

Software Evolution and Natural Processes: A Taxonomy of Approaches

Djamel MESLATI, Labiba SOUICI-MESLATI, Seifeddine MILI
LRI Laboratory, Badji Mokhtar University, BP 12, 23000, Annaba, Algeria
meslati_djamel@yahoo.com, souici_labiba@yahoo.fr, seif03051982@yahoo.fr

Nowadays, we notice a proliferation of bio-inspired approaches that use various biological metaphors to deal with software complexity and maintenance. This proliferation is emphasized by the amount of improvement that has occurred in hardware technology as well as development tools and methods. Unfortunately, we notice a lack of a suitable framework that positions approaches, characterizes their concepts and favor their enhancement. In this article, we propose a set of criteria for characterizing bio-inspired software systems and show that software evolution can be dealt with using natural processes.

Epigenesis, Ontogenesis, phylogenesis, Software Evolution

1. INTRODUCTION

The software systems are now approaching levels of complexity such that their synthesis and maintenance raise many problems whose solutions exceed the human skills. Since biological systems have developed a set of well-tried mechanisms and desirable characteristics over millions of years, more and more engineers are beginning to look at nature to find inspiration for the design and maintenance of artificial systems. The recent proliferation of bio-inspired systems is due to:

- The confidence in the existence of biological metaphors that can appropriately resolve many problems [1].
- The success of certain approaches such as neural networks and genetic algorithms [6, 3].
- The amount of improvement in hardware technology as well as development tools and methods [11].

In front of this proliferation, it becomes necessary to provide a framework that positions approaches, characterizes their concepts and favor their enhancement. At the heart of this framework there is a taxonomy of approaches that allows:

- Characterizing and relating approaches
- Searching for common and unifying concepts
- Facilitating the study of bio-inspired systems
- Finding new promising inspiration directions
- Unifying bio-inspired systems terminology
- Elicitation of system requirements ...

Finding a suitable taxonomy of bio-inspired software systems is not an easy task since we need to partially achieve some of the previous goals before deriving taxonomy. It is not a dilemma but rather an incremental process where we derive taxonomy to partially achieve some goals and then refine the taxonomy. This process will lead to the enhancement of the bio-inspired approaches and ultimately promote a multidisciplinary synergy that best suits their nature.

Traditionally, bio-inspired approaches taxonomy is based on disciplines and sub-disciplines such as artificial intelligence, distributed artificial intelligence, artificial life, evolutionary computation and

cybernetics. Within each discipline/sub-discipline, approaches are classified again, using criteria reflecting the system goals, operating modes and various other characteristics.

For example, multiagents systems are classified within the distributed artificial intelligence and further classified as cognitive or reactive systems [2]. These classifications reflect the evolution of our inspiration from biological organisms rather than the intrinsic properties of bio-inspired systems. Some of its drawbacks are:

- The lack of preciseness which precludes the discrimination of approaches that are actually different. For example a multiagents system where agent's structure evolves using a phylogenetic process is different from another where coordination between agents emerges from a phylogenetic process [1]. This drawback will increase with the use of multiple mechanisms or metaphors and their hybridizations.
- The lack of naturalness. While some biological mechanisms are being intensively used, it seems difficult to maintain a correspondence between the designed systems and their counterparts in the nature. Even if this has no effect on the system effectiveness, it can be a helpful quality in its comprehension. For example, when using an evolutionary process within a robot, it is not obvious to identify what is the individual and what is the species. The robot is what corresponds, at first glance, to an individual, but, within one individual, phylogeny is meaningless.
- The use of disciplinary boundaries doesn't reflect current tendencies and constitutes an impediment to the synergy of approaches. Today, most approaches are hybrid, that is, they cut across these boundaries.

In this article, we propose a set of criteria for characterizing bio-inspired software systems and show that software evolution can be dealt with using natural processes. In section 2, we describe the natural processes that deal with evolution of organisms. In section 3, we derive a set of criteria used in the taxonomy of software systems. In section 4, we highlight the fact that the aim of natural processes is to deal with the various facets of evolution. In section 5, we describe some current approaches and characterize each using the set of criteria proposed. In section 6, we discuss related work, and in 7, we give a conclusion and some perspectives.

2. MAIN PROCESSES OF BIOLOGICAL AND BIO-INSPIRED SYSTEMS

It is now commonly admitted that living organisms are shaped by three main processes or organization levels: ontogenesis, epigenesis and phylogenesis. These processes, also known as the three axis of the Poetic model, have been identified in the framework of the Reconfigurable POETic Tissue project [12], conducted under the aegis of European Program of Information Society Technologies. The project involved one institute and four European universities [11, 12]. Processes of the Poetic model (*POE processes* for short) are derived from there corresponding counterparts in biology. We describe them in what follows.

Ontogenesis: Living multi-cellular organisms are not born in the fully developed form we know. The organism begins life as a single cell, endowed with a developmental program coded in its genome. The latter is continuously processed by the cell, which leads to its repeated division in a multitude of identical cells that have the same genome. Then, a form of communication appears between cells, allowing each one to execute the part of the genome corresponding to its position in the whole. This cellular differentiation ultimately leads to the formation of organs and gives an individual the morphology specific to its species. The whole developmental process that shapes the organism during its life is called ontogenesis. Ontogenesis is a deterministic process whose execution is influenced by the environment where it operates [5].

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. Being different from those of the parents, the genome controls the ontogenesis and produces an organism that is different from his parents. Due to the change in the genome, the descendant acquires new properties on which its survival may depend. The mutation and crossing over produce progressive change and evolution of species from one generation to another. This evolution is called phylogenesis [10]. Phylogenesis is a non deterministic process which has no effect on the organism itself, but does have on species. Phylogenesis introduces diversity within living organisms, and this diversity is important for their survival and their continuous adaptation as well as for the appearance of new species. The phylogenetic process is

based on the natural selection which allows the survival of individuals that are adapted to their environment. Therefore, in the phylogenetic process, the environment may have a major impact on the evolution of species.

Epigenesis. Since the genome is limited in the amount of information that can be stored, and since alteration of the genome by the environment, through the ontogenetic and phylogenetic processes, are slow and limited, complex organisms are shaped by a third process, called epigenesis. The latter 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. The structures used in these systems are easily alterable by the environment and allows the complex living organisms to learn and achieve symbolic processing of information [1, 11].

When applying, the previous processes, the structural aspect of an organism can be seen as composed of a genome and a phenotype. The latter includes all properties derived from the genome using any of the POE processes. The phenotype has an innate part derived by ontogenesis and acquired part derived by epigenesis.

The POE processes have been a source of inspiration for many systems. However, phylogenesis and epigenesis metaphors have been used frequently and successfully across many domains. Roughly speaking, genetic algorithms try to imitate the biological phylogenesis in solving optimization problems. Candidate solutions for a given problem are considered individuals and encoded as a genome having the form of abstract symbol strings. Individuals in the space of candidate solutions, are then evolved by means of crossover and mutation operations, and selected from a generation to the next one using a fitness function (i.e. quality criterion). This iterative procedure continues until either no essential improvement is done for a number of steps, or until a given number of iterations are performed. Despite the simplicity and brute force aspect of genetic algorithms, they have been applied successfully through a wide range of applications [1, 3].

Concerning epigenesis, metaphors of the three systems (nervous, immune and endocrine) are now been used, however the nervous system was the first system to be explored. It has received the most attention, giving rise to the field of artificial neural networks (ANN). The ANN metaphor is an attempt to mimic the characteristics of biological neurons [1, 6]. ANNs can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edges (with weights) are connections between neurons (i.e. synapses).

The ability to learn from examples is a fundamental trait of ANNs. A learning process in the ANN context can be viewed as the problem of adjusting network architecture and connection weights [6], so that a network can efficiently perform a specific task. ANNs are best suited for problems where there is a little or incomplete understanding but abundant data is available. This is typically the case of pattern recognition problems. The ANNs successes are comparable with those of genetic algorithms, without having a similarly wide range of applications.

Concerning ontogenesis, the first inspiration attempt was that of Von Neumann with his self replicating machine. The latter is an automaton capable of universal computation (i.e. equivalent to a Turing machine) and of universal construction (i.e. capable of constructing any automaton described by an artificial genome) [11]. Current metaphors try to mimic other ontogenetic mechanisms such as cellular division and cellular differentiation [5]. Current trends in bio-inspired systems tend to propose new biological metaphors and their combination (i.e. hybridization) to construct systems exhibiting living organisms' desirable properties such as emergent behaviors, adaptability to the environment and self healing.

3. DERIVING CRITERIA FOR A TAXONOMY

Finding a set of discriminating criteria is not an easy task, since the range of bio-inspired software systems is continuously increasing, and various metaphors are used. What makes the task even more difficult is the lack of consensual definitions of the concepts used.

Before deriving the criteria set, we have undertaken an analysis and comparison study of existing systems and approaches. What comes out from this study is the noticing of the presence (in one form or another) of an implementation of the POE processes and their combination. Therefore, