metacode, is a kind of meta-meta-model that completes and closes the frameworks shown in Fig. 4.1.

K3 organizes the multiplicities of codes and their overlapping (Trifonov 1999).

K3 represents an efficient way to explore codes and may favor the strategies that increase the rate of adaptation.

The architecture shown in Fig. 4.3 offers suggestions for modeling genetic control hierarchies. Biological progress may be accelerated if models are formulated and applied for global genetic control structures.

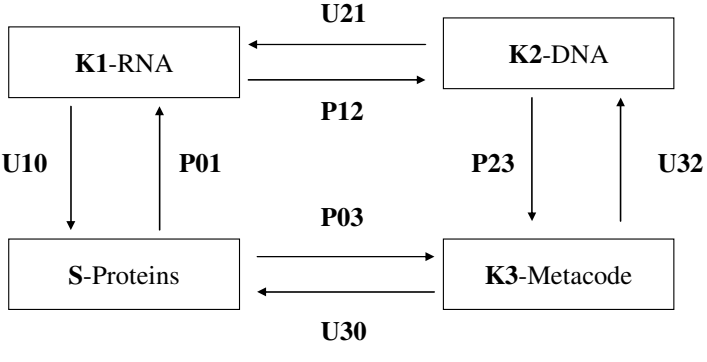


Fig. 4.3 Higher order genetic code hypothesis

The concept of evolution as a complex hierarchical process was illustrated by a framework similar to that shown in Fig. 4.3. Gould identified three levels or "tiers" of evolution (Gould 1985). The first tier selection is the conventional Darwinian selection of individual organisms and corresponds to K1. The second-tier selection corresponding to K2 emerges from differential speciation and extinction among lineages. The third tier selection reflects the infrequent catastrophes which may eliminate forms of life without respect to their adaptive or competitive advantage. It corresponds to K3 in Fig. 4.3.

Confirmed at least in part by the study of real biosystems such unconventional models offer in turn interesting suggestions for chemical and biochemical inspired computations and devices as for example in genetic or chemical programming (Keller and Banzhaf 1999, Matsumaru and Dittrich 2006).

4.1.6.3 Chemical Programming

The architecture of the chemical programming developed in organic computing studies outlines four realms similar to that shown in Fig. 4.3.

In this case the realms shown in Fig. 4.4 are as follows: S-Hardware, sensor, actuator, K1-ChemOS, chemical operating system, K2-ChemVM, chemical virtual machine, and K3-Compiler.

The compiler takes a high level description of a chemical program as input. The chemical program consists of a list of molecules and reaction rules including kinetic laws. The compiler generates chemical byte code which can be processed by the chemical virtual machine that is able to run a chemical program. Communication between the chemical program and other hardware such as sensors or actuators is handled by the chemical operating system.

The integrative closure refers to compiler and hardware relationship.

Taking bio-chemical information processing as an inspiration for organic computing is attractive since biochemical systems possess a number of desirable properties. However chemical programming does not aims to replace the current computing systems. For example, implementing a word

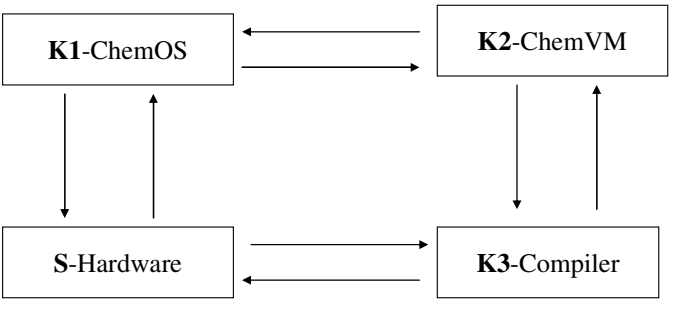


Fig. 4.4 Chemical programming framework

processor on a chemical basis is not feasible, and techniques for programming chemical-like technical systems are still missing. It is more likely that "artificial chemistry" or "artificial chemical engineering" will be integrated as subsystems together with other high-level computing concepts and methodologies (Matsumaru and Dittrich 2006). There exists applications were the molecular level analog computer may have distinct advantages.

4.2 Artificial Neural Networks

4.2.1 Architecture Problem

The domain of evolutionary computation is important for challenging applications as visual pattern classification, failure diagnosis, signal detection, sensor fusion, identification and control, planning and robotics, trading and so forth (Baeck et al. 1997).

The success and speed of training for neural networks NN is based on the initial parameter settings, weights, learning rates and architecture. NN simulation studies show that many complex problems cannot be solved by a learning algorithm in conventional fully connected layered NNs.

In spite of much research activity in the area of neural networks, NN, the design of architecture of NN was considered as less significant that the learning rules.

Evolutionary artificial neural network, EANN, refer to a class of artificial neural networks NNs, in which evolution is another fundamental form of adaptation in addition to learning. Evolutionary neural networks, EANN make use of evolutionary algorithm, EA, to improve NN architecture. EA are used to perform various tasks such as connection weight training, architecture design, learning rule extraction and adaptation (Yao 1999, Balakrishnan and Honavar 1995).

Evolutionary behaviour is considered as a preliminary method to confront complexity advent in computation. As a condition for evolutionary behaviour, EANN systems should be characterized by closure between the dynamics that is the phenotypic physical process of the material aspects of NN and the constraints that is the genotypic, syntactic rules or schemes of the symbolic aspects of the NN organization.

Evolutionary computational systems should be able to change both architectures and learning rules for automatic implementation without operator intervention.

The evolutionary artificial NN, the development models algorithms, are naturally represented in the PSM framework.

Innovative is the architecture indirect encoding based on the wave equation, WE. For genotypic, that is symbolic aspects of evolutionary systems the WE, generating the computation schemes or architectures is operational.

The novelty of the PSM framework lies also in hierarchical multiple scaled structure, in specificity of operations of decoding, from genotype to phenotype, and encoding, from phenotype to genotype.

The existing algorithms for evolutionary computations have many elements in common.

This follows the fact that they are inspired by the evolution theory in biology.

It would be beneficial to outline the common elements, to standardize the algorithms by applying the same generic framework to different complex problems.

The elements of a generic framework are that of the basic SKUP, that is the states S, the conditions K, the operators U and the possibilities P.

The state space S-represents the material aspects, the phenotypes, the realized experiments. For biological systems S corresponds to proteins, amino-acids, neurons.

The states changes are governed by dynamical laws and represent the factual aspects of the closure concept. The semantic closure concept refers to two levels architecture.

The conditions space K, describes genotypes, schemes, architectures.

For real biological system K is associated to DNA or to genome.

DNA contains the information needed by a biological organism to carry out its functions.

DNA represents the logic or informational part of the closure.

K is the space of genotypes. K elements may be indirectly specified by the wave model, WE. The WE solutions pertain to the set of conditions, K.

The genotypes are elements of a high dimensional search space. In conventional genetic algorithms, GA, the genotypes are binary strings of some fixed length, say n, that code for points in an n-dimensional Boolean search space. More generally a genotype can be considered as a string of genes.

Each genotype encodes for one or for a set of phenotypes. Such encodings employ genes that take on numerical values.

For SKUP framework U-is the operator associating to any element of K elements from S.

It is a decoding operator that produces the phenotypes corresponding to genotype and to previous phenotypes. This operator associates results, in S, to experiments designed in K.

Operators U may be stochastic or deterministic.

In real biological systems the genetic information written in DNA is translated in amino acids by means of a set of molecules known as amino acyl-tRNAs. These represent the biological interpretation of operators U.

The possibilities P-are related to the procedure to learn, to express information, to fitness evaluation. P facilitates the encoding process. This suggests that to the complementarity between K and S correspond a complementarity relation between U and P.

The phenotypes may be equipped with learning algorithms that train using evaluations on a given task. This evaluation of the phenotype determines the fitness of the corresponding genotype. In real biological systems P may be associated to RNA-replicase an enzyme that catalyzes the self-replication of RNA.

The generic SKUP framework elements for some of the existing evolutionary computation algorithms will be emphasized in what follows.

4.2.2 Graph Generation Grammar

Classical learning algorithm for NN aim at finding weights for an NN whose architecture is frozen. Considerable performances are resulting by modifying NN architectures and the learning rules. The application of evolutionary algorithms, EA, to neural network, NN illustrates the increasing interest in combining evolution and learning (Yao 1999).

The graph generation grammar, developed by Kitano (1990) combines a genetic algorithm, GA with an L-system (Lindenmayer 1968). The L-system is a rewriting formalism introduced to model the growth of plants and a neural net, NN to enable modeling of the development process (Appendix A7). The GA is used to acquire graph rewriting rules, for the graph L-system, instead of directly acquiring the NN network topology. The introduction of developmental stages is considered more plausible biologically and computationally efficient.

The developed graph generation grammar (Kitano 1990) contains rules of the form:

$$A \rightarrow \begin{matrix} B & C \\ D & E \end{matrix} \tag{4.1}$$

The left-hand-side LHS of the production rule is a symbol and the right hand side RHS is a matrix of symbols from an alphabet.

The terminal symbols are constant rules as for instance:

$$0 \rightarrow \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \quad 1 \rightarrow \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \tag{4.2}$$

After a specified number of steps in L-system the matrix symbols restrict to 0 or 1. The resulting matrix is considered as a connectivity matrix and as in direct encoding methods an NN graph is associated to this.

Subsequently the NN weights are modified by learning methods as the back-propagation, for example.

A new graph generation model is presented in what follows, outlining the generic PSM framework elements and based on WE.

In this generic PSM framework, the genome corresponds to the elements of K.

Starting from an alphabet the WE generates more elements of K.
Consider the WE model with Y=y_0y_1 and T=t_0t_1.
The wave equation WE model has among its convection type solutions:

$$y_0 = \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \tag{4.3}$$

$$y_1 = \begin{matrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{matrix} \tag{4.4}$$

The general solution may be the categorical product Y=$y_0 \times y_1$:

$$Y = y_0 \times y_1 = \begin{matrix} 00 & 01 & 02 & 03 & 10 & 11 & 12 & 13 \\ 01 & 02 & 03 & 00 & 11 & 12 & 13 & 10 \\ 02 & 03 & 00 & 01 & 12 & 13 & 10 & 11 \\ 03 & 00 & 01 & 02 & 13 & 10 & 11 & 12 \\ 10 & 11 & 12 & 13 & 00 & 01 & 02 & 03 \\ 11 & 12 & 13 & 10 & 01 & 02 & 03 & 00 \\ 12 & 13 & 10 & 11 & 02 & 03 & 00 & 01 \\ 13 & 10 & 11 & 12 & 03 & 00 & 01 & 02 \end{matrix} \tag{4.5}$$

The matrix Y contains elements of K. To this matrix, a connectivity matrix CM and an NN is associated in different ways.

Suppose for instance, that instead of the double digit vectors (ij) in Y we put 0 if the difference between digits is equal or higher than 2 and 1 if the difference is lower or equal to 1.

The rule may be:

$$u(ij) = \begin{matrix} 1 & i - j \leq \pm 1 \\ 0 & i - j \geq \pm 2 \end{matrix} \tag{4.6}$$

Using this rule, the connectivity matrix CM is resulting instead of Y and an NN may be associated to this matrix. In this particular case the neuron 1 is connected with 2, 5, 6 and 7, the neuron 2 is connected with 1, 4, 5, 6 and 8 and so on.

$$CM = \begin{matrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{matrix} \tag{4.7}$$

Elements of generic PSM framework are identifiable and easy to be compared with those of Kitano model.

The conditions K may include the initial alphabet and the matrices resulting as WE solutions.

The grammar rules are based on concatenation as for instance:

$$0 \rightarrow \begin{matrix} 00 & 01 & 02 & 03 \\ 01 & 02 & 03 & 00 \\ 02 & 03 & 00 & 01 \\ 03 & 00 & 01 & 02 \end{matrix} \quad 1 \rightarrow \begin{matrix} 10 & 11 & 12 & 13 \\ 11 & 12 & 13 & 10 \\ 12 & 13 & 10 & 11 \\ 13 & 10 & 11 & 12 \end{matrix} \tag{4.8}$$

These are resulting by categorical product "×" operations used in WE solutions.

The WE model includes grammar as that used in Kitano model.

The operator U-determine the rule translates a matrix like Y in the connectivity matrix CM. It is a specific rule u, associated to this translation.

The states S- may be identified with the NN weight of the connections associated to NN.

The possibilities P, give the learning rules for weights.

Table 4.9 contains the solutions Y = y_0 and Y = $y_0 \times y_1$ presented together (equations 4.3 and 4.5). Selected states are bolded and underlined in Table 4.9.

Table 4.9 WE solutions at m=0 and m=1. Selected states

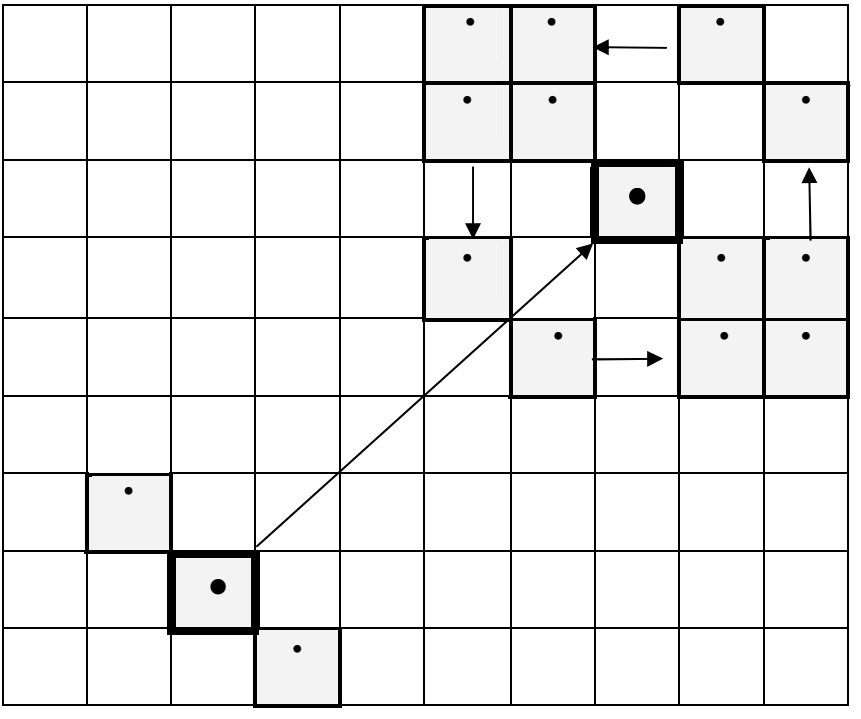
00	01		02	03	**10**	**11**		**12**	**13**
01	02		03	00	**11**	**12**		**13**	**10**
		0					**1**		
02	03		00	01	**12**	**13**		**10**	**11**
03	00		01	02	**13**	**10**		**11**	**12**
10	11		12	13	00	01		02	03
11	**12**		**13**	10	01	02		03	00
		1					0		
12	**13**		**10**	11	02	03		00	01
13	10		11	12	03	00		01	02

Applying for the area selected the rule u, the Table 4.10 results.

It is supposed that only "1" represents a large node "•" at the level m=0 and a smaller node "•" at the level m=1. Table 4.10 shows the cells resulting after the application of the rule u and area selection.

High border thickness outlines the level m=0 while medium border thickness outlines the level m=1. Next steps in NN development should establish connections between different nodes. The superposed arrows show an oriented NN.

Table 4.10 Associated NN



4.2.3 Cell Space Encoding

4.2.3.1 Growing Neural Networks

Several indirect encoding methods are inspired by the development and morphogenesis biological processes. An illustrative example is the algorithm due to Nolfi and Parisi (Nolfi and Parisi 1995a, 1995b). This algorithm presents a method for encoding NN architecture into a generic string, suggested by the real neural development. Inherited genetic material specifies developmental instructions that control the axonal grow and the branching process of a set of neurons. The neurons are encoded with coordinates in a two-dimensional space.

The neurons located in a 2D space are associated to the space K of conditions in generic PSM framework. K is the set of initially established neurons or cells.

The dendrite growing is part of the states S-process. For different neurons or cells, the GA mechanism induces a tree as a random walk in the space S. If the growing axonal branch of a particular neuron reaches another neuron a connection is established.

The resulting NN contains only the completely connected neurons and the branches.

The temporal dimension of developmental process was taken into account in the Nolfi and Parisi models. Several time scales may be naturally considered in WE study.

The NN architecture was tested for specific problem as for instance for pattern recognition.

Then according to fitness criteria S is modified. This is the learning process associated to possibilities P.

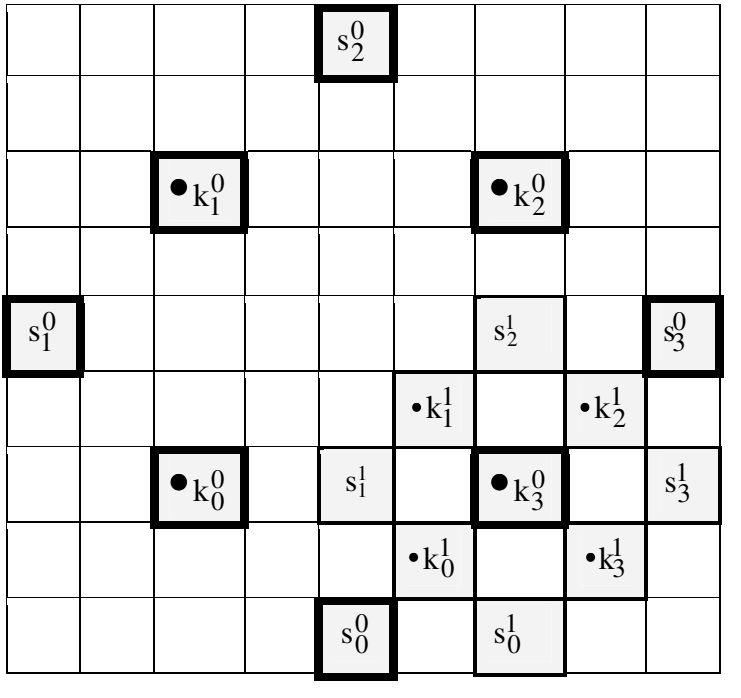
4.2.3.2 Schema for Cell Encoding Algorithm

The SKUP framework for new cell space encoding model is presented in what follows. The hierarchy of neurons will be considered. Here the large dots "•" denotes the level m=0, while smaller dots "•" denotes the level m=1 (Table 4.11).

For two level evolution, m=0, m=1 the quadruple SKUP consists of the vectors

$S = (s^0, s^1)$; $K = (k^0, k^1)$; $U = (u^0, u^1)$; $P = (p^0, p^1)$

Table 4.11 Schema for cell space encodings



Let: k_0^0, k_1^0, k_2^0 and k_3^0 denotes the possible conditions at the level m=0. The upper index refers to levels while the lower index refers to the time step. It should be emphasized that the time steps at different levels may be different and this is a key feature for evolutionary behavior. The states and the conditions at the level m=0 are represented by high thickness border cells in Table 4.11.

The system initial state is s_0^0. With possibility $\mathrm{p}^0(k_0^0|s_0^0)$ the condition k_0^0 is selected. Based on this, the operator $s_1^0=\mathrm{u}^0(k_0^0,s_0^0)$ allows the transition to the new state s_1^0. Then with possibility $\mathrm{p}^0(k_1^0|\, s_1^0)$ the new condition, k_1^0 arises. In the new condition, the operator $\mathrm{u}^0(k_1^0,\, s_1^0) = s_2^0$ allows the system to reach the state s_2^0.

Observe that: $s_1^0=\mathrm{u}^0(k_0^0,\, s_0^0)$ implies $s_2^0=\mathrm{u}^0\ (k_1^0,\, \mathrm{u}^0(k_0^0,\, s_0^0))$.

With possibility $\mathrm{p}^0(k_2^0|s_2^0)$, the condition k_2^0 is selected and finally the new state results $s_3^0=\mathrm{u}^0(k_2^0,\ s_2^0)$ results. Observe that $s_3^0=\mathrm{u}^0(k_2^0,\ \mathrm{u}^0(k_1^0,\ \mathrm{u}^0(k_0^0,\ s_0^0)))$.

The states are resulting not necessarily in a recursive way since, in practical cases the operators may varies with the step.

The possible states at the level m=0 are: s_0^0, s_1^0, s_2^0, s_3^0, s_4^0. The interpretation of the higher-thickness border cells trajectory is the process description as follows: from the state s_0^0 through condition k_0^0 towards the state s_1^0, then through condition k_1^0 towards the state s_2^0, and so on.

The net development may be continued at the level m=1 for different new conditions $k_0^1, k_1^1, k_2^1, k_3^1$.

The states and the conditions at the level m=1 are represented by medium thickness border cells. The system initial state at the level m=1 is s_0^1. With possibility $\mathrm{p}^1(k_0^1|s_0^1)$ the condition k_0^1 arises. Based on this, the operator $\mathrm{u}^1(k_0^1,\, s_0^1) = s_1^1$ describes the transition to the new state s_1^1. Then with possibility $\mathrm{p}^1(k_1^1|\, s_1^1)$ the new condition, k_1^1 arises. In the new condition, the operator $\mathrm{u}^1(k_1^1,\, s_1^1)=s_2^1$ allows the system to reach the state s_2^1.

Observe that: $s_2^1=\mathrm{u}^1(k_1^1,\, \mathrm{u}^1(k_0^1,\, s_0^1))$ and $s_3^1=\mathrm{u}^1(k_2^1,\, \mathrm{u}^1(k_1^1,\, \mathrm{u}^1(k_0^1,\, s_0^1)))$.

The states at the level m=1 are: s_0^1, s_1^1, s_2^1, s_3^1. The conditioning at the level m=1 is represented by the loop: k_0^1, k_1^1, k_2^1, k_3^1.

The interpretation of the standard border thickness trajectory is as follows: from the initial state s_0^1 through condition k_0^1 to the state s_1^1, then through condition k_1^1 to the state s_2^1, and so on.

The elements of generic PSM framework are clearly indicated here.

The conditions are $\mathrm{K} = (\mathrm{k}^0, \mathrm{k}^1)$ were k^0 corresponds to large neurons "•" while k^1 corresponds to smaller neurons "•"

The states $\mathrm{S} = (\mathrm{s}^0, \mathrm{s}^1)$ are the NN. There are two types of NN here and this allows evolutionary computation. This corresponds to a decomposition of the problem in sub-problems more easy to solve.

Rules for $\mathrm{U} = (\mathrm{u}^0, \mathrm{u}^1)$ and $\mathrm{P} = (\mathrm{p}^0, \mathrm{p}^1)$ depend on the studied case.

4.2.4 Perspectives

4.2.4.1 Hierarchy of Prediction Layers

From the point of view of a natural agent the external environment does not provide any direct indication on how the agent should act to attain a given goal. The environment provides a large number of data, the sensory states. The system should be able to extract regularities from time series through prediction learning.

Nolfi and Tani (1999) shows that the ability to extract regularities from data can be enhanced if we use a hierarchical architecture in which higher layers are trained to predict the internal state of lower layers when such states change significantly.

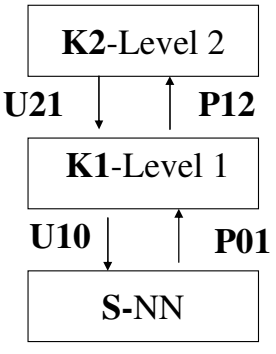


Fig. 4.5 Hierarchy of prediction layers

Fig. 4.5 shows the architecture with three levels one corresponding to sensor states and two prediction levels.

For this type of architecture the sensory information will be progressively transformed going from lower to higher levels.

S denotes the environment that is the NN nodes.

K1 and K2 are the two prediction levels. K1-represents the basic level while K2-is the meta-level.

A change in K2 has higher leverage because it represents multiple changes at lower level K1. U10: K1→S describes the decoding and actions, P01: S→K1, describe the fitness evaluation for S. The information change between the basic level and the meta-level is characterized by the operator U21 and the possibilities P12.

The first prediction level K1 predicts the states of the sensors by receiving as input their state in the previous time step and the planned action. The higher level K2 predicts the internal state of the lower level K1 by receiving as input their previous state.

By training an architecture of this type to predict the next sensory state of an animate navigating in an environment divided into two rooms Nolfi and Tani (1999) showed how the first level prediction layer extracts low level regularities such as walls and corners while the second level prediction layer extracts higher level regularities such as the left side wall of the large room.

That prediction learning can extract higher level regularities from time series was shown by Elman studies of languages (Elman 1990, 1993). He showed how by training a simple recurrent NN to predict the next word in sentences of a language the network was able to extract high level regularities for words such as nouns or verbs.

The architecture shown in Fig. 4.5 lacks some elements of a potential top level K3 and also the integrative links between the level, K2 or K3 and environment S.

The missing links induce limitations and prevents this architecture to become evolvable and to detect regularities for changing environments as for instance the presence of a new object for animates or of new words for language.

4.2.4.2 Structured GA

It was recognized that genetic algorithms work well in some cases and not in others, but it is not yet clear why this happens.

To address some of the difficulties encountered by the traditional GA in problem solving Dasgupta and Mc Gregor (1992) introduced a two-level structure for the chromosome in genetic algorithms.

The structured GA is based on hierarchical genomic structure and a gene activation mechanism in its chromosome. Genes at different level can be either active or passive. Higher level genes activate or deactivate sets of lower level genes. Thus the dynamic behaviour of genes at any level is governed by the high level genes.

In biological systems, there appear to be many possible strands of evidence supporting this model.

Fig. 4.6 outlines the categorical framework of evolutionary computation algorithm for the three levels. S represents the phenotype that is the NN nodes.

K1 and K2 are the two conditions levels. K1-represents the basic GA chromosome.

K2-is the meta-level representing the structured GA, sGA chromosomes. The strategies are defined at the meta-level since a change in K2 has higher leverage because it represents multiple changes at lower level K1.

U10: K1→S describes the decoding P01: S→K1 describes the fitness evaluations. The information change between the basic level and the meta-level of GA is characterized by the operator U21 and the possibilities P12. They describe sGA rules.

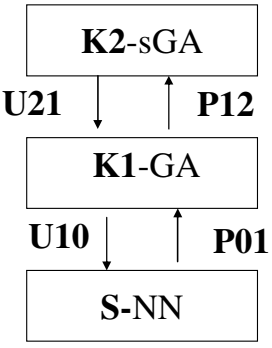


Fig. 4.6 Evolutionary designs for artificial neural networks

U21 corresponds to transcription processes.

The gene activation mechanism may be expressed by the categorical product.

It is possible to make use of different types of categorical product in K1 and K2.

Observe that the architecture shown in Fig. 4.6 may act like a complex network in which the genes corresponding to K1 and K2 regulates one another activity either directly or through their products, from S. This makes the architecture applicable to the GA-deceptive problems.

Other three level hierarchical architectures useful for modeling are the contextual GA (Rocha 1997) and the chemical GA (Suzuki and Sawai 2002).

4.2.4.3 n-Graphs for Growing Neural Networks

The Nolfi and Parisi growing neural networks (Nolfi and Parisi 1995a, 1995b) may be studied in the n-graphs framework (Appendix A5).

Fig. 4.7 illustrates a potential development for NN dendrites architecture using n-graphs.

For the stage n=0 there are isolated neurons. The 1^{st} order evolutionary step is allowed by interactions with the substrate. At n=1 interactions and branches appear. It is the 1-categories level

Branches are separated for the n=1 stage but they interact as the 2^{nd} evolutionary step shows. The n=2 level corresponds to 2-categories and allows arrays of interacting branches, the coupling of two or more branches in macro-branches or trees.

The 3^{rd} order evolutionary step outlines the final stage, n=3 corresponding to a kind of single tree. The single tree pattern is specific to the growing. Some neurons and some branches remain undeveloped. Unconnected branches or neurons are removed. It is the 3-categories level.

The integrative closure, connecting also the n=0 and n=3 levels represents the challenge for such NN systems. A possibility shown in Fig. 4.7 is the

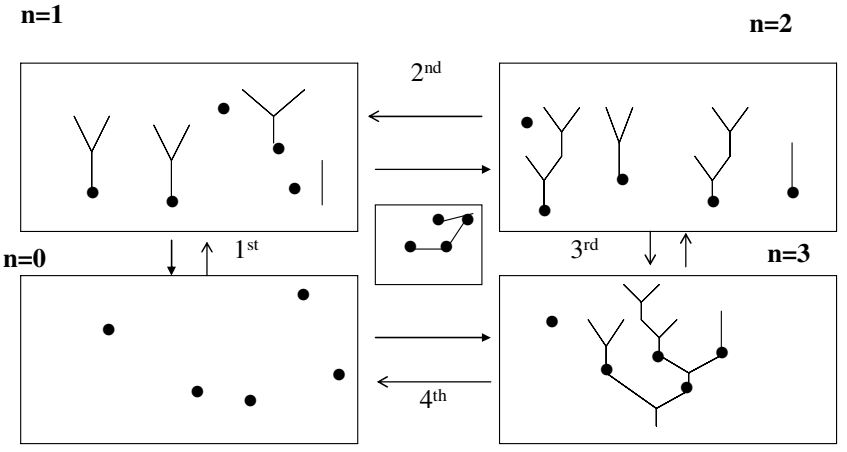


Fig. 4.7 n-graphs for growing neural networks

restriction to the central structure retaining only the four neurons connected at the levels 1, 2 or 3. Growing net may restart from this central structure and follow the same steps on a higher methodological plane that is at a new dimension in modeling. The branching should be reversible.

The integrative closure between cognitive levels and environment was studied in the so-called artificial life NN, ALNN (Nolfi and Parisi 1977). This study models an NN that lives in a physical environment. An active, embodied knowledge and knowledge acquisition makes ALNN closer to Piaget schema rather than to the classical NN.

Networks that adapt or self-organize structurally to the environment by adding and removing neurons and connections in the system exploit mechanisms that are similar to those used in the growth of an organism (Fritzke 1994). A developmental mechanism implies the presence of a mechanism for cell creation, a requirement for structural adaptation and thus can simplify the design of structure adaptable systems.

4.2.4.4 Protein Based Neural Networks

Many proteins in living cells appear to have as a primary function the transfer and processing of information accomplished by the physical or chemical transformation of metabolic intermediates or the building of cellular structures (Bray 1995, Spiro et al. 1997).

Cells perform calculations as a mean of monitoring and responding to their internal and external environment.

It was observed that the highly interconnected network of protein based pathways in living cells share the properties of neural nets allowing cognitive capacities. This refers to memory capacity, pattern recognition, handling fuzzy data, multifunctionality, signal amplification, integration and crosstalk

and signal amplification (Paton and Toh 2004). Moreover the mathematical formalism of artificial neural networks is a more accurate approximation for networks of protein molecules than for networks of real neutrons (Bray 1995). Ideas from the category theory can be used to illustrate this point. The internal organization of a protein can be modeled by a diagram of domains that is cooperating objects in which links represents functional relations. A colimit glues a pattern into a single unity in which the degrees of freedom of the parts are constrained by the whole. A limit represents the relationship between the whole that is the single unity and its components. It is possible to reason about functions with regard to how a whole is integrated and coheres out of its parts. Part-whole relations may be described as emergent cohesion reflecting the internal synergy in which interactions and local measurement generate cohesion. Cohesion concerned with part-whole relations is correlated to colimits in the sense that the whole keeps the parts together.

Proteins molecules can act as logical elements and assemblies of proteins can be artificially coupled to perform computations.

Bacterial chemotaxis illustrates the computation potentialities of protein networks in living cells. Chemotaxis is the process by which a cell alters its speed or frequency of turning in response to an extracellular chemical signal.

A four realms framework may describe the circuit mediating the chemotactic response of bacteria.

Fig. 4.8 shows illustrates the organization of protein based neural network for *E. Coli* chemotaxis. Here S denotes the receptors. These include chemoreceptors and amino-acids as ligands. The proteins network includes also a hierarchy of signaling proteins as: CheA, CheB, CheR, CheW, CheY and CheZ.

It may be assumed that the categories of signaling proteins are: K1={CheA, CheW}, K2={CheB, CheY} and K3={CheR, CheZ}.

The signification of the functors U and possibilities P is explicit in Fig. 4.8.

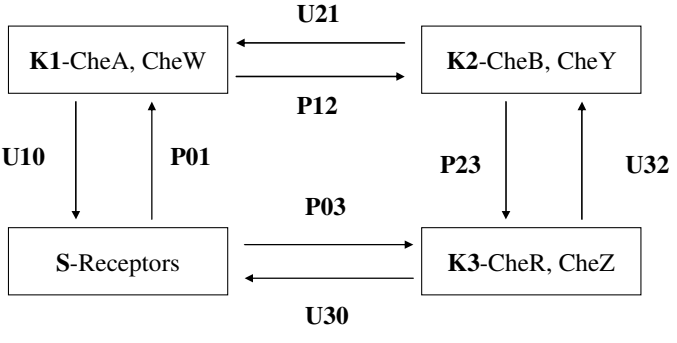


Fig. 4.8 Protein network mediating the chemotaxis

For instance, the operators U10 describes the interaction {CheA, CheW} → Receptors while the operator U30 describes the interaction {CheR, CheZ} → Receptors.

The mechanism is based on three inter-correlated realms for signaling proteins (Bray 1995). Other chelating proteins and other hierarchical order have been considered in the vast literature dedicated to chemotaxis.

The framework shown in Fig. 4.8 doesn't aims to reproduce the details of the *E. coli* chemotaxis but to retain the basic pattern and to make suggestions how to design artificial neural networks.

It is known that despite its relatively simple structure, *E. Coli* is capable of embodiment and highly autonomous behavior (Quick et al. 1999).

Viewed as signal processing systems, cell signalling networks like that shown in Fig 4.8 can be considered as special purposes computers. In contrast to conventional silicon-based computers, the computation is not realized by electronic circuits but by chemically reacting molecules in the cell.

Such system may achieve the essential properties of integrative closure. A critical feature of this closure is that the steady-state values reached after a changed input should still ensure the autonomous core functioning of the entire system.

As computational devices the proteins networks can be compared to analog computers. Several analog computers have long been displaced by the digital computers due to programming and stability issues. However there are situations were it is required to interface computation with chemical interactions when artificial protein based neural networks may be used to implement special computation and signal processing tasks. This may have direct applications to the so called smart drugs or for other bio-medical interventions.

4.3 Artificial Neural Codes

4.3.1 Neural Coding

The neural coding problem in perception of signals involves the interpretation of the neural correlations of sensory registrations (von der Malsburg 1994, Freeman 2000, Cariani 2001).

Sensed information can be encoded in patterns of neurons that respond, the so-called channel codes or in temporal relations between spikes that is temporal codes.

Progresses in the domain of understanding neural codes have been achieved starting from neural activity modeling.

In the firing rate model it is speculated that the information is conveyed by being encoded in the rate at which action potential are generated by neurons. Over short times the network structure of the brain is commonly regarded as fixed. Brain states may be regarded as semantic symbols. They are lacking hierarchical and syntactical structure.

The correlation theory of brain function (von der Malsburg 1986, 1994) challenged the semantic symbol system sketched above and proposed a different interpretation of data in terms of semantic symbols with a richer structure. The correlation theory suggests that information is conveyed in the brain through correlations of neural firing patterns.

This theory received support from a model developed by Damasio which holds that entities and events are represented in the brain by time-locked synchronous neural firing patterns (Damasio 1989).

An approach based on the PSM method and the wave equation, WE, may generate solutions looking like the temporal response patterns registered in brain studies.

The contact with existing symbolic neural architectures may be established and on this basis potentially neurocognitive architectures are resulting.

Neural-symbolic systems are hybrid systems that integrate symbolic logic and neural networks. The goal of neural-symbolic integration is to benefit from the combination of features of the symbolic and connectionist paradigms of artificial intelligence, AI.

In the basic SKUP framework K is associated to symbols while S to neural networks.

An open problem is how to put together in the same framework K and S.

It may be assumed that a compositionality principle would allow computing the meaning of complex formulas using the meaning of the corresponding sub-formulas.

On the other side, it was assumed that NN are non-compositional from principle, making them unable to represent complex data structure like formulas, lists, tables, and so forth.

Two aspects can be distinguished, the representation and the inference problem.

The first problem states that complex data structures can only implicitly be used and the representation of structured objects is a challenge for connectionist networks. This is correlated to possibilities P in the basic SKUP framework.

The second problem, of inference, tries to model inferences of logical systems with neural account. This is correlated to operators U in the SKUP.

4.3.2 Symbolic Connectionist Hybrids

Some authors claimed that connectionist models as the neural networks, NN, did not support symbolic processing and were incapable of adequately representing evolving neurocognitive structures (Fodor and Pylyshyn 1988).

Symbolic connectionist models, implemented as hybrid devices allowed a rebuttal of Fodor and Pylyshyn criticism in both theory and practice.

Hybrid symbolic connectionist techniques allow vectors representing the constituents of a symbol structure to be combined into a single vector representing the whole structure, and for this vector to be decoded into the

vectors representing the original constituents. In this manner representations for compositional structures can be built up and, then processed by NN.

The recursive auto-associative memory (RAAM) was among the NN models developed to address the question of how compositional structures may be stored within a connectionist framework (Pollack 1990). The data for a RAAM network consists of a collection of trees and a representation that is a pattern of "0", "1" and so on for each terminal symbol occurring in those trees. The task for the network is to provide a means of compressing each tree into a representation, an activation vector, and reconstructing the tree from its representation. The elements of the SKUP are naturally associated to the RAAM elements. The input and output units may be associated to the set of conditions, K. The RAAM architecture contains encoding or compressor networks, associated to possibilities P in the SKUP framework. The RAAM contains also decoding or reconstruction networks associated to operators U. The hidden units are associated to the states S. As shown in Sect. 2.3.1, to any matrices containing discrete information as "0", "1" and so on, classification trees may be associated based on similarity calculations. Another promising NN architecture is the distributed associative memory developed by Austin (Austin 1996). Associative memories operate differently from the memories typical for current computer architectures. This type of architecture take a set of data often in the form of an image, and scan the entire set of data in memory until it finds a set that matches it, as much as possible.

Fig. 4.9 shows a typical neurocognitive architecture making use of two categorical frames for conditions, K1 and K2 with two types of tensorial product, the coproduct "$\cup$" for K1 and the product "$\times$" for K2.

The architecture proposed by Austin makes use for symbolic processing, the component-wise operations in GF (2). The categorical product "$\times$" is in this case a vectorial outer product, while the categorical coproduct, "$\cup$" is a concatenation followed by superimposed coding.

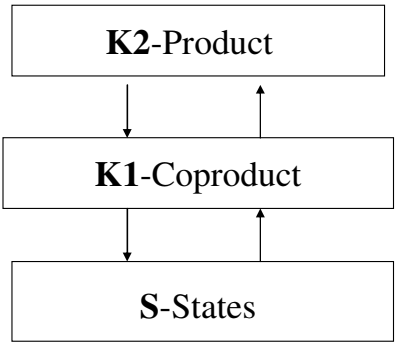


Fig. 4.9 Typical three levels framework

Another strategy to meet the challenges posed by connectionism critiques for both models and devices is offered by the so-called holographic reduced representations HRR, (Plate 1995). Associative memories are conventionally used to represent sets of pairs of vectors. Plate describes a method for representing complex compositional structures in distributed representations. The method uses circular convolution to associate items which are represented by vectors. The representation of an association is a vector of the same dimensionality as the vectors which are associated. The method allows encoding relational structures in fixed width vector representation but it should be noted that this increases the risk of missing the emergent structures.

Plate calls his models, holographic reduced representations, since convolution and correlation based memory mechanisms are close related to holographic storage. The circular convolution may be associated to the categorical product, "$\times$", while the superposition may be associated to categorical coproduct, "$\cup$" (Fig. 4.9).

The properties of higher neurocognitive processes and how they can be modelled by NN have been extensively studied by Halford and collaborators (Wilson and Halford 1994, Halford et al. 1998). They proposed the so-called STAR model of analogical problem solving.

The rank of tensor used by Halford is linked to the arity of relation, that is, to the number of attributes to the relation, and in the end, to the Piaget stages of neurocognitive development. The STAR model uses a tensor of rank-3 to represents a predicate of two arguments.

Halford studies suggests that for early Piaget stages in neurocognitive development, the categorical coproduct, "$\cup$", prevails allowing the associative knowledge. This is a fast and parallel process. During the higher Piaget stages the categorical product, "$\times$" seems preponderant, allowing the relational knowledge. It is a slow, sequential, effortful, higher neurocognitive process (Fig. 4.9).

The study of tensor product networks using distributed representations outlined the significant role of Hadamard matrices (Wilson and Halford 1994). As shown in Sect. 2.2.3 these matrices are special solutions of the WE equations.

Notice that Halford and associates evaluated the significance of Klein-4 group and of Latin squares for learning transfer in NN and in neurocognitive systems. Such structures correspond to the INRC group studied by Piaget (Inhelder and Piaget 1958) as well as to standard solutions of the WE model.

4.3.3 Temporal Synchrony

A promising way of dealing with variable binding in connectionist systems is to use the temporal aspects of nodes or neurons activation. Phase synchronization can be used since it allows different phases in an activation cycle to represent different objects involved in reasoning, and representing variable

binding by the in-phase firing of nodes (von der Malsburg 1986, Hummel and Biederman 1992).

Based on temporal synchrony, SHRUTI system (Shastri and Ajjanagade 1993) provides a connectionist architecture performing reflexive reasoning. SHRUTI shows how synchronous activation can be harnessed to solve problems in the representation and processing of high level conceptual knowledge. LISA system (Hummel and Holyoak 1997, Hummel and Choplin 2000) used the synchronous activation approach to model analogical inference. Both computational systems demonstrates that temporal synchrony in conjunction with structured neural representations suffices to support complex forms of relational information processing specific to neurocognitive systems.

The problem for such systems is their suitability for reflexive or reflective neurocognitive processes. Reflexive processes are linked to categorical coproduct while reflective processes, are linked to the categorical product (Fig. 4.9).

While reflexive and reflective processes follow different kinds of computational constraints, in most cases, the two types of processes interact and need to be integrated in the performance of a single task.

SHRUTI represents a restricted number of rules with multiple place predicates. There are several types of nodes or neurons in the architecture, denoted for example by circles, triangles and pentagons.

Relational structures as frames and schemas are represented in SHRUTI by focal clusters of cells, and inference in SHRUTI corresponds to a transient propagation of rhythmic activity over such cell-clusters. Dynamic bindings between roles and entities are represented within such a rhythmic activity by the synchronous firing of appropriate role and entity cells. Rules correspond to high-efficacy links between cell-clusters, and long-term facts correspond to coincidence and coincidence-failure detector circuits.

SHRUTI was designed for reflexive reasoning tasks and the model is not suited to account for reflective processes.

To ensure applicability to complex situations, SHRUTI was coupled with systems activating the reflective component of problem solving. Such systems are capable of attention shifting, making and testing assumptions, evaluating uncertainty. The resulting neurocognitive systems presented both reflexive and reflective capabilities and has been used to model decision making in imposed time frames.

LISA is a computational model based on temporal synchrony and designed for analogical inference and for schemes induction. The data for LISA network consists of a collection of trees and a representation that is a pattern of “0”, “1” and so on for each terminal symbol occurring in those trees.

The LISA system is shown in Fig. 4.10.

The basic level includes semantic units, s, the next includes the so-called localist units, L, (predicate/object or object/roles), the next level includes the sub-problems and the higher level the problems.

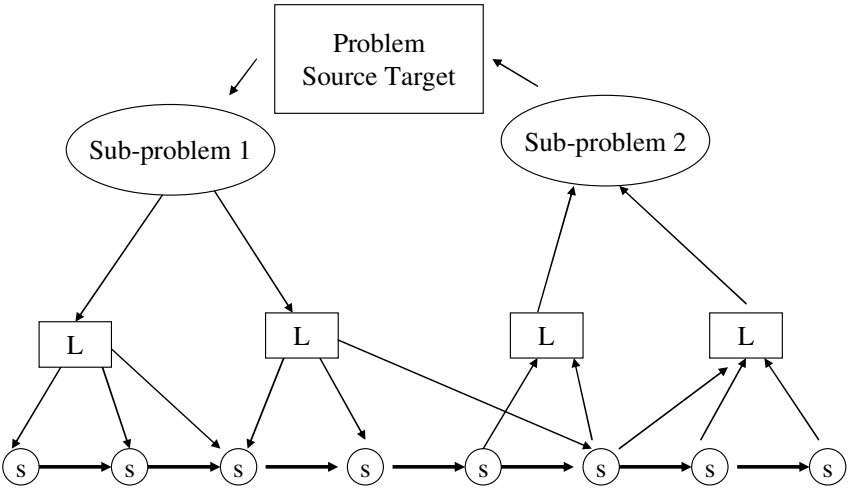


Fig. 4.10 LISA neurogonitive frameworks

4.3.4 Perspectives

4.3.4.1 Three Realms Neurocognitive Architectures

Three realms multi-agent architectures may achieve integrative closure, in this case cognitive evolvability and autonomy (Zachary and Le Mentec 2000, Di Marzo Serugendo et al. 2007).

An agent architecture grounded in models of human reasoning such as Cognet is shown in Fig. 4.11 (Zachary and Le Mentec 2000).

Cognet is a research framework concerning the analysis and modeling of human behavioral and neurocognitive processes in real-time, multi-tasking environments.

Meta-cognition refers to cognition about cognition and in this case to the ability to explicitly and strategically think about and control an agent's own neurogonitive processes. The Cognet architecture allows a meta-cognitive control of neurocognitive processing. An emphasized aspect is that of self-awareness of resources and processes.

The categorical framework is shown in Fig. 4.12. It shows the architecture of conditioning levels with two-sided dependence.

The elements of the categorical framework are as follows:

S-Environment interface allowing action and perception

K1-Cognition processes, K2-Metacognitive processes

The neurocognitive level is structured in K1 and K2 to allow performing integrated cognitive/behavioral tasks.

U10: K1→S actions physical or verbal

P01: S→K1 sense of visual and auditory cues

U20-motor action resources, P02-perception resources

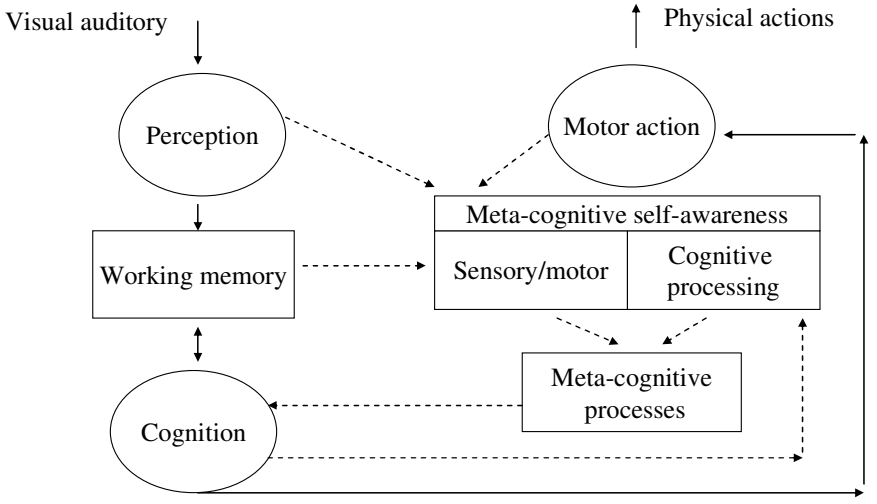


Fig. 4.11 Cognet information processing framework

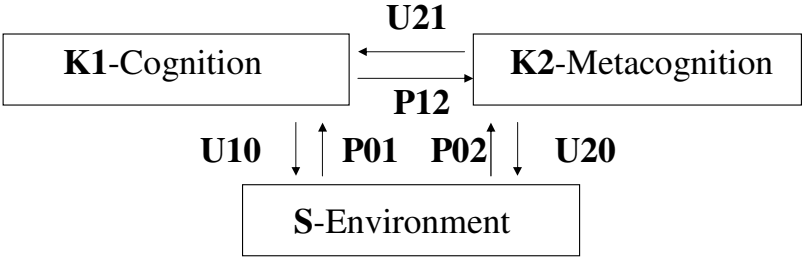


Fig. 4.12 Three realms neurogonitive framework

A similar framework is shown by the self-adaptive and self-organizing SASO architectures (Di Marzo Serugendo et al. 2007).

In this case, the elements of the categorical framework are as follows:

S-Application components, services
K1-Metadata, K2-Policies
U10: K1→S defines acting
P01: S→K1 defines sensing
U20 and P02 describes the application and acquisition of policies

Elements of the cognitive architecture shown in Fig. 4.12 may be correlated to the adaptive resonance theory ART (Carpenter and Grossberg 1987).

In this case, the elements of the categorical framework are, S-receiver of the input signals.

K1-classifier of the afferent input patterns and K2-attention/orienting subsystem.

Viewed abstractly, the ART classifier network meets the definition of an algebraic structure known as grupoid. Formally a grupoid is any mathematical structure consisting of a set of inputs and an operation on this set possessing the property of closure. A grupoid may be seen as a category in which any morphism is an isomorphism.

4.3.4.2 Four Levels or Realms Neurocognitive Architectures

A four level architecture for LISA approach is presented in Fig. 4.13.

This architecture takes into account that the working capacity of human is typically limited at four relations (Halford et al. 1998). Hummel and Holyoak (1997) correlate the four levels of memory in the LISA neurocognitive framework to the limits in mental storage capacity. Probably, this fact is related to the four modular architecture of the neurocognitive system and to cerebral rhythms (Freeman 2000).

The signification of the functors U and possibilities P is explicit in Fig. 4.13.

S-Semantic units, K1-Localist units, K2-Sub-problems, K3-Problems

U10, U21, U32 corresponds to implementation operations

Observe that: U10:K1-Localist→S-Semantic, U21: K2-Sub-problems→K1-Localist, and U32: K3-Problems→K2-Sub-problems.

P01, P12, P23 and P03 are synthesis steps.

P01: S-Semantic→ K1-Localist, P12: K1-Localist→K2-Sub-problems, and P23: K2-Sub-problems→K3-Problems.

The four realms approach emphasizes the need of contact between the problem and the ground semantics units.

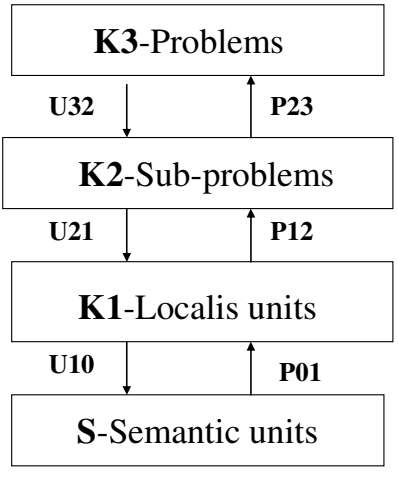


Fig. 4.13 Four levels neurogonitive framework

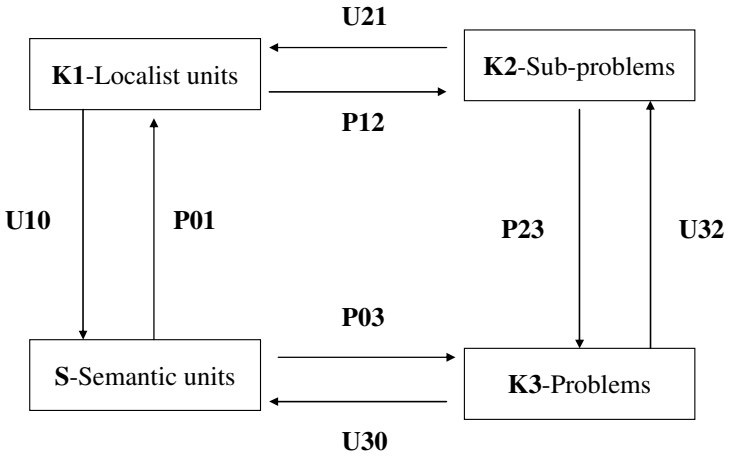


Fig. 4.14 Four realms neurogonitive framework

Fig. 4.14 shows a prospective four realms architecture developing the LISA approach towards integrative closure and evolvability. It is expected to facilitate the interaction between S-Semantic units and K3-Problems.

U30 correspond to implementation operations. In this case U30: K3-Problems→S-Semantic units and U30=U10oU21oU32.

P03 is a synthesis step. In this case P03: S-Semantic units →K3-Problems and P03=P01oP12oP23

The basic realm n=0 includes semantic units, the next realm n=1 includes the so-called localist units, the realm n=2 includes sub-problems and the realm n=3 the problems to solve.

The task for the LISA network is to provide a means of compressing each tree into a representation, the so-called activation vector, and reconstructing the tree from its representation. SKUP elements are naturally associated to the LISA elements. The problems to solve may be associated to the set of conditions K. LISA contains a driver network associated to operators U, and to the reflective reasoning. They are U10, U21, U32 and U30. As a difference from SHRUTI, the initial LISA model was not developed to account for the reflexive processes. However the representational structure of LISA provides at least a starting point for reflexive reasoning capabilities. LISA propositions are retrieved into memory via guided pattern matching. During retrieval and comparisons the proposition are divided into two mutually exclusive sets: a driver and one or more recipients or receivers. The receiver network is associated to possibilities P. The elements are P01, P12 and P23. The calculus of possibilities for LISA model was studied by Taylor and Hummel (2009).

The switch between reflexive and refractive reasoning passes trough the semantics. The LISA semantics elements are associated to the states S in SKUP.

The activation of semantic units is controlled by time. Often the analysts do not have the time to allow runaway activation of semantics because they needs make inferences quickly. Notice that in contrast to reflexive inferences which are fast, the reflective inferences may require more effort. An open problem is to establish, for imposed time frames, the number of switching from reflexive to refractive and the order in which the switching should be performed.

4.3.4.3 Spatial Cognition

A four level hierarchical architecture was operated in the study of complexity of behavior for spatial cognition (Mallot 1999) (Fig. 4.15).

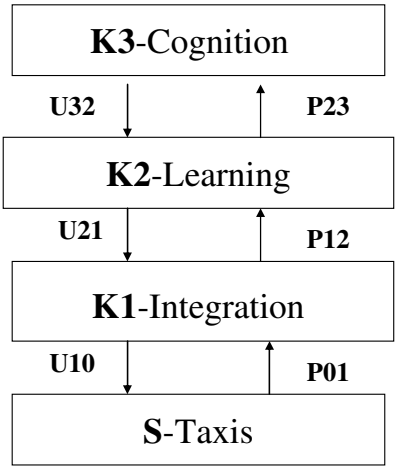


Fig. 4.15 Four levels for complexity of behavior

The basic level S-Taxis, describes the reflex-life behaviors.

The level K1-Integration, requires spatiotemporal combination of data on the basis of a simple working memory. The level K2-Learning, requires long-term memory for procedures.

The level K3-Cognition, requires declarative memory consisting of neurogonitive maps allowing changing behavior according to current goals. Cognitive behavior is characterized by goal-dependent flexibility.

It is difference in time scale for the four levels in the sense that higher levels are slower.

The spatial cognition is important in the study of autonomous vehicles (Trullier and Meyer 1997). The functional model of the role of the hippocampus in navigation was implemented as a multi-level feed forward neural-network (Burgess et al. 1994). The first layer identified as K1 in hierarchy

consists of sensory neurons that discharge selectively when the environment-S elements are sensed. The place cells represent the main elements of the K2 layer while the goal cells may be associated to the level K3.

A model of animat navigation based on several neurogonitive modules was proposed by Schmajuk and Thieme (1992). One module encodes topological representation and the other selects movements on the basis of predictions generated by the first module.

The module K2 is the comparator and elaborates next place predictions based on slow changing signals. A neurogonitive map is part of this. The module K1 allows goal predictions based on fast changing signals. The module K3 corresponds to goals.

Another model for spatial cognition was based on mesoscopic dynamics of brain activity (Freeman 2000). Freeman proposed a hierarchy of models that have capacity to show aperiodic behavior similar to that shown by electroencephalograms.

For the K-sets computational model due to Freeman (Freeman 2000) the basic level S is linked to columns, the next level K1 to bulbs, the next level K2 to cortex and the next level K3 to hemispheres of the brain. Notice that the corresponding Freeman notations are KI, KII, KIII and KIV.

4.3.4.4 Arrays of Neurogonitive Tiles

In an attempt to develop a model of memory von Foerster (von Foerster1969) proposed a tessellation of neurogonitive tiles. The idea was to advance a conceptual minimum element capable of observing the desired neurocognitive characteristics of memory (Rocha 1995). The SKUP are considered as neurogonitive tiles. SKUP includes a self-referential level of interactions, in which an internal meaning of measured states of the memory empowered organization is generated, a closed system, with external meaning is obtained.

The SKUP is seen as a through-put system. Sensory information S is compared in a feed-forward fashion and altered in respect to the existing scheme in K. The feedback loop incorporates the delay of the system, that is, the associated time-scales.

We can think at these neurogonitive tiles as a suggested conceptualization of the necessary connections between symbol and matter in order to obtain closure.

The SKUPs may be correlated in an array to obtain an autonomous classification function that is autonomous neurogonitive architectures.

It seems to be of interest to use such arrays not only in higher level modeling approaches to evolvability and cognition as in traditional artificial intelligence, AI, models but also at the presumable lower level of artificial life, AL, models. In a cell we can find different processes with different time-scales. If these processes can be organized into semantically closed groups, then they can be represented by neurogonitive tiles and a functioning of the cell by a tessellation. The network of SKUPs includes processes that affect the

time-scales of other processes. With an array of SKUPs we may be able to recognize the true temporal pattern recognition and not simply sequence recognition as in artificial NN. We can thus start to consider a tessellation of neurogonitive tiles as a proper measurement device which becomes dynamically coherent.

Implemented by a network of interacting SKUPs such arrays will be responsible for the recognition of temporal rather than spatial patterns of inputs. The recognition of appropriate temporal patterns will then dictate the neurogonitive system response. Clearly time plays the important role on the functioning of these arrays, unless the time-less switches of conventional artificial NNs.

Complex networks of tessellations can be organized as blocks of tiles dynamically closing higher semantic loops based on other semantic loops. This may correspond to categorification process (Appendix A4).

4.3.4.5 n-Graphs for Neural Symbolic Computation

In neurodynamics studies the entities are embodied in the network's nodes and activated by associations. Logical systems define symbols that can be composed in a generative way but do not posses a microstructure appropriate for perception and learning tasks.

An illustration of the neural symbolic frames is based on the representation of the NN multi-scale evolution in term of n-graphs (Appendix A5).

The n-graphs characterize asynchronous systems with multiple entrances and exits.

Fig. 4.16 describes the process of self-structuring in a neural network and the emergence of symbols. The model is biomimetic.

The role of neocortical self-structuring as a basis for learning in neurodynamics was emphasized by von der Malsburg (1986, 1994), Doursat and Bienenstock (2006), Doursat (2007). It should be emphasized that Doursat (2007) approach is limited to a three level hierarchy.

The level n=0 represents to the 0-graphs or sets. This is associated in this case to the unit isolated neurons. The level n=1 correspond the 1-graphs. These are directed graphs including the morphisms that is, the connections between neurons.

Here the connected neurons are denoted by A, B, C and so on.The morphisms are 1-cells, describing relations. Their coupling in the right order allows the complete signal transfer process.

The level n=2 corresponds to the 2-graphs. These are graphs plus the so-called 2-cells between paths of same source and target. These 2-cells describe relations between relations and express the natural lumping of the simplex A, B, C of neurons in just one loop with specific role.

The level n=3 corresponds to the 3-graphs. These are 2-graphs that include 3-cells that is, the cells between 2-cells. The 2-graphs and 3-graphs represent graphs modification and should be subjected to conditions of

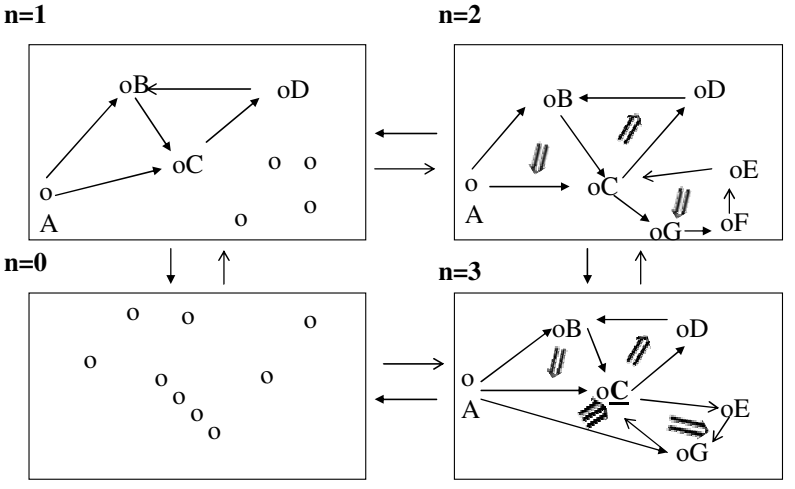


Fig. 4.16 n-graphs for neural symbolic computation

natural transformations. These are related to travels time between neurons and association of neurons.

Denote by τ_{AB} the time necessary to travel from node A to node B (Doursat 1991) Conditions as $\tau_{AB} + \tau_{BC} = \tau_{AC}$ or $\tau_{CG} + \tau_{GF} + \tau_{FE} = \tau_{EC}$ should be imposed at the reality level n=2 to ensure the signal route equivalence. Double arrows emphasize these equivalences. There exist two different compositions of 2-cells. The vertical composition corresponds to sequential 2-cells, while the horizontal composition corresponds to parallel 2-cells.

The level n=3 corresponds to the interaction between two component networks of interactions.

In that case coherence conditions are: $\tau_{AG} + \tau_{GC} = \tau_{AC}$.

Triple arrows characterize equivalences between routes including the two component nets.

The integrative closure, connecting n=0 and n=3 represents the challenge for such systems.

The n-graphs for successive stages n, outlines a hierarchy of correlations of successive orders for neural patterns, a hierarchy of binding levels (von der Malsburg 1994, 2004). The mechanism is formally similar to that of concentration zones, CZ, as discussed by Damasio (1989). Damasio modelled the brain as an interacting system of hierarchical sub-networks for different major neural computing tasks. Healy and Caudell (2006) elaborated a category theory approach to NN emphasizing the potential role of Damasio approach. The categorical model, with functors from a category of concepts to a category of NN components and natural transformations between these functors, provides a mathematical model for neural structures consistent with concept-subconcepts relationship. Colimits of diagrams show how concepts

can be combined and how a concept can be re-used many times in forming more complex concepts. The functors map commutative diagrams to commutative diagrams capturing this aspect of the colimit structure. Natural transformations express the fusion of single mode sensor representations of concepts in the same neural architecture.

This categorical model is compatible with the model of binding proposed by von der Malsburg or of concentration zones proposed by Damasio.

The architecture shown in Fig. 4.16 suggests considering three stages of binding or of concentration followed by the 4^{th} order stage of integrative closure and embodiment.

The stages may be considered in terms of categorical colimits. Categorical limit is the emergent concept summing up in itself the properties of its constituents. This generates the n-graphs hierarchy where at any stage the objects are the colimits of objects of the previous stage. This means that higher level of binding needs n-categorical models.

This idea may be linked to the attempts to make computers more close to natural brain-body system. It was observed that models suited for "off-line" computation such as Turing machine should be replaced with frameworks that are more readily to accommodate "on-line" and "real-time" processing of environment input streams.

An approach in line with "artificial chemical engineering" was proposed by Maass (2007). Maass proposed a framework calling it the "liquid computing" which is a generalization of classical finite states machine to continuous input streams and state space. The "reservoir computing" paradigm (Schrauwen et al. 2007) develops the main idea of "liquid computing" paradigm, that is the separation of the producing output stream from processing the input stream. Finding out what is a good reservoir represents an active research area. The n-categorical point of view may introduce a right structuring in levels of the "reservoir computing".

4.4 Evolvable Circuits

4.4.1 Evolutionary Circuits

4.4.1.1 Evolutionary Behavior for Circuits

Evolutionary or proactive circuits have the capability to change the preemptively embedded circuitry elements in order to keep on and to accomplish un-programmed tasks. Evolutionary behavior is a forward-looking perspective from engineering point of view, enabling to the designed circuit to modify faster in the anticipation of the future constraints of the environment.

Evolutionary circuits make use of self-construction elements offered by the basic generic framework, and by the environment. The evolutionary devices and sensors developed by the cybernetician Gordon Pask (Cariani 1989,

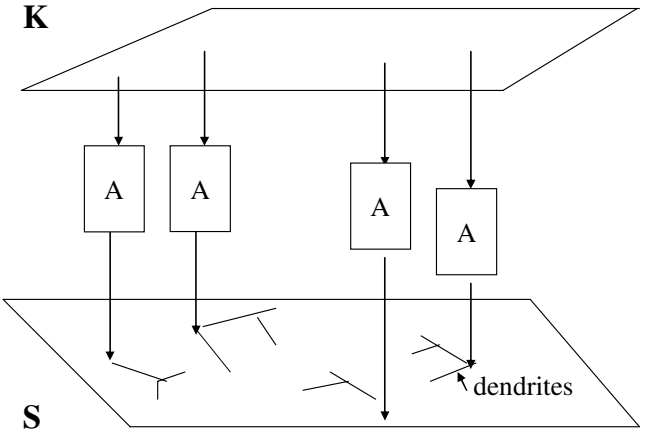


Fig. 4.17 Pask's evolutionary device

1993), the "evolved radio" described by Bird and Layzell (2002), some developments of "evolutionary hardware" (Thompson 1998) may be regarded as a kind of proactive circuitry implementations.

One way to achieve to the circuit a degree of autonomy is to have sensors constructed by the system itself instead of sensors specified by the designer. Cariani refers to "Pask's Ear" as a first example of such evolutionary circuits (Fig. 4.17). The Pask's system is an electrochemical device consisting of a set of platinum electrodes in an aqueous ferrous sulfate/sulfuric acid solution. When current is fed through the electrodes iron dendrites tends to grow between the electrodes. If no or low current passes through a thread, it dissolves back into the acidic solutions.

The threads that follow the path of maximum current develop the best. In the complex growth and decay of threads, the system mimics a living system that responds to rewards that is more current and penalty that is less current. The system itself is able to discover the most favorable forms for the condition, which may embed information concerning other factors of the environment such as magnetic fields, auditory vibrations, temperatures cycles. The system resembles a model of ants leaving pheromone to reach a target (Virgo and Harvey 2008).

This circuit was trained to discriminate between two frequencies of sound by rewarding structures whose conductivity varied in some way with the environmental perturbation. The Pask's evolutionary device created a set of sensory distinctions that it did not previously have, proving that emergence of new relevance criteria and new circuits is possible in devices. The Pask's device may be considered as an analogous realization of the SKUP and of the categorical framework. The dendrite structures forming in malleable

materials correspond to the category S, the resistance, capacitance or ionic resistance linkage to energy is linked to the category K. The evaluation of the signal network developed in malleable material is part of the possibilities P. Amplifying servomechanisms, A, may be linked to the operators U.

Following similar ideas, Bird and Layzell (2002) built an "evolved radio". Like Pask's ear the evolved radio determined the nature of its relation to environment and the knowledge of a part of the environment.

Bird and Layzell emphasized that novel sensors are constructed when the device itself rather than the experimenter determines which of the very large number of environmental perturbations act as useful stimuli.

The relation with von Uexküll *Umwelt* concept is obvious. Both of these devices, the Pask's ear and the evolved radio show epistemic autonomy that is, they alter their relationship with the environment depending on whether a particular configuration generates rewarded behavior.

Evolutionary systems include the four basic parts of the von Uexküll functional cycle: object, sensors, the command generator and the actuator (von Uexküll 1973). These parts are associated to the SKUP with the states S, the possibilities P, the conditions K, and the operators U, respectively. In the categorical framework the perception is associated to the functor P and the action is associated to the functor U. They can be viewed as adjoint connections of the different categories K and S.

Moreover, as in the functional cycle the evolutionary systems includes two levels for K, one related to the control K1, and a higher one K2 related to the coordination.

4.4.1.2 Evolutionary Hardware

Research for evolutionary devices, is associated to the domain of MEMS, micro-electro-mechanical systems, MECS, micro-energy-chemical systems and to evolutionary hardware (Thompson 1998, Mahalik 2005).

MEMS represent the integration of mechanical elements, sensors, actuators and electronic circuits on a substrate through the utilization of microfabrication technology.

MECS focus on energy transfer, microfluids and chemical reactions. The evolutionary circuit is the candidate for the "brain" part of the systems while MEMS or MECS would allows to the micro-systems to sense and to control the environment.

In the associated SKUP quadruple, the environment corresponds to the states S, the evolutionary circuit itself to conditions K, the MEMS or MECS control part is linked to the operators U and MEMS or MECS sense part to possibilities P.

For embedded EDOE, the MEMS or MECS and ultimately, the prined circuits board, PCB may be the physical support material. Coupling evolutionary circuits with PCB and MEMS or MECS may ensure robustness and autonomy (Cheung et al. 1997, Mahalik 2005).

Evolutionary hardware represents an emerging field that applies evolutionary computations to automate adaptation of reconfigurable and polymorphic structures such as electronic systems and robots (Thompson 1998).

Evolutionary computation methods in designs that take the performance of a scheme as prediction for the performance of a modified scheme are suitable for evolutionary circuit development (Koza 1992). Koza elaborated genetic programs that could design band-pass filters that are electrical circuits able to separate signals of one frequency versus another. There is no explicit procedure for conventional designing these circuits due to the large number of optimization criteria. The algorithms work by starting with simple circuits and evolving them. The program, then created different variations, tested them, select the best and used them for the next generations. Implemented on silicon such programs may result in a circuitry that has attributes of novelty. The program may be used to generate evolutionary circuit schemes.

An interesting suggestion for evolutionary hardware architecture is the CellMatrix (Macias 1999). CellMatrix develops self-modifying, self-assembling and self-organizing circuits. These circuits are designed for, and constructed out of, a unique hardware substrate. The Cell Matrix may modify circuit architecture in the direction of locally connected; re-configurable hardware meshes that merge processing and memory.

4.4.1.3 Electrochemical Filaments Circuits, ECFC

Based on electrochemical filaments development, a new type of evolutionary circuits, ECFC became possible.

ECFC construction starts with a generic framework representing the elements of the set of conditions K.

The K-framework elements may be that generated by wave equation, WE. The process in K generates successive K-frames, K0, K1,..., Km, at different levels.

The generic circuitry represented by K-frames is completed by additional circuitry, traces, dendrites, filaments, and supplementary matter, corrosion or degradation products. The processing for these additional circuits is an S-process. S-denotes the physical circuit based on filaments, threads, or microchannels for fluids allowing the electrical contact or interaction. The K steps and the real environment S-steps have complementary contributions in circuit building. ECFC are expected to be at least partially autonomous. The autonomy includes the capability in building, assembly, modifying, organizing, repairing and destroying. As a difference, if compared to adaptive and self-adaptive devices based mainly on feedback, ECFC make use of the preemptively embedded multi-scale K frames. The appropriate K designs and the selective addition and the subtraction of appropriate elements from environment are the processes allowing both self-functionality and evolutionary behavior.

- **ECFC design**

Suggestions of evolutionary behavior may be detected in conventional multi-purpose circuit designs. These circuits have only holes and conductors, which have to be connected or interrupted according to the specific assembly needs. Often components are assembled directly on the components side so that it is possible to make small changes and to keep the whole circuit under control. This technique permits easy adjustment and trials of different components to modify the circuit from design stage. Adjacent to multi-purpose design methodology is the existing design re-use. The need to decrease time to market imposes to make use of known good sub-circuits or known good blocks as building elements. The sub-circuits may be developed individually as component DOE in a design centered EDOE.

The basic elements of ECFC technology are the K-valued generic framework, linked to class of tasks, the environment media for self-construction in non-stationary or oscillatory fields and the self-learning capability by exposure to environmental complexity and to variable tasks.

The ECFC that results by coupling the electrochemical filaments, ECF of different orders m, ECFm, over a pre-existing K-frames, K0, K1, K2, . . . , Km is considered here. The circuit may be described using the categorical tensor "$*$" that links different levels in circuitry: ECFC=K0$*$ K1$*\ldots*$Km$*$ECF0$*$ ECF2$*\ldots*$ ECFm.

The tensorial product "$*$" may be the categorical product "$\times$", the coproduct "$\cup$" and so on.

The categorical presentation of this architecture is shown in Fig. 4.9.

The K-framework should be a quasi complete printed circuit, with several opens. These opens allows the ECFC versatility and multiple potentialities. The environment is able to fill the opens sequentially in a way that ensure functionality. Potential geometrical variants and architectures for ECFC are: dots, cells, hexagons, triangles, squares, circles arrays, circular crowns, dyadic structure, labyrinths and mazes, high density circuitry, self-similar nested structures, tiles, pre-fractals and multi-fractals used for evolving antennas.

It was established that the wave equation, WE, is able to generate fractal structures making use of categorical product "$\times$". For example Hadamard-Sylvester matrices, reduce to Sierpinski triangles if only the "1"s are considered while the "-1"'s or with other notations the "0"s are neglected since they breaks the circuit (Barnsley 1993).

The switch from categorical product to categorical coproduct determines the size and the shape of the circuit. The switch is determined by the oscillatory fields that accompanies the ECFC development.

- **Materials for ECFC**

The materials should offer opportunities for wet chemistry and for solid physics to play significantly. ECFC's make use of composites and multi-phase media. The materials should be as rich as possible in structural possibilities, for example in phase transitions, on the edge of chaos, in non-linear regimes.

Interesting options are the existing self-adaptive or smart materials that allow phase transition, such as the piezoelectric, thermoelectric, electrorheological, electro active polymers and so forth.

Laminate known as filaments non-resistant as polyester rigid woven glass, paper phenol, or specially contaminated laminates represents valid opportunities since they allow the electrochemical filament fast formation.

Possible K-frames conductor lines make use of materials like Cu, Ag, Sn, Sn/Pb, Zn, Al, Mg and Fe. Metallic inorganic salts for conduction may be: sulfates, chlorides or nitrates of Fe, Cu, Ag, Sn, Pd, Pt, Zn, Al, Mg and catalysts. Metallic oxides may be useful as potential dielectrics. Damaged or fatigued printed or integrated circuits represent new potentialities for proactiveness.

- **ECFC processing**

ECFC should be processed in the environment that is in real field conditions in which the circuit should be functional such as:

- Mechanical (vibration, pressure)
- TRB (temperature, relative-humidity, bias) with direct current, alternative current or pulse plating of variable frequency
- Light, radiation
- Cyclical operation of various types
- Superposed oscillatory fields

These kinds of fields are the usual field of stresses for reliability tests. This suggests that evolutionary circuits may results from some over-tested circuits still able to show new capabilities.

An example of flow chart for ECFC fabrication is based on the following steps:

- Build K-frames-based on the wave equation, WE, solutions
- Select appropriate environment
- Introduce the K-frames and media in field conditions and allows periodic signals, stress field
- Develop the first level of filaments, ECF0 during training for signal that needs to be sensed or for any encountered new signals
- Build the circuit ECFC=K0∗ECF0
- Repeat ECF0 procedure and allows ECF1 corresponding to another signal and so on
- Build the circuit ECFC=K0∗K1∗. . . ∗Km∗ECF1∗ECF2∗. . . ∗ECFm
- Resulting circuits may be covered with gel, organic coating, photo coating or lacquer to ensure protection and robustness.

The operators U from the associated SKUP describe the evolutionary circuit at different levels of its construction.

The ECFC would be a circuit useful and stable in its building conditions. For any new level another frequency domain of oscillatory field is associated. As much as the oscillatory field still exists the new level would be developed.

If the structured dendrite structures were located in a specific field, the resulting structure would be able to recognize the patterns of that field. Learning and removal of information is possible if any dendrite may continually be formed broken and regenerated. Training to discriminate signals may be accomplished with the help of WH waves. The similarity associated to WH waves as defined in Sect. 2.2.3 is associated to the potentialities P of the SKUP.

- **n-graphs for dendrite circuits development**

The growing of tiny gold wires circuits in voltage controlled colloids is an example of ECFC (Miller and Downing 2002).

Fig. 4.18 illustrates a potential development for dendrites architecture using n-graphs (Appendix A5).

For the stage n=0 the filaments, are isolated wires. At n=1 interactions and dendrites may appear. This is allowed by interaction within the substrate. Filaments are separated in the n=1 stage but they interact in the n=2 stage to form arrays of interacting filaments. The n=2 stage shows the coupling of two or more dendrites in macro-wires.

The final stage, n=3 corresponds to a kind of single dendrite. The single dendrite pattern is specific.

The integrative closure, connecting also n=0 and n=3 is still an open problem for such systems. The dendrites development should be reversible.

A process like this may be compared with the operadic development (Appendix A6). The transition from 2-graphs to 3-graphs may be described as an operad.

The relation between higher categories, n-folds operads and dendrite circuit growth was investigated by Forcey (2008).

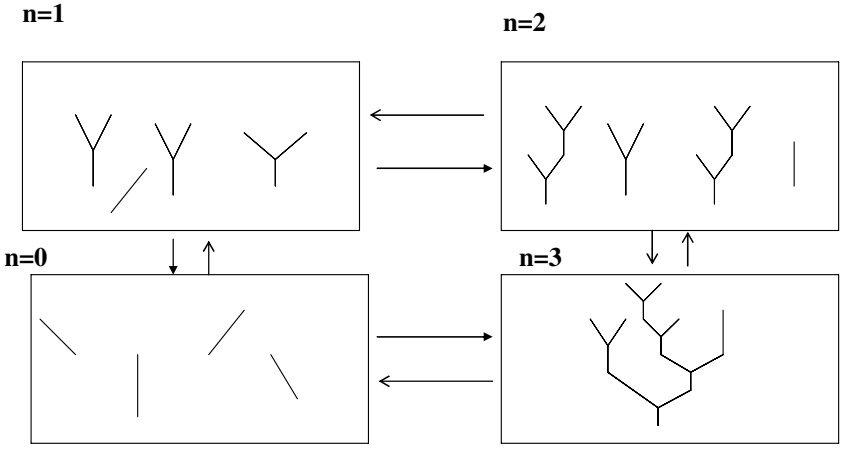


Fig. 4.18 n-graphs for dendrites framework

4.4.2 Evolvable Circuits

4.4.2.1 Evolvability Challenges

Evolvable circuits, EC, and evolutionary circuits are closely related. The difference between them corresponds to the degree of achievement from the point of view of full evolvability. This refers to the number of involved categorical levels and to closure. Evolutionary circuits show hierarchically architecture while evolvable ones suppose the integrative closure. This is also correlated to embodiment degree and to the scales. Transition from fixed circuits to evolvable circuits implies a change of scale, an increasing of the number of correlated scales. This is related to the categorical level too. Evolutionary circuits refer mainly to micro and meso-structured circuitry components while EC focuses also on molecular and nano-molecular structures facilitating the integrative closure. The evolutionary circuit is based on largely extrinsic designed and built circuits, while EC is expected to self-construct and to modify most part of their circuitry based on a genotype-phenotype-like scheme inherent to evolvability. Evolutionary circuit design is mainly from exterior while EC should be autonomous and self-programmed from the interior of the devices in a complex interaction with their environment. There is a threshold below which evolutionary circuit tends towards fixed circuits and above which they may progress towards fully EC.

Evolvable designs of experiments, EDOE was presented as a new modeling and simulation framework for complex problems solving (Iordache 2009). Additionally, EDOE may support the neurogonitive architecture for fully evolvable circuits, EC, practical implementation. The challenge is to build circuits that take advantage and control of their environment in increasingly complex ways. EC is supposed to be an embodied EDOE, able to run EDOE intrinsically, with emergent, behavior.

Unconventional principles, design of configurations, materials, fabrication methods, testing and applications have to be evaluated for evolvable circuitry (Bedau et al. 2000, Miller 2008, Rasmussen et al. 2004, Zauner 2005, Mahalik 2005).

Cellular automata suggest interesting architectures for EC soft. An example is the EvoCA cellular automata system (Taylor 2002). The EC's are supposed to be organizationally closed for matter but informational open. In order to realize evolvable systems, an important representational distinction should be between genotype and phenotype plus a biotic structure. As illustrated by EvoCA, semantically closed constructions may lead to novelty. EvoCA is a system where the environment is represented by a layer made of cellular automata, the physical or dynamical part, S and the genotypes represented by a second genome layer the inert or symbolic part, K. Each genotype controls a given cell in the first layer and evolves through a genetic algorithm. EvoCA-like constructions lead to operationally closed evolvable circuits, embedded in a dynamic environment, having metabolic-like potential, and being capable of self-replication and self-maintenance.

4.4.2.2 Molecular Electronics

Molecular and nano-molecular systems represent the promising domain able to ensure the objectives for EC that is to add evolution capability to devices, to self-construct systems going beyond learning and being capable to act completely autonomous in an indeterminate environment. The circuits may be electronic, optical, molecular, micro-fluidic, and so forth.

As expected, bio-molecules provided potential substrates to build technical information processes systems as EC. For example biologically available conjugated polymers, such as carotene, can conduct electricity and can be assembled into circuits.

Among the bio-molecules, the *Bacterio-rhodopsin*, BR, and the deoxyribonucleic acid, DNA received extensive attention. Hybrid systems that combine the best features of bio-molecular architecture, with optic, electronic, micro-fluidic circuits represent a necessary step in EC development. The hybrid character refers to both formal models and practical devices.

These hybrids are digital-analog devices. The analog aspects are related to rate-dependent processes, and the digital aspects are related to macro-states and to macro-state transition rules. The issue of digital-analog or symbolic connectionist complementarity is closely related to the closure concept and to evolutionary behavior for devices (Pattee 1995, Cariani, 2001).The potential of the hybrid devices and hybrid models remains to be developed, but by all indications, such representational method can provide strategies of unifying low-level connectionist and high-level symbolic models of neurogonitive processes.

4.4.2.3 *Bacterio − rhodopsin* for Optoelectronic Circuitry

Early use of molecules in information processing has been in the field of optical computing. This suggested as candidate for EC base material, the Bacterio-rhodopsin, BR, which can serve as computer switch (Birge 1995, Mann 1996, Vsevolodov 1998).

BR has two useful properties for molecular level calculation. It exhibits photo chromic switching and shows photoelectric effect also.

The photo-cycle of BR, the sequence of structural changes induced by light-allows the storage of data in memory. Green, red and blue light induce structural changes of BR. Green light transforms BR in an intermediate denoted by “k” that relaxes to the “o” state. Red light transforms “o” state in “p” state that relaxes to “q” state. Blue light converts “q” state back to BR (Birge 1995). Any long lasting states can be assigned to digital values making possible to story information as a series of BR molecules in one or another state.

Discrete states as “0”, “1” and so on, allows operating the EC devices. With these identifications the BR substrate may be the source for the symbolic language such as pixels and strings.

The photoelectric effect is another BR property useful for EC realization. Practical use of this property is exigent because it requires the preparation of BR films with highly oriented molecules. The possibility to interface BR electrically is the basis for several applications. The light of a specific wavelength range can be used to change the BR conformational state and the conformation change is accompanied by a color change that can be detected by optical means. It should be observed that the circuits are in this case, at least in part, of optical type.

A significant step in the development of the optoelectronic circuitry and computing was the study of all-light-modulated transmission mechanism of BR films. When a yellow beam and a blue beam illuminate the BR film, the two transmitted beams suppress mutually. Based on this mechanism, an all-optical operating device in which all 16 kinds of double-variable binary logic operations was implemented. The intensity of an incident yellow or blue beam acts as the input to the logic gate and the transmission bears the output of the gate. It is possible to turn this all-optical device into different states using different wavelengths and different intensity illuminations.

Full evolvability of hybrid symbolic connectionist models and associated circuitry that may be based on the unique properties of BR will be evaluated in the following.

4.4.2.4 Embedded EDOE

The perspectives of a hybrid optoelectronic device based on BR molecules properties, in which conventional electronics is used to implement DOE analysis, are evaluated in the following.

Photo-cycle and photoelectric effects allows a direct writing DOE embedding in the BR based substrate. BR memorizing digits should be complemented by standard electronics able to perform the real valued operations.

As shown in Sect. 2.2.4, the DOE are resulting as particular solutions of the wave equation. Consider the solution:

$$Y(T, Z) = Z \oplus (V \otimes T) \tag{4.9}$$

A computing "cell" with three BR molecules is retained here for illustration purposes. Table 4.12 shows the wave solution for V=1.

Table 4.12 Convection model: Y (T, Z)

Z\T	0	1	2
#0	0	1	2
#1	1	2	0
#2	2	0	1

Table 4.13 DOE associated to three molecules cell

Exp	Molec.	Time	Operation
1	#0	0	g
2	#0	1	r
3	#0	2	b
4	#1	0	r
5	#1	1	b
6	#1	2	g
7	#2	0	b
8	#2	1	g
9	#2	2	r

The Table 4.12 is resulting by Galois field, GF (3) calculations and is a 3x3 Latin-square. The factors are the time steps 0, 1, 2, the molecules #0, #1, #2 and the operations 0=g, 1=r, 2=b corresponding to the three colors green, red, blue able to induce transitions. The time is multiple of the same time-step.

Standard DOE table may be developed by indicating the conditions associated to any element of the 3x3 Latin square (Table 4.13). Experimental results of DOE application may be the resolution, or any other value or data to be memorized. The DOE selects the significant results and also the significant factors by standard ANOVA calculations done by an external computer. This is Fourier analysis over the real field, for the device functioning parameters.

Successive steps will continue the experiment in the direction of beneficial results. The new experiment means a new DOE based on GF(m) algebra calculation and the wave equation. Following the EDOE suggestion, hardware may be achievable in 2-D or 3-D structures with concentric hierarchically located levels or planes. Light sources should be placed externally (Birge 1995).

Based on special BR properties, new classes of evolvable circuits, embedding and evolving DOE became possible. The evolvability, for the proposed architectures is the challenged result. Among the possible sets of DOE matrices, for n runs, m factors, and s settings we select the Walsh matrix of design, Wn,m,s, or Latin square matrices Ln,m,s generated by first order wave equation, WE (Iordache 2009). As for the EDOE structures, after the implementation of the DOE matrix of the type Wn,m,s or Ln,m,s it is required to perform at least two steps: factor evaluation, on columns in DOE, and experiment classification, on rows in DOE. It is necessary to define thresholds as degrees of acceptability for results. This help to decide when to recognize a pattern to be classified, as new. Various areas throughout the chosen EC layers may be written and addressed simultaneously. It is conceivable to embed Wn, m, s or Ln,m,s matrices in any active areas with memory. EC would be built using in succession similar additive and subtractive steps

as for printed circuits and integrated circuits fabrication. Matrices such as Wn,m,s or Ln,m,s play the role of masks in printed or integrated circuits fabrication. These evolvable circuits should be able to drive the input signal and to decode the signal in a manner similar to logical thinking processes. As a difference, if compared to conventional circuits, this kind of EC will be continuously formed and erased, allowing the operation to be in succession forward and backward. The parallel search may be organized to achieve amplification, resonance and coherency. The EC works associatively as well as serially. By parallel processing the experiments would be performed at once, and the recorded results can be presented simultaneously to the center DOE. The EC should be able to record data from different areas to analyze and to give rise to a decision. This means that EC need to have monitoring functions, that is sensors, and executive functions, that is actuators, since the long term technological challenge is to get results by EC, independent of any external analyst or "operator". The EC should be a system that confronts the environment having the ability to evolve autonomously. New environmental conditions for EC may be materialized by a new row in the existing, embodied, component DOE matrices. This is the discrete symbolic step of the EC. Then follows the step in which real field values are associated to discrete DOE. This real valued step goes after data expansion and precedes data compression. With a learned degree of acceptability the sensor information goes backward and is classified in inner levels or layers and finally come back in the center. In this way the material embodiment may regenerate the symbolic description represented by DOE.

4.4.2.5 Hybrid Controlled Micro-fluidic Circuits

In micro-fluidic devices the circuitry from printed or integrated circuits is replaced or completed by micro-channels for fluids. The MEMS became in fact MECS (Mahalik 2005). The transport of molecules in complex biological or chemical process may be programmed as the electric current in standard electronic circuits (van Noort et al. 2002, Verpoorte and de Rooij 2003, Erickson and Li 2004).

The micro-fluidic devices supposes the existence and the development of sensors, able to monitor changing environment, of actuators able to influence environment, coupled with computing and control capabilities for communication and data processing, all physically wired together. Tangen et al. (2006) presented elements of an interesting development in this direction. It focuses on the application of on-line programmable micro-fluidic bio-processing as a vehicle towards the design of artificial cells.

The electronically controlled collection, separation and channel transfer of the bio-molecules are monitored by sensitive fluorescence setups. This makes combinatorial fluidic circuitry and biochemical reactions circuitry feasible.

The basic elements of the SKUP quadruple may be identified for the "biomolecular console" described by Tangen et al. (2006). The reconfigurable

electronic interface is linked to the space of conditions, K. The micro-fluidic network represents the states-S. This includes chemicals reservoirs and products. The parallel actuator network is related to operators U, while the monitoring system is linked to possibilities P. An electronic computer guides and controls the molecular circuits and ensures the cyclic functioning.

Another promising micro-fluidic technology consisting of a fluidic layer with a network of micro-channels superposed on layer with external computer programmable electrodes and actuators controlling the flow, has been proposed by Goranovic et al. (2006).

The project applies micro-fluidic nano-techniques to programming molecular reactions and priming an evolution of artificial cells and cell assemblies,

The basic elements of the SKUP are obvious for this technology. The genetic channel is linked to the space of conditions K. The temperature cycles ensures the gene replication. This fits with the cyclic character of the time T.

The metabolic channel is naturally linked to states S. The replication of selected proto-cells is linked to operators U, while the metabolism of selected proto-cell step is related to the possibilities P. The switch from categorical product to coproduct is determined by the oscillatory temperature fields and is able to control the proto-cell replication.

A categorical presentation of the architecture is shown in Fig. 4.9.

An important specificity of this micro-fluidic device is the realization of closed or loop operations, essential for the transition from fixed circuits to evolutionary and then to fully evolvable circuits.

4.4.2.6 Self-constructed Molecular Circuits and Computing

Self-construction and separation in classes may be considered as computational processes and may be utilized to build information processors. Observe that the basic elements of the SKUP quadruple are naturally associated to any self-construction or separation processes. Suppose that from an unstructured environment S, some molecules considered as symbols are able to assembly in a supra-molecular structure linked to the conditions space K. These K structures should be recognized by a receptor and possibly amplified to provide an action U, redirected towards the unstructured environment S. The selection of specific symbols from the environment is done according to possibilities P. This may be a process driven by an optimization criterion as for instance energy or entropy production minimization or maximization (Prigogine 1980, Dewar 1993).

The self-construction may be described by the WE, too. According to the interpretation of the tensor product two main types of configurations are resulting. The tree-like forms are resulting for if the tensor product is a categorical product and a multiple cells stacked configuration are to be expected if the tensorial product is a coproduct. The transition between the two configurations is mediated by the environment conditions.

A categorical presentation of the architecture is shown in Fig. 4.9.

Elements of the general scheme of self-constructed computing are present in different DNA experiments (Adleman 1994, Winfree 2000).

Adleman proposed an approach to information processing with bioprocesses that allowed solving combinatorial problems by making use of specific set of DNA molecules.

DNA-based computing consists of four basic operations: encoding, hybridization, ligation and extraction. Problem solutions are obtained through an exhaustive parallel search by means of the pattern recognition intrinsic to DNA hybridization that is to self construction of complementary DNA strands. Involved chemical reactions such as the activity of restriction enzymes, ligases polymerases or simple hybridization can operate in parallel and this explains the possibility to solve complex problems.

Following similar ideas, cellular automata architectures describing DNA self-constructed circuit patterns for various forms of DNA tiles have been studied by Winfree (2000). Cook et al. (2004) showed how several common digital circuits, including de-multiplexers, random access memory, and Walsh transforms, could be built in a bottom-up manner using biologically inspired self-construction.

The Walsh-Hadamard matrices may be obtained as particular solutions of the wave equation WE. Table 4.14 shows a solution of the kinetic model in which we suppose the rate Q to be constant in the wave equation WE.

It is in fact the so-called Hadamard-Sylvester matrix, similar to Sierpinski triangle as presented by Cook et al. (2004). To highlight this parallelism the bolding and underlining is used for the wired "1" cells. It was assumed that "-1" breaks the circuitry. The non-wired digits are italicized. Notice that only two digits "0" (replaced here by "-1") and "1" need to be present in this case.

Based on operations in GF (4) Sierpinski square like fractals may be generated (Carbone and Seeman 2002 a, 2002 b).

At the present stage, a number of researchers are rather skeptical whether existing DNA based computation strategies will ever follow Bacterio-rhodopsin, BR, on its path in information processing. Critical problems with DNA

Table 4.14 Kinetic model, Y (T)

Q\T	000	001	010	011	100	101	110	111
000	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
001	**1**	*-1*	**1**	*-1*	**1**	*-1*	**1**	*-1*
010	**1**	**1**	*-1*	*-1*	**1**	**1**	*-1*	*-1*
011	**1**	*-1*	*-1*	**1**	**1**	*-1*	*-1*	*1*
100	**1**	**1**	**1**	**1**	*-1*	*-1*	*-1*	*-1*
101	**1**	*-1*	**1**	*-1*	*-1*	*1*	*-1*	*1*
110	**1**	**1**	*-1*	*-1*	*-1*	*-1*	*1*	*1*
111	**1**	*-1*	*-1*	*1*	*-1*	*1*	*1*	*-1*

circuits and DNA-bio-computers are related to their inflexibility and to the ineffective accommodation to the variety of computation requests in real conditions. The assembly of DNA molecules of "tiles" has been designed to simulate the operation of any Turing machine. The self-construction DNA structures may be mapped naturally onto the grammars of the Chomsky hierarchy (Chomsky 1966, Winfree 2000). However for strictly algorithmic operations the DNA tiling computer can't compete with silicon machines. It was expected that the DNA computers may be eventually advantageous for the complementary domain of computation, beyond Turing machine capability. Again, it is not the case for the self-assembled DNA circuits as much as they map the Chomsky hierarchy of grammars.

The 1-D chain of DNA or the 2-D crystal tiles represent only the informational that is the K part of the SKUP. Major parts of the actual DNA computation have been accomplished with human operator involvement. To solve complex problems the K structure should be part of SKUP quadruple. K elements should be recognized by a receptor and amplified to provide an action U towards the external non-assembled environment S. The selection of specific symbols from the environment S would be done according to the possibilities P. For this reason the tilling need to be flexible and the tiles could be cycled through alternating assembly and disassembly stages. The self-construction and reconstruction operation may be programmable using glued and un-glued tiles (Carbone and Seeman 2002 b). According to the signification of the tensor product in WE solution, two main types of configurations are resulting. The tree-like forms are resulting if the tensor product is a categorical product and a multiple tiles stacked configuration is to be expected if the tensorial product is a categorical coproduct. The switching from one tensor interpretation to another is induced by environment changes.

A categorical presentation of the architecture is shown in Fig. 4.9.

Interactions between tiles and between tiles and their environment are mandatory to challenge Turing machines.

4.4.2.7 Conventional Circuits versus Evolvable Circuits

For the forthcoming evolutionary and evolvable circuits fabrication a natural query is why do not use traditional methods, such as the physical and chemical study, followed by the modeling and extrinsic implementation of the models in the usual computer based control of circuit fabrication. The answer is that the envisaged control and computing task are impossibly to be extrinsically operated for evolvable systems of high complexity. In conventional circuits design the majority or non-linear interactions that could possibly contribute to the problem are deliberately excluded. The properties characterizing EC constructions should be, at least in part, the consequences of their own dynamic of the computational environment, not of the decision

Table 4.15 Comparison of conventional circuits and evolvable circuits

Conventional circuits-PC	Evolutionary or evolvable circuits
Single objective for any fabrication step	More general classes of objectives
Defined-based on previous learning	Undefined-open for learning, innovative
Top-down, linear	Top-down, bottom-up, cyclic, multi-scale
Aims for best solution, optimal	Makes workable, evolvable, active
Looks for perfect elements	Accepts elements with small defects
Conventional design-detailed models	Generic design-based on wave equation
Clear processing steps, complete data	Incomplete data and variable ad-lib steps
Independent on previous designs	Use everything at hand, if useful
Insulate the elements, serial or sequential	Combine elements, distributed, parallel
Builds	Builds, disbands, embeds and reorganizes
Divide and conquer	Divide and integrates, opportunistic
Maintain functionality in different media	Sensitive to environment, multifunctional
Restricted, static	Less restricted, rich, dynamic
Isolate from medium-protection	Medium, opportunistic exploitation
Avoid variability, interactions, transitions	Accept, use variability, interactions
Reliable	Robust, multi-reliable
High maintenance	Low and proactive-maintenance
Catastrophic degradation	Degradation in steps, hindered

of the designer who is anyway unable to predict the evolution of its construction. EC are supposed to work for their evolution more efficiently than an external computer or operator can do. EC has the potentiality to be developed towards an autonomous system allowing survivability in completely unforeseen situations. It was observed that the more an autonomous system is optimized to perform a task the less capability it has to deal with unexpected changes in an uncertain environment. This implies that in complex environments, evolvability rather than adaptability or versatility may be an appropriate measure of a circuit's potential to carry out tasks.

Complex systems, natural or artificial, seem to opt for evolvability rather than for optimization and adaptability. This may be because in a complex environment it is impossible to achieve the optimum particularly when there are strong interactions between conditions K and states S. One way to proceed is to diversify several acceptable circuit options in a given environment and to let them evolve. This means that evolvable circuits may have several possible non-optimal but acceptable and useful architectures. This implies the discovery of environment properties that can be utilized to solve the imposed tasks.

Table 4.15 summarizes some of the differences between conventional circuits and unconventional ones such as evolutionary circuits and evolvable circuits, EC.

At the present technological level, a project grouping in a manufacture and in a product all the described faculties of EC is unrealistic but it is expected to manufacture EC of increasing capability in small steps.

4.4.3 Perspectives

4.4.3.1 Evolutionary Devices

The categorical architectures shown in Fig. 4.19 may be considered for the study of evolutionary devices. Similar frames have been described by Cariani (Cariani 1989, 1998). The three levels are outlined in Fig. 4.19.

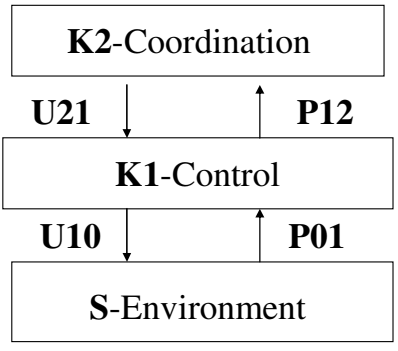


Fig. 4.19 Three levels framework for evolutionary devices

Cariani shows that a hierarchical framework similar to that from Fig. 4.19, are embedded in the internal structure of individual organisms and evolutionary devices. The elements of the categorical framework are as follows:

S corresponds to the environment; K is structured in two levels to allow performing integrated neurogonitive/behavioral tasks.

K1-Evaluation and control level
K2-Coordination and decision level
U10: K1 → S control actions, decoding
P01: S → K sensor and measuring devices, encoding
U21-effectors resources
P12-evaluation/selection capability

The resulting device is able to evolve new goals to have a creative direction.

