Modeling The Phylogenetic Dimension Of Evolutionary Systems: Toward a New Taxonomy Of Bio-inspired Systems

SEIF EDDINE MILI[1]
Laboratoire de Recherche en Informatique (LRI)
University Badji Mokhtar Annaba, Algeria
seifmili@gmail.com

DJAMEL MESLATI [2]
Laboratoire de Recherche en Informatique (LRI)
University Badji Mokhtar Annaba, Algeria
meslati_djamel@yahoo.com

Abstract--Computer systems involving and using a life based process are increasing in numbers and are more and more used, this shows that bio inspired approaches are multiplying and so a classification needs to be done. Finding a common number of discriminating criteria is not an easy task, in fact the number of bio inspired approaches is significantly increasing day by day as well as the used metaphors. The fact that there aren't any common concepts and definitions makes this task even harder, in this context our intention is to find a common form that allows us to compare different systems and so highlight the common criteria in order to achieve a classification.

In this paper we first give the reason justifying the importance of making a taxonomy, then we propose three views based on model driven architecture (MDA) and biology inspired, that allows us to find the characteristics of bio inspired approaches in either their structural aspects and behaviour aspects. As an illustration of our method we will consider the taxonomy of evolutionary approaches.

Keywords: Evolutionary systems, MDA, Design Patterns, *BioInspired System*, Phylogenies, Taxonomy

I. INTRODUCTION

We mean by a biological metaphor an analogy which we try to determine in between the biological world and the artificial one, in a way that enable us to propose approaches which imitate certain aspects of the first one and ignore others. As a rule, metaphors are not trying to reproduce whatever is biological, but are translated based on whatever is possible to do and create with. And so we can come to the conclusion that biological metaphors are evolving depending on our comprehension of reality and our aptitude to extract a practical and useful elements. The above shows that, metaphor diversity is leaking from the diversity of the biological world this will endeavour several possibilities of inspiration. It is necessary to define the characteristics of different bio inspired approaches in order to evaluate their impact concerning an aspect or another ie: dynamic adaptability, failure resistance [1].

As it has been said above the diversity of the biological organisms leads to a multitude of inspirations and so approaches which is necessary to classify, below are some reason justifying the importance of making a classification:

- Highlighting the characteristics and comparing approaches.
- Finding common and unifying concepts
- facilitating the study of bio inspired systems

Finding common characteristics is not an easy task; in fact this is due to the lack of common agreed on definitions and concepts. Two parts are involved in the bio inspired systems: process (behaviour aspects) and architectural structure (structural aspects). As a result to this, putting characteristics for systems means finding characteristics for both parts. We will use the Model Driven Architecture (MDA) strength to achieve this.

Our common form enables us to classify the bio inspired systems either from a structural aspect and behaviour aspect.

In this paper we will apply our work in order to establish most needed evolutionary approaches, we will first take into consideration evolutionary algorithm and model engineering followed by concepts of common forms ”proposed approach”. We will highlight the characteristics of evolutionary approaches in goal to make a classification and end up with related work and conclusion.

II. BACKGROUND

In the background section, the main concepts, terminology, and ideas presented in the paper are defined. The section first explores some concepts relating to the field of biology than looks at evolutionary system, and finally MDA.

A. Biology: ontogeny, phylogeny, epigeny

Living multi-cellular organisms are not created in the completely achieved form we usually know. The organism begins life as a single cell, endowed with a developmental program coded in its genome. In other words ontogeny investigates the “developmental model” of an individual organism from the earliest embryonic stage to maturity.

Phylogeny is the study of phylogenesis within given species; reproduction consists in transmitting the genome of one or two parents to offspring. The genome of the descendant first cell is obtained from that/those of the parents, through mutations and crossing over[1].

Epigenesis, relating to epigenetics, uses specific structures to store and handle a huge number of interactions with the environment. The epigenetic process is supported by three systems: the nervous system, the endocrine system and the immune system [8].

The section has explored some of the biological processes involved in shaping and guiding the evolution and development of biological individuals.

B. Evolutionary approach

Since the publication of the origin of species in 1859, we understood that evolution process based on natural selection and recombination genetic code is the centre and the main way of life adaptation systems to their environment. Even if the link between evolution and optimisation is excessive [2], the dynamic evolution is not less than an extraordinary force by its aptitude to discover original solutions to problems that are constantly faced and renewed in the same way as programmer were inspired by living creatures to conceive a formal neural network, they were also inspired by the Darwin theory to conceive evolutionary algorithm.

Biologist have proposed landscape adaptive concept [3] to represent adaptation level of a population in between different species of possible genotype. Evolutionary algorithm behaves in a similar way; we examine the level of adaptation of different genetic configuration to get closer to the highest points. We then take into consideration the individual fitness (the resulted value of function evolution and representing the individual adaptation to its environment)

The idea is to build potential solution group randomly and by Darwin selection, evolving those solutions to achieve optimal results.

Most of the evolve systems are based on four major pillars of natural evolution: maintaining a population, diversity creation, and selection mechanism and inheritance process.

C. Model Driven Architecture

Key to MDA is the importance of models in the software development process. Within MDA the software development process is driven by the activity of modelling the business software system. The MDA development process does not look very different from a traditional lifecycle, containing the same phases (requirements, analysis, low level design, coding, testing, and deployment)[4,5]. One of the major differences to traditional development processes lies in the nature of the artefacts that are created during the development process[6]. These artefacts are formal models, i.e. models that can be understood by computers and finally be transformed into a representation that lends itself to execution [7].

In summary MDA is a framework defined that separates the platform specific concerns from platform independent concerns, which is represented by different views of a system.

III. The Proposed Approach

Our method is inspired from both: POE model (for more details see [8,9]) and MDA approach, the Architectural Units (AU) are the central points of the method. Our purpose is to define several Architectural Units to describe the ontogenetic, epigenetic and phylogenetic views.

The AU consists of a number n of input models and a transformation that produces the k output models. Transformations can have attributes and operators that are applied to produce the output models (see figure 1). Models as well as transformations can be of various types. The environment supplies diverse stimuli such as events that help in triggering or stopping the transformation [1].

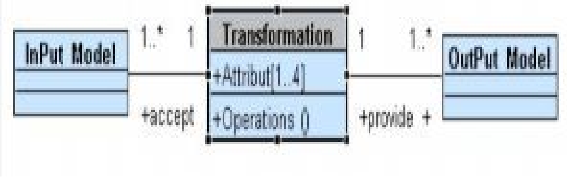

Figure 1. This figure show the description of the architectural unit

We can formulate the Architectural Unit like a function as show below:

Name_of_AU(IM1,IM2,...,IMn)→ OM1,OM2,...,OMk.

This form of description will be used below to describe the phelogenetic view of evolutionary systems.

There are three parts involved in all bio-inspired systems: the processes, the structures and the environment where the system is designed to operate [1]. Therefore, characterizing a system comes to characterize each part. The structure consists of all the models available in a system. We show in table 1, the derived criteria set.

Role: A model can play two possible roles for each transformation where it is involved. The individual role or the species role. That is, a model can be involved simultaneously as a species in a process and as an individual in another.
Description type: A model can be a genome, a phenotype or any other description. Genome models are often coded using low level symbols such as a sequence of bits, while the phenotype is more abstract. Models can be implemented in hardware or stored in some memory. All models are interpretable.
Element/Set: The model can be a single element or a set of elements.
Granularity: Characterizes the item available to transformations. Models range from fine grained to coarse grained. When we use phylogenesis to adjust a neural network, the grain is the weight attached to each connection. In other cases, the grain can be a symbol, a rule, an instruction or a function in a program. The finest grain is the bit.
Alterability: Defines how easy the model is alterable. Models can be highly alterable when they are stored in a soft memory. They are less alterable or reconfigurable when implemented in hardware. Furthermore alterability can be manual or fully/partially automated.
Composition: A model can be simple or composed. A composed model can be decomposed into sub-models and transformations.

Table 1 The model criteria set

A. The Ontogenetic View

The Ontogenetic view is constructed using one AU: the development AU. Formally, the development unit can be written using the functional notation: Develop(D, M) → M' Which means that M' is obtained from M by a modification according to some description in D. M, D and M' are models.

B. The Phelogenetic View

The phylogenetic view process is constructed using two types of AU: the Reproduction AU and the Selection AU. The reproduction AU allows combination of input models using genetic operators (i.e. crossover and mutation) to produce output models. The transformation attributes include the mutation rates, the crossover type. Formally, the reproduction is written: Reproduce (RM,S) → S' Where RM is a model containing the description of the reproduction, S and S' are sets of models. Each element in S' is obtained (according to RM) from one or more elements of S using mutation and crossover operators. The abstraction levels of S and S' are the same.

The selection unit allows the selection of one or more models for the set of input models (i.e. output models are a subset of the input models). Models themselves are not altered. The transformation operators include the fitness functions and attributes, the selection threshold. Formally, the selection is written: Select(SM, S) → S' Where SM is a model containing the description of the selection, S' is a subset of S containing elements selected according to SM. The abstraction levels of S and S' are the same.

C. The Epegenetic View

The epigenetic view is constructed using two AUs: the interpretation AU and the adjustment AU. The interpretation AU accepts executable models and data models as inputs and produces a data model as output. The adjustment unit adjusts one model according to another input model. The interpretation can be written: Interpret (P, I) → O.

O is obtained by transforming the I model according to some description in P. The abstraction levels of I and O are the same. However, compared to P, they may have greater or lesser abstraction level. The adjustment can be written: Adjust (M, P) →P'.

D. Characterization of biological process

In this section, we characterized the biological processes using the functional expressions:

Ontogenesis

Iterate (C, Assign(Ph, Develop(G,Ph)))

Phylogenesis

Iterate (C, Assign(S, Select(FM, Reproduce(RM,S))))

Epigenesis

(Assign(M, Null),
Iterate (C,(Iterate(SC, Assign(M, Develop(D, M))),
Assign(D, Adjust(Interpret(M,IDM), D)))))

From the previous, we remark that the three processes are similar since they all aim to deal with evolution, but in the same time there is some differences such as :

- The degree of alterability of the used models
- The abstraction levels of the used models
- The process cycle frequency

• The intervention of the environment on the processes
In figure 2 we summarize our vision of the relationship between the three biological processes.

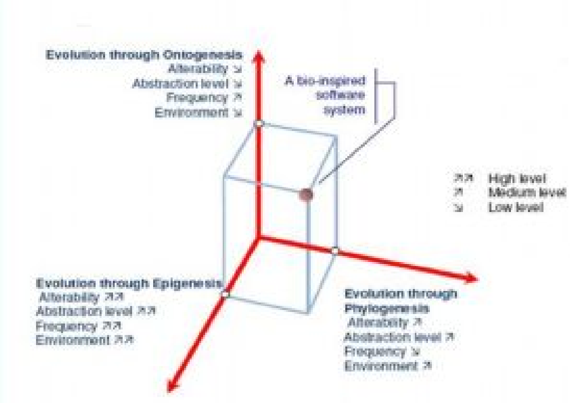

Figure 2. Relationship between the biological processes.

IV. Description of Evolutionary Approaches

Regarding to [10] there are different types of evolutionary algorithm, witch are different to each other by genetic representation and operators, in this paper we will discuss two types:

A. Genetic Algorithm

Operate on binary representations of the individuals and emphasize the role of building blocks and crossover [11]. Genetic programming operates on tree-based representations of computer programs and circuits. Evolutionary programming often relies on tournament-based selection with gradual population replacement and does not use crossover [12]. The behaviour aspect was given by the expression below:
Iterate(C, assign (S, Select (SM, Reproduce (OM , S))))
Where C : is the convergence condition as an example in optimisation the satisfaction of objective function
S: is the solution model that satisfies our problem
SM: is the selection model in this case we can have the proportionate selection model, generational replacement selection model, Truncated rank-based selection model or Tournament selection model
OM: is the reproduction model, and can be of two types, either by mutation or cross-over and both of them have their own mode, in this case it is more the cross over model that is used.
The expression function means that the system is reproduced according to the model OM this process with the selection one tune up the system with new population
Below is the structural aspect of genetic algorithm

Role: genetic algorithm model play role of Population.

Description type: the genome is describe in several manner as chromosome form in genetic algorithm, as tree in genetic programming..

Element/Set: a genetic algorithm system is a set of two classes' individual and population.

Granularity: the granularity of phenotype is a set of bit that represents words

Alterability: the phenotype is alterable and it's the main principal of genetic algorithm.

Composition: a genetic algorithm systems are a succession of two transformations the first is reproduction and second is selection.

We describe below the meta model of genetic algorithm using UML 2.0

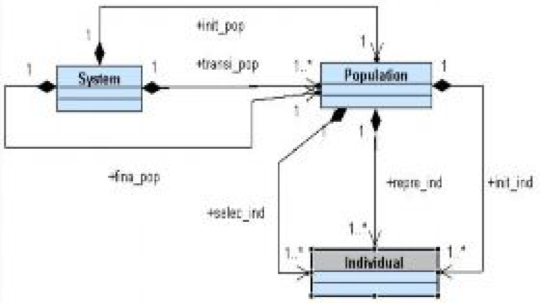

Figure 3. A view of meta model of genetic algorithm

System: class system define our system and it's the evolution of a population from initial state to final state. We need to use singleton pattern to represent system because they are one instance of object system.

Population: define the features of population like seize and individual, the population evolve according individual (selected one, reproduced one).

Individual: define the characteristic of individual like representation, seize of individual…

We present below the selection transformation.

The figure 4 shows the representation in UML 2.0 of the selected transformation, witch take population model like input and provide new population model like output. We use a strategy pattern to design selected transformation model.

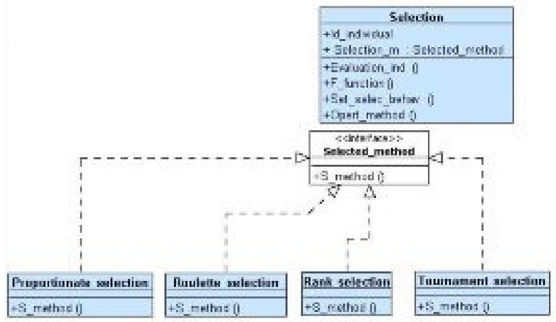

Figure 4. A view of meta model of selection transformation

A selection class has tow attribute: Id_individual that identify the individual and Selection_m. the last one is used in Set_selec_behav method to describe the selection behaviour. Operat_method is used to encapsulate the behaviour of selection and use S_method for each behaviour type (roulette, rank…).S_method define an algorithm for each selection behaviour.

We can use also strategy pattern to design reproduced transformation.

The figure 5 shows the representation in UML 2.0 of the reproduction transformation, witch take individual model like input and provide new individual model like output. We use a strategy pattern to design reproduction transformation model.

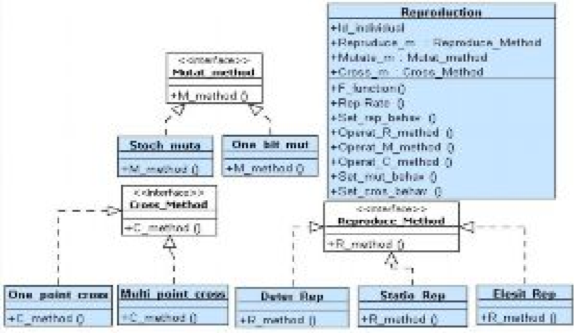

Figure 5. A view of meta model of reproduction transformation

A reproduction class has four attribute: Id_individual that identify the individual repruduce_m which is used in Set_rep_behav method to describe the reproduction behaviour, Mutate_m and Cross_m. are used respectively in Set_mut_behav and Set_cros_behav methods to describe the mutation and crossover behaviour. Operat_R_method, Operat_M_method, Operat_C_method are used respectively to encapsulate the behaviours of reproduction, mutation and crossover.

The population class evolves from initial state to final state crossing transitional state according the selection transformation and reproduced transformation, the iteration transformation make refinement.

B. Simulated Annealing

Is a function optimization procedure based on random perturbations of a candidate solution and a probabilistic decision to retain the mutated solution. Simulated annealing takes inspiration from the process of shaping hot metals into stable forms through a gradual cooling process whereby the material transits from a disordered, unstable, high-energy state to an ordered, stable, low-energy state. In simulated annealing, the material is a candidate solution (equivalent to the individual phenotype of an evolutionary algorithm) whose parameters are randomly initialized. The solution undergoes a mutation and, if its energy (equivalent to the inverse of the fitness) is lower than that at the previous stage, the mutated solution replaces the old one. The procedure stops when the annealing temperature approaches the zero value [10]. The behaviour aspect was given by the expression below:

Select (SM, Iterate(C, assign (S, Adjust (OM , S,))))

C: is the condition of convergence, it is the temperature

S: is the solution model that satisfies our problem

SM: is the selection model, in this case we choose a random initial solution.

OM: is the adjustment model supporting only the mutation using mathematical methods like metropolis criterion.

Below is the structural aspect of simulated annealing

Role: simulated annealing model play role of individual.

Description type: the genome can be described in several manners as chromosome form, as tree form...

Element/Set: a simulated annealing system is an element that's representing by class named individual.

Granularity: the granularity of phenotype is a set of bit that represents words

Alterability: the phenotype is alterable and it's the main principal of simulated annealing

Composition: a simulated annealing systems are a succession of two transformations the first is selection and second is adjustment.

Figure 6 show the meta model of simulated annealing using UML 2.0

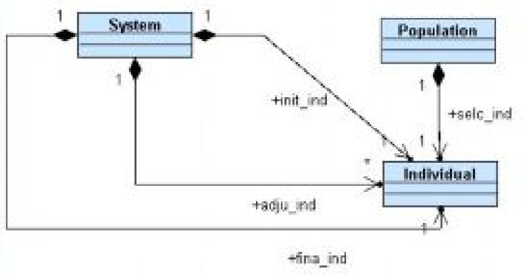

Figure 6. A view of meta model of simulated annealing

System: class system define our system and it's the evolution of a individual from initial state to final state. We need to use singleton pattern to represent system because they are one instance of object system.

Population: define the features of population like seize and individual, the population is used to select one individual for the evolution.

Individual: define the characteristic of individual like representation, seize of individual. The individual evolve according adjustment.

We present below the Adjust transformation

The figure 7 shows the representation in UML 2.0 of the adjustment transformation, witch take individual model like input and provide new individual model like output. We use a strategy pattern to design adjustment transformation model.

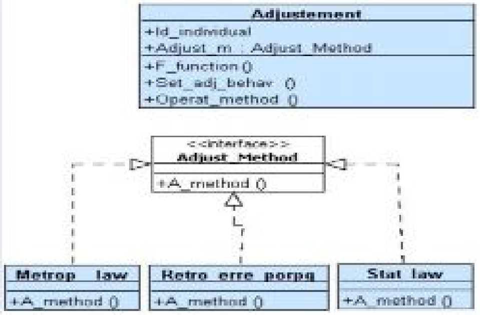

Figure 7. A view of meta model of adjustment transformation

An Adjustment class has two attribute: Id_individual that identify the individual and Adjust_m. the last one is used in Set_adj_behav method to describe the adjustment behaviour. Operat_method is used to encapsulate the behaviour of adjustment and use S_method for each behaviour type (metropolis, retropropagation of error…).S_method define an algorithm for each adjustment behaviour.

The individual class evolves from initial state to final state crossing transitional state according the selection transformation and adjustment transformation, the iteration transformation make refinement.

V. Related Work

Most of the taxonomies are based on disciplines or sub-disciplines. To the best of our knowledge, only one work is directly related to ours. The authors of this work used the Poetic model as the basis for the taxonomy of bio-inspired systems [8]. This Poetic taxonomy is it self bio-inspired and deals with a wide range of systems; however, some weaknesses can be raised:

•Some definitions used may be subject to discussions, such as considering that the environment has no effect during the ontogenesis, where it actually does.

• Processes can be combined, but the Poetic classification cannot discriminate the diverse combination forms. For example, within a combined phylogenetic-epigenetic approach, many combination forms may exist.

• The dichotomy individual/species is not considered as an important criterion. We think that the dichotomy is important and allows a better understanding of approaches.

Our work is based on the Poetic taxonomy and can be considered as a refinement that uses POE processes as the main discriminating criterion but adds a set of criteria to characterize a wide range of hybrid bio-inspired approaches.

VI. Conclusion and Discussion

The study that we have established had allowed us to make the following remarks:

Solution improvement by fitness function, which gives the quantification on an individual and is highly implicated in the selection criteria

The evolution is guided by a goal which is the objective that needs to be reached, all systems have the condition C which define the objective.

When we look at the composition of the process of studied systems, all systems are of a same value to a certain level of abstraction (the same process of transformation) the only difference is in the component of the model and the evolution mechanism.

We can extract tow kind of evolutionary system one centred individual and one centred population.

Evolutionary system based population is centred population were the population is the main of the system and the evolution is to evolve population class from initial state to final state.

Evolutionary system based individual is centred individual were the individual is the main of the system and the evolution is to evolve individual class from initial state to final state.

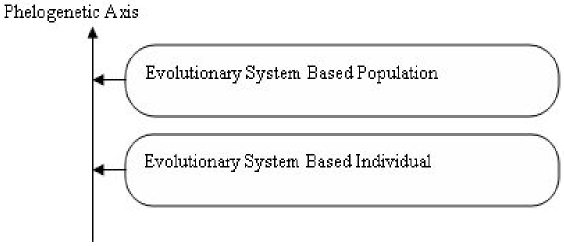
Phelogenetic Axis
Evolutionary System Based Population
Evolutionary System Based Individual

Figure 8 Phelogenetic Axis supporting two types of evolutionary system

A classification of bio-inspired systems would be of great value for bio-inspired system designers. Unfortunately little has been done in this direction.

In this paper we have proposed a common form which allows to make the criteria of bio inspired systems based on two aspects, structural and behaviour, using at the same time MDA and POE model. our common form has allowed us to classify the evolutionary system into two category which we have named centred population and centred individual, we have proved that there isn't a big difference between the systems that are classified in [9], and this difference is resumed in the transformation mechanism which is either the reproduction for some or the adjustment for the others.

We need to bear in mind that a good classification of bio inspired systems is very useful for the conception work of systems and approaches improvements. Our work isn't yet achieved, as a perspective we model the evolutionary approach presented, from the analyse steps to the generation code steps using tool that support MDA concepts” EMF (Eclipse Modeling Framework) and ADT (ATL Development Tools) “, we are trying to describe artificial neural network approach using model transformation with case study relative to XOR function.

References

[1] D. Meslati, L. Souici-Meslati and S.E Mili. Software Evolution and Natural Processes: A Taxonomy of Approaches. 3rd International Symposium on Innovation and Information and Communication Technology (ISIICT 2009), pages 48-57, Philadelphia University, Amman (Jordan), 15-17 December 2009.

[2] S. Gloud, The struture of Evolutionary Theory, Cambridge Harvard University Press 2002.

[3] Jean philippe Rennard, les reseaux Neuronaux chapitre 7 livre edition vuibert 2006 (in french).

[4] B. Tekinerdogan, M. Aksit and F. Henninger. Impact of Evolution of Concerns in the Model-Driven Architecture Design Approach. 2sd International Workshop on Aspect-Based and Model-Based Separation of Concerns in Software Systems, Bilbao (Spain), 10 July 2006. Electronic Notes in Theoretical Computer Science, Elsevier, Vol. 163, No. 2, pages 45-64, April 2007.

[5] X. Blanc. MDA en action: Ingénierie logicielle guidée par les modèles. Ed, Eyrolles, April 2005 (in french).

[6] L. Xiao and D. Greer. Adaptive Agent Model: Software Adaptivity using an Agent-oriented Model-Driven Architecture. Journal of Information and Software Technology, Elsevier, Vol. 51, No. 1, pages 109-137, January 2009.

[7] B. Bauer and J. Odell. UML 2.0 and agents: how to build agent-based systems with the new UML standard. Journal of Engineering Applications of Artificial Intelligence, Elsevier, Vol. 18, No. 2, Special issue on Agent-oriented Software Development, pages 141-157, March 2005.

[8] M. Sipper, E. Sanche, D. Mange, M. Tomassini, A. Pérez-Uribe and A. Stauffer. A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems. IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pages 83-97, April 1997.

[9] E. Sanchez, Phylogeny, Ontogeny, and Epigenesis. From Biology to Hardware, Volume 1259 de LCNS, pages 33-54 Springer, Berlin 1997.

[10] Dario Floreano,Claudio Mattiussi, Bio-Inspired Artificial Intelligence Theories, Methods, and Technologies, The MIT Press Cambridge, Massachusetts London, England, 2008

[11] A. E. Eiben, and J. E. Smith, Introduction to Evolutionary Computing. Springer-Verlag, Berlin , 2003

[12] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992