The same general framework corresponds to the scientific or engineering methodology. While this method includes only measurements and computations, organisms and evolvable device may act directly on the environment through effector organs of devices. The effectors convert signs into action on the material world. This corresponds to the control U. The basic informational operations of signs (semantic functionalities) present in organisms and devices can be described in terms of measurement (sensing), computation (coordination) and effecting (action). For living organisms, capable of neurogonitive behavior this was described by von Uexküll (1973).

It would be interesting to compare Fig. 1.5 showing the functional cycle with Fig. 4.19 and Fig. 4.9 showing two categorial frames for conditions, K1

and K2 with two basic interpretations of tensorial product, the coproduct "∪" for K1 and the product "×" for K2.

The interaction between S and K1 corresponds to a 1^{st} order evolutionary step, while that between K1 and K2 to a 2^{nd} order evolutionary step.

Many architectures proposed as evolutionary designs are based on less than four levels and may be considered as still incompletely developed.

The Pask's evolutionary device and the evolved radio appear to lack some elements of the top levels and also a link between coordination level K2 and environment S.

The missing levels and links may induce severe limitations and prevent this device to become evolvable.

4.4.3.2 Three Realms Frameworks and Molecular Computation

There are several molecular computation studies suggesting how to design synthetic chemical or biochemical circuitry able to perform specified algorithms (Miller, 2008).

A method to make use of molecules in computing architectures was by reproduction of computer solid-state components with molecular structure. This is the usual approach taken in molecular electronics research. Typical objectives are the molecular wires, rectifiers or transistors (Siegmund et al. 1990). Another research direction was the chemical computing based on the fact that chemical reaction networks are able to process information in parallel. Kuhnert et al. (1989) demonstrated contour detection, contrast enhancement and same image processing operations on images projected onto a thin layer of a light-sensitive variant of chemical waves reaction medium. This system is a chemical realization of an associative memory and suggests the potential to implement learning networks by chemical means. The research into parallel chemical information processors led to artificial neural network, NN design based on mass-coupled chemical reactors (Hjelmfelt et al. 1992). Real chemical computing employs real chemical processes to compute. For example the simplest nonlinear function XOR can be implemented with reaction-diffusion behavior of palladium chloride (Adamatzky and Costello 2002).

Studies in molecular and supra-molecular design and engineering opened the perspectives for the realization of electronic, ionic and photonic circuits (Lehn 2004). Orchestrated, supra-molecular architectures deliberately designed to carry information allow to accelerate and to direct molecular interactions.

Artificial chemistry and organic computing suggests innovative ways to go beyond the chemical kinetic level and encompass supra-molecular interactions (Dittrich et al. 2001, Dittrich 2005). Interesting projects involve genetic programming (Harding 2005, Miller 2008).

Genetic programming GP, introduced by Koza (1992) is a development of genetic algorithms, GA methods. In GP the operations are as in GA but

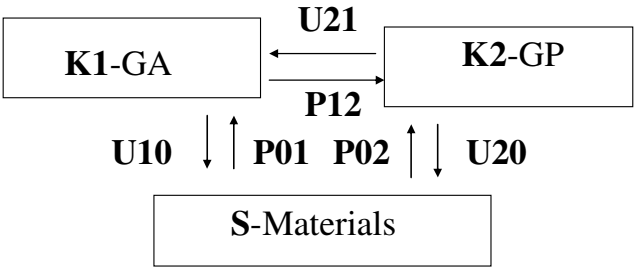


Fig. 4.20 Genetic programming framework

on populations of programs not on strings. The behavior of each program in population is evaluated using a fitness function. Programs that do better are copied into next generation.

Fig. 4.20 outlines a three realms categorical framework for evolutionary computation.

The notations are as follows: S-Materials, K1-GA, genetic algorithm, K2-GP, genetic programming. The frame is useful for the study of evolution in *materio* (Miller and Downing 2002, Harding 2005).

S represents the material substrate. K1 and K2 are conditioning levels. K1-represents the basic genetic algorithm GA. K2-is the meta-level representing the genetic programming, GP. A change in K2 has higher impact because it represents multiple changes at previous realm K1.

The categorical framework describes interactions as: U10: GA → Materials, U20: GP → Materials, P01: Materials → GA, P02: Materials → GP.

What is interesting in this three realm architecture is the connection between K2 and S that is, between meta-model level and materials. This computation in *materio*, can ensure evolvability and autonomy. For this reason such frameworks were proposed for extracting computation from physical systems, for autonomous experimentation (Lovell et al. 2009).

4.4.3.3 Four Level Frameworks

Driven by the continuously changing environment, living beings developed hierarchical self-repair and self-replicating mechanisms. Embryonics project brings the worlds of biology and electronics closer, by implementing in silicon these features.

Progresses have been reported in the construction of multi-cellular self-replicating systems (Mange et al. 2004). This is significant since one of the characteristic of evolvability is self-reproduction. Mange and coworkers proposed the "Tom thumb" algorithms that made possible to design self-replicating loops with universal construction and universal computation that can be easily embeddable into silicon.

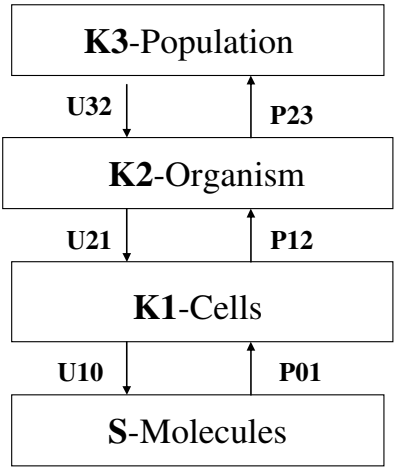


Fig. 4.21 Four level organization of embryonics system

Mange et al. (1998) proposed a bio-inspired architecture for evolutionary electronic devices.

Embryonics bio-inspired devices are made up of four hierarchical levels (Fig. 4.21).

The multi-scale structure in embryonics project was correlated to the four levels of organization analogous to molecules, cells, organisms and population. The molecular level is represented by the basic field programmable gate array, FPGA elements.

The FPGA is the molecule of the devices. The FPGAs can be put together through a set of programmable connections to realize different types of digital circuits.

Each cell is a simple processor for instance a binary processor realizing a unique function within the organism, defined by a set of instructions.

The organism level is an association of cells while the population level is an association of organisms. The functionality of the organism is obtained by the parallel operation of all the cells. The size that is, the number of cells of an organism is also programmable and given enough space the organisms replicate automatically. Since the functionality of an organism is identical in each replicated copy, this mechanism provides an intrinsic fault tolerance. Given an appropriate cell structure the organisms are capable of learning.

In living cells, the genetic information is processed sequentially. Designing a memory that is inspired by biology suggests a different type of memory, called cyclic memory. Cyclic memory does not require any addressing mechanisms. Instead it consists of a simple storage structure that circulates synchronously its data in a closed circle, much as the ribosome processes the genome inside a living cell.

Observe that for the embryonics project, an architecture showing the integrative closure mappings between the top level K3 and the lower level denoted by S is still missing.

The embryonics architecture is restricted to the 3^{rd} order evolutionary step. For this reason the embryonics project may be considered only as evolutionary devices rather then fully evolvable, autonomous devices.

Critical for EC autonomy is the embodiment of the computing capacity that is the interconnection between K3 and S realms. This is the 4^{th} order evolutionary step.

Strategies to correlates K3 to S both in programming as in fabrication are suggested by organic computing studies (Müller-Schloer et al. 2004). In this general frame Pietzowski et al. (2006) proposed a system that use the paradigm of antibodies and developed the organic computing middleware system.

For the organic computing middleware architecture, the four levels may be identified as: S-Transport connector interface, K1-Event dispatcher, K2-Service interface, K3-Organic manager. Existing computational systems can be redesigned and redeveloped to engineer evolvable capabilities into them. Evolvability capabilities have to be added gradually and incrementally as organic computation studies suggests. Complete evolvability may be attained only step by step.

4.4.3.4 n-Graphs Organization of Immuno-embryonic Systems

In order to create technological systems that are autonomous robust and evolvable, new engineering approaches must draw inspiration from natural complex systems.

For example in computer security, systems able to mimic the biological immune system can provide solutions against attacks on computer networks.

The immune system has been a major source of inspiration in the design of pattern recognition applications including computer security and virus protection.

Inmunotronics is another bio-inspired concept that has been successfully implemented in evolutionary hardware (Bradley et al. 2000).

Immunity is a multi-layered and multi-scale architecture starting with physical barriers, through physiological barriers, through cellular interactions.

Antibody mediated immunity protects the body from bacteria using B cells to generate antibodies and helper T cells to activate the production of antibodies.

The geometrical shape plays a crucial role for this type of immunity.

The embryonic cells proposed in embryonics lack a real-time method of verifying that each is performing the correct operation with respect to neighboring cells.

Bradley et al. (2000) proposed to incorporate emrbryionic and immune cells.

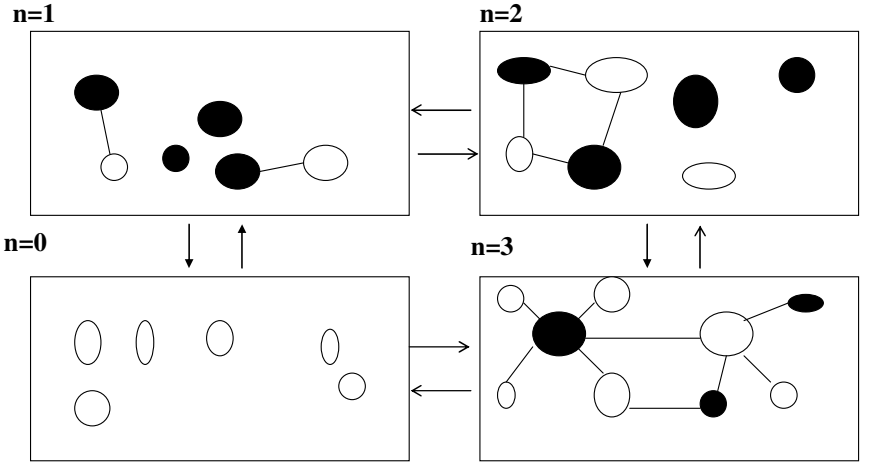


Fig. 4.22 n-graphs for immuno-embryonics framework

Fig. 4.22 illustrates a potential development cycle for architecture using n-graphs (Appendix A5). It is an appropriate tool for multi-scale systems study.

The immune cells are black and the embryonic cells are white.

At n=0 the cells, prepared to be immune or embryonic are isolated. At n=1 interactions and couples cells appear. This is allowed by interaction within the body. Cells are separated in the stage n=1 but they interact in the stage n=2 to form arrays of interacting cells. The stage n=2 shows the coupling of two or more cells in frames going beyond cells areas isolation. It describes interactions of interactions.

The final stage, n=3 corresponds to a kind of global action of the whole immuno-embryonics architecture. Whole architecture pattern allow to identify faults and to review critical cases. The integrative closure including the interconnection of stages n=0 and n=3 represents the challenge for such systems.

The n-graphs are naturally correlated to n-categories (Appendix A5).

Katis et al. (2000) proposed symmetric monoidal categories with feedback as appropriate modeling frameworks for concurrent and distribute processes as those shown in Fig. 4.22.

In this case the objects in category are the cells, immune or embryonic. Their interconnections represent the relations. In bicategories the objects are cells, the relations between cells corresponds to 1-graphs, and the relations between relations to the 2-graphs. There are two different compositions of 2-cells, the vertical and the horizontal.

Notice that Katis et al. (2000) study is restricted to the 2-graphs that is, to 2-categories.

The tricategorical development would include the 3-graphs as a step towards the integrative closure.

References

Adamatzky, A., Costello, B.D.L.: Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond. Phys. Rev. E 197(22), 344–352 (2002)

Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science 266(5187), 1021–1024 (1994)

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J.D.: Molecular Biology of the Cell. Garland Publishing, NY (1994)

Ardell, D.H., Sella, G.: No accident: genetic codes freeze in error-correcting patterns of the standard genetic code. Philos. Trans. R Soc. London B Biol. Sci. 357(1427), 1625–1642 (2002)

Austin, J.: Distributed associative memories for high speed symbolic reasoning. International Journal on Fuzzy Sets and System 82(2), 223–233 (1996)

Baeck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: Comments of the history and current state. IEEE Trans. Evolutionary Computation 1, 3–17 (1997)

Balakrishnan, K., Honavar, V.: Evolutionary design of neural architectures-A preliminary taxonomy and guide to literature. Department of Computer Science Technical Report 95-01 (1995)

Banzhaf, W.: Self-replicating sequences of binary numbers, foundations I. General, Biological Cybernetics 69, 269–274 (1993)

Barnsley, M.F.: Fractals Everywhere. Academic Press, New York (1993)

Bedau, M., McCaskill, J.S., Packard, N.H., Rasmussen, S., Adami, C., Green, D.G., Ikegami, T., Kaneko, K., Ray, T.S.: Open problems in artificial life. Artificial Life 6, 363–376 (2000)

Benyo, B., Biro, J.C., Benyo, Z.: Codes in the codons: Construction of a codon/amino acid periodic Table and a study of the Nature of specific nucleic acid-protein interactions. In: Proceedings of the 26th Annual International Conference of the IEEE EMBS, San Francisco, pp. 2860–2863 (2004)

Bird, J., Layzell, P.: The Evolved Radio and its Implications for Modeling the Evolution of Novel Sensors. In: Proceedings of the Congress on Evolutionary Computation, CEC 2002, pp. 1836–1841 (2002)

Birge, R.: Protein-based computers. Sci. Am. 272, 90–95 (1995)

Bradley, D.W., Ortega-Sanchez, C., Tyrrel, A.M.: Embryonics+Immunotronics: A bioinspired approach to fault tolerances. In: Proceedings of the 2nd NASA/DoD Workshop on Evolvable Hardware (2000)

Bray, D.: Protein molecules as computational elements in living cells. Nature 376, 307–312 (1995)

Burgess, N., Recce, M., O'Keefe, J.: A model of hippocampal function. Neural Networks 7, 1065–1081 (1994)

Caporale, L.H.: Is there a higher level genetic code that directs evolution? Molecular and Cellular Biochemistry 64, 5–13 (1984)

Carbone, A., Seeman, N.C.: Circuits and programmable self-assembling DNA structures. PNAS 99, 12577–12582 (2002a)

Carbone, A., Seeman, N.C.: A route to fractal DNA assembly. Natural Computing 1, 469–480 (2002b)

Cariani, P.: On the Design of Devices with Emergent Semantic Functions. Ph D Dissertation, Binghamton University (1989)
Cariani, P.: To evolve an ear: epistemological implications of Gordon Pask's electrochemical devices. Systems Research 10, 19–33 (1993)
Cariani, P.: Towards an evolutionary semiotics: the emergence of new sign functions in organisms and devices. In: Van de Vijver, G., Salthe, S., Delpos, M. (eds.) Evolutionary Systems, pp. 359–377. Kluwer, Dordrecht (1998)
Cariani, P.: Symbols and dynamics in the brain. Biosystems 60, 59–83 (2001)
Carpenter, G., Grossberg, S.: ART2: Selforganization of stable category recognition codes for analog input patterns. Applied Optics 26, 4919–4930 (1987)
Cheung, P., Berlin, A., Biegelsen, D., Jackson, W.: Batch fabrication of pneumatic valve arrays by combining MEMS with PCB technology. In: Symposium on Micro-Mechanical Systems, ASME, Dallas, TX, 62, pp. 39–46 (1997)
Chomsky, N.: Cartesian Linguistics. Harper & Row, New York (1966)
Cook, M., Rothemund, P.W.K., Winfree, E.: Self-assembled circuit patterns. In: Chen, J., Reif, J.H. (eds.) DNA 2003. LNCS, vol. 2943, pp. 91–107. Springer, Heidelberg (2004)
Damasio, A.R.: The brain binds entities and events by multiregional activation from convergence zones. Neural Comput. 1, 123–132 (1989)
Danckwerts, H.J., Neubert, D.: Symmetries of genetic code-doublets. J. Mol. Evol. 5, 327–332 (1975)
Dasgupta, D., McGregor, D.R.: Designing Neural Networks using the Structured Genetic Algorithm. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN), Brighton, UK (1992)
Dewar, R.C.: Informational theory explanation of the fluctuation theorem, maximum entropy production, and self-organized criticality in non-equilibrium stationary states. J. Phys. A 36, 631–641 (2003)
Di Marzo Serugendo, G., Fitzgerald, J., Romanovsky, A., Guelfi, N.: A Generic Framework for the Engineering of Self-Adaptive and Self-Organising Systems. Technical Report CS-TR-1018, School of Computing Science, University of Newcastle (2007)
Dittrich, P.: The Bio-Chemical Information Processing Metaphor as a Programming Paradigm for Organic Computing. In: ARCS Workshops, pp. 95–99 (2005)
Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries-a review. Artificial Life 7(3), 225–275 (2001)
Doursat, R.: A Contribution to the Study of Representations in the Nervous System and in Artificial Neural Networks. Ph D Thesis, University Paris (1991)
Doursat, R.: Of tapestries, ponds and RAIN. Toward fine-grain mesoscopic neurodynamics in excitable media. In: International Workshop on nonlinear brain dynamics for computational intelligence. Salt Lake City, USA (2007)
Doursat, R., Bienenstock, E.: Neocortical self-structuration as a basis for learning. In: 5th International Conference on Development and Learning, ICDL 2006, Indiana University, Bloomington, IN (2006)
Eigen, M., Schuster, P.: The hypercycle a principle of natural self-organization. Springer, Berlin (1979)
Eigen, M., Winkler-Oswatitsch, R.: Transfer-RNA: The early adaptor. Naturwiss 68, 217–228 (1981)
Elman, J.L.: Finding structure in time. Cognitive Science 14, 179–211 (1990)
Elman, J.L.: Learning and development in neural networks:The importance of starting small. Cognition 48, 71–99 (1993)

Erickson, D., Li, D.: Integrated microfluidic devices. Analytica Chimica Aca. 507, 11–26 (2004)
Findley, G.L., Findley, A.M., McGlynn, S.P.: Symmetry characteristics of the genetic code. Proc. Natl. Acad. Sci. USA 79, 7061–7065 (1982)
Fodor, J.A., Pylyshyn, Z.: Connectionism and cognitive architecture: a critical analysis. Cognition 28(1/2), 3–71 (1988)
von Foerster, H.: What is memory that it may have hindsight and foresight as well? In: Bogoch, S. (ed.) The Future of the Brain Sciences, pp. 19–65, 89–95. Plenum Press, New York (1969)
Forcey, S.: 2-fold operads Young diagrams and dendritic growth, University Research Symposium, Tennessee State University (2008)
Freeman, W.J.: Neurodynamics. An exploration of mesoscopic brain dynamics. Springer, London (2000)
Fritzke, B.: Growing cell structure-a self organizing network for unsupervised and supervised learning. Neural Networks 7(9), 1441–1460 (1994)
Geard, N., Wiles, J.: Structure and dynamics of a gene network model incorporating small RNAs. In: IEEE Congress on Evolutionary Computation, pp. 199–206 (2003)
Goranovic, G., Rasmussen, S., Nielsen, P.E.: Artificial life forms in microfluidic systems. In: Proceedings microTAS 2006, Tokyo, Japan, vol. 2, p. 1408 (2006)
Gould, S.J.: The paradox of the first tier: an agenda for paleobiology. Paleobiology 11, 2–12 (1985)
Halford, G.S., Wilson, W.H., Phillips, S.: Processing capacity defined by relational complexity. Implications for comparative, developmental and cognitive psychology. Behavioural and Brain Sciences 21(6), 803–831 (1998)
Harding, S.: Evolution in Materio. Ph D Thesis University of York (2005)
Haronian, D., Lewis, A.: Elements of a unique bacteriorodopsib neural network architecture. Applied Optics 30, 597–608 (1991)
Hartman, H.: Speculations on the evolution of the genetic code. Origins of Life 6, 423–427 (1975)
Healy, M.J., Caudell, T.P.: Ontologies and worlds in category theory: Implications for neural systems. Axiomathes 16(1), 165–214 (2006)
Hjelmfelt, A., Weinberger, E.D., Ross, J.: Chemical implementation of finite-state machines. PNAS USA 89, 383–387 (1992)
Hummel, J.E., Biederman, I.: Dynamic binding in a neural network for shape recognition. Psychological Review 99, 480–517 (1992)
Hummel, J.E., Choplin, J.M.: Toward an integrated account of reflexive and reflective reasoning. In: Gleitman, L., Joshi, A.K. (eds.) Proceedings of the Twenty Second Annual Conference of the Cognitive Science Society, pp. 232–237. LEA, Mahwah (2000)
Hummel, J.E., Holyoak, K.J.: Distributed representation of structure. A theory of analogical access and mapping. Psychological Review 104, 427–466 (1997)
Inhelder, B., Piaget, J.: The Growth of Logical Thinking from Childhood to Adolescence. Basic Books, New York (1958)
Iordache, O.: Evolvable Designs of Experiments Applications for Circuits. J. Wiley VCH, Weinheim (2009)
Jacob, F., Monod, J.: On the regulation of gene activity. Cold Spring Harbor Symp. Quant. Biol. 26, 193–211 (1961)
Jimenez-Montano, M.A.: Protein evolution drives the evolution of the genetic code and vice versa. Biosystems 54, 47–64 (1999)

Jimenez-Montano, M.A., de la Mora-Basanez, R., Poschel, T.: Poschel T The hypercube structure of the genetic code explains conservative and non-conservative amino acid substitutions in vivo and in vitro. Biosystems 39, 117–125 (1996)
Kargupta, H.: A striking property of genetic-code like transformations. Complex Systems 13, 1–32 (2001)
Katis, P., Sabadini, N., Walters, R.F.C.: On the algebra of systems with feedback & boundary. Rendiconti del Circolo Matematico di Palermo Serie II, Suppl. 63, 123–156 (2000)
Keller, R., Banzhaf, W.: The evolution of genetic code in genetic programming. In: Proc. Genetic and Evolutionary Computation Conf., pp. 077–1082. Morgan Kaufmann Publishers, San Francisco (1999)
Kitano, H.: Designing neural network using genetic algorithm with graph generation system. Complex Systems 4, 461–476 (1990)
Koonin, E.V., Novozhilov, A.S.: Origin and evolution of the genetic code: The universal enigma. IUBMB Life 61(2), 99–111 (2009)
Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
Kuhn, H., Kuhn, C.: Diversified world: drive of life's origin? Angew. Chem. International 42, 262–266 (2003)
Kuhn, H., Waser, J.: Molecular self-organization and the origin of life. Angew. Chem. Int. Ed. Engl. 20, 500–520 (1981)
Kuhn, H., Waser, J.: Hypothesis on the origin of genetic code. FEBS Letters 352, 259–264 (1994)
Kuhnert, L., Algadze, K.I., Krinsky, V.I.: Image processing using light-sensitive chemical waves. Nature 337, 244–247 (1989)
Lehn, J.M.: Supramolecular chemistry- from molecular information towards self-organization and complex matter. Reports on Progress in Physics 67, 249–265 (2004)
Lindenmayer, A.: Mathematical models for cellular interaction in development, I, II. J. Theor. Biol. 18, 280–315 (1968)
Lovell, C.J., Jones, G., Zauner, K.P.: Autonomous Experimentation: Coupling Machine Learning with Computer Controlled Microfluidics. In: ELRIG Drug Discovery, Liverpool, September 7-8 (2009)
Maass, W.: Liquid computing. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, pp. 507–516. Springer, Heidelberg (2007)
Macias, N.J.: The PIG paradigm: The design and use of a massively parallel fine grained self-reconfigurable infinitely scalable architecture. In: Proceedings First NASA/DoD Workshop on Evolvable Hardware (1999)
Mahalik, N.P.: Micromanufacturing and Nanotechnology. Springer, New York (2005)
Mallot, H.A.: Spatial cognition: behavioral competences, neural mechanisms and evolutionary scaling. Kognitionwissenchaft 8, 40–48 (1999)
von der Malsburg, C.: Am I thinking assemblies? In: Palm, G., Aertsen, A. (eds.) Brain theory, pp. 161–176. Spinger, NY (1986)
von der Malsburg, C.: The correlation theory of brain function. In: Domany, E., van Hemmen, J.L., Schulten, K. (eds.) Models of Neural Networks II: Temporal Aspects of Coding and Information Processing in Biological Systems, pp. 95–119. Springer, New York (1994)
von der Malsurg, C.: Vision as an exercise in organic computing. GI Jahrestagung (2), 631–635 (2004)

Mange, D., Sanchez, E., Stauffer, A., Tempesti, G., Marchal, P., Piguet, C.: Embryonics: A new methodology for designing Field-Programmable Gate Arrays with Self-Repair and Self-Replicating Properties. IEEE Transactions on VLSI Systems 6(3), 387–399 (1998)

Mange, D., Stauffer, A., Petraglio, E., Tempesti, E.: Self-replicating loop with universal construction. Physica D 191, 178–192 (2004)

Mann, S. (ed.): Biomimetic Materials Chemistry. VCH, Weinheim (1996)

Matsumaru, N., Dittrich, P.: Organization-oriented chemical programming for the organic design of distributed computing systems. In: Proc. of BIONETICS 2006, Cavalese, December 11-13. IEEE, Los Alamitos (2006)

Miller, J.F.: Evolution in materio. In: International Conference on Evolvable Systems, Prague, Czech Republic (2008)

Miller, J.F., Downing, K.: Evolution in materio: Looking beyond the silicon box. In: Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware, pp. 167–176. IEEE Computer Society Press, Los Alamitos (2002)

Müller-Schloer, C., von der Malsburg, C., Wurtz, R.P.: Organic Computing. Informatik Spektrum (27), 332–336 (2004)

Nikolajewa, S., Friedel, M., Beyer, A., Wilhelm, T.: The new classification scheme of the genetic code, tRNA usage and early code evolution. J. Bioinf. Comput. Biol. 4, 609–620 (2006)

Nolfi, S., Parisi, D.: Genotypes for Neural Networks. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (1995a)

Nolfi, S., Parisi, D.: Evolving artificial neural networks that develop in time. In: Moran, F., Moreno, A., Merelo, J.J., Chacon, P. (eds.) Advance in Artificial Life, Proceeding of the third European Conference on Artificial Life. Springer, Heidelberg (1995b)

Nolfi, S., Parisi, D.: Neural networks in an artificial life perspective. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.D. (eds.) Proceedings of the 7th International Conference on Artificial Neural Networks (ICANN 1997), pp. 733–738. Springer, Berlin (1997)

Nolfi, S., Tani, J.: Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. Connection Science 2(11), 129–152 (1999)

van Noort, D., Wagler, P., McCaskill, J.S.: The role of microreactors in molecular computing. Smart Mater. Struct. 11, 756–760 (2002)

Patel, A.: The triple genetic code had a doublet predecessor. J.Theor. Biol. 233, 527–532 (2005)

Paton, R.C., Toh, C.-H.: Computational aspects of protein functionality. Comp. Funct. Genom. 5, 85–90 (2004)

Pattee, H.H.: Evolving self-reference: matter, symbols, and semantic closure. Communication and Cognition-Artificial Intelligence 12(1-2), 9–28 (1995)

Pattee, H.H.: Causation, control and the evolution of complexity. In: Anderson, P.B., et al. (eds.) Downward Causation, pp. 63–77. Aarhus University Press, Aarhus (2000)

Perlwitz, M.D., Burks, C., Waterman, M.S.: Pattern Analysis of the Genetic Code. Advances in Applied Mathematics 9, 7–21 (1988)

Pietzowski, A., Satzger, B., Trumler, W., Ungerer, T.: A bioinspired approach for self-protecting in organic middleware with artificial antibodies. In: de Meer, H., Sterbenz, J.P.G. (eds.) IWSOS 2006. LNCS, vol. 4124, pp. 202–215. Springer, Heidelberg (2006)

Plate, T.: Holographic Reduced Representations. IEEE Transactions of Neural Networks 6, 623–641 (1995)
Pollack, J.B.: Recursive Distributed Representations. Artificial Intelligence 46, 77–105 (1990)
Prigogine, I.: From Being into Becoming. W. H. Freeman, San Francisco (1980)
Quick, T., Dautenhahn, K., Nehaniv, C., Roberts, G.: The Essence of Embodiment: A Framework for Understanding and Exploiting Structural Coupling Between System and Environment. In: Proc. Third International Conference on Computing Anticipatory Systems, Liège, Belgium (CASYS 1999) (1999)
Rasmussen, S., Chen, L., Deamer, D., Krakauer, D.C., Packard, N.H., Stadler, P.F., Bedau, M.A.: Transition from nonliving to living matter. Science 303, 963–965 (2004)
Rocha, L.M.: Artificial semantically closed objects, Communication and Cognition. Artificial Intelligence 12(1-2), 63–90 (1995)
Rocha, L.M.: Evidence Sets and Contextual Genetic Algorithms: Exploring Uncertainty, Context and Embodiment in Cognitive and Biological Systems, Ph.D. Dissertation, Binghamton University (1997)
Ronneberg, T., Landweber, L.F., Freeland, S.J.: Testing a biosynthetic theory of the genetic code: fact or artifact? PNAS 97(25), 13690–13695 (2000)
Schmajuk, N.A., Thieme, A.D.: Purposive behavior and cognitive mapping: An adaptive neural network. Biological Cybernetics 67, 165–174 (1992)
Schrauwen, B., Verstraeten, D., van Campenhout, J.: An overview of reservoir computing theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks, pp. 471–482 (2007)
Shastri, L., Ajjanagadde, V.: From simple associations to systematic reasoning: A connectionist encoding of rules, variables and dynamic bindings using temporal synchrony. Behavioral and Brain Sciences 16(3), 417–493 (1993)
Siegmund, E., Heine, B., Schulmeyer, P.: Molecular electronics: The first steps towards a new technology. Int. J. of Electronics 69, 145–152 (1990)
Spiro, P.A., Parkinson, J.S., Othmer, H.G.: A model of excitation and adaptation in bacterial chemotaxis. Proc. Natl. Acad. Sci. USA 94, 7263–7268 (1997)
Suzuki, H., Sawai, H.: Chemical genetic algorithms-Coevolution between codes and code translation. In: Artificial Life VIII, pp. 164–172. MIT Press, Cambridge (2002)
Tangen, U., Wagler, P.F., Chemnitz, S., Goranovic, G., Maeke, T., McCaskill, J.S.: An electronically controlled microfluidics approach towards artificial cells. Complexus 3, 48–57 (2006)
Taylor, E.G., Hummel, J.E.: Finding similarity in a model of relational reasoning. Cognitive Systems Research 10, 229–239 (2009)
Taylor, T.: Creativity in evolution: Individuals, Interactions and Environments. In: Bentley, P.J., Corne, D.W. (eds.) Creative Evolutionary Systems. Morgan Kaufmann, San Fransisco (2002)
Thompson, A.: Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution. Springer, Heidelberg (1998)
Trifonov, E.N.: Elucidating sequence codes. Three codes for evolution. Annals of the NY. Acad. Sci. 870, 330–338 (1999)
Trifonov, E.N.: Consensus temporal order of amino acids and evolution of the triplet code. Gene 261, 139–151 (2000)
Trullier, O., Meyer, J.A.: Biomimetic Navigation Models and Strategies in Animats. AI Commun. 10(2), 79–92 (1997)

von Uexküll, J.: Theoretische Biologie. Frankfurt a. M.: Suhrkamp Taschenbuch Wissenschaft, xxiv+378 (1973)
Verpoorte, E., de Rooij, N.F.: Microfluidics meets MEMS. Proc. of the IEEE. 91, 930–953 (2003)
Virgo, N., Harvey, I.: Adaptive growth processes: a model inspired by Pask's ear. In: Bullock, S., Noble, J., Watson, R.A., Bedau, M.A. (eds.) Proceedings of the Eleventh International Conference on Artificial Life, pp. 656–661. MIT Press, Cambridge (2008)
Vsevolodov, N.: Biomolecular Electronics, an Introduction via Photosensitive Proteins. Birkhaeuser, Boston (1998)
Zachary, W., Le Mentec, J.C.: Incorporating metacognitive capabilities in synthetic cognition. In: Proceedings of The Ninth Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL, pp. 513–521 (2000)
Weberndorfer, G., Hofacker, I.L., Stadler, P.F.: On the evolution of primitive genetic codes. Orig. Life Evol. Biosph. 33(4-5), 491–514 (2003)
Wilhelm, T., Nikolajewa, S.L.: A new classification scheme of the genetic code. J. Mol. Evol. 59, 598–605 (2004)
Wilson, H.W., Halford, G.S.: Robustness of Tensor Product networks using distributed representations. In: Proceedings of the Fifth Australian Conference on Neural Networks (ACNN 1994), pp. 258–261 (1994)
Winfree, E.: Algorithmic self-assembly of DNA: Theoretical motivations and 2d assembly experiments. Journal of Biomolecular Structure & Dynamics 11, 263–270 (2000)
Wong, J.T.: A co-evolution theory of the genetic code. Proc. Natl. Acad. Sci. USA 72, 1909–1912 (1975)
Wu, H.L., Bagby, S., van der Elsen, J.M.: Evolution of the genetic triplet code via two types of doublet codons. J. Mol. Evol. 61, 54–64 (2005)
Yao, X.: Evolving Artificial Neural Networks. Proceeding of the IEEE 87(9), 1423–1447 (1999)
Zauner, K.P.: Molecular Information Technology. Critical Reviews in Solid State and Material Sciences 30(1), 33–69 (2005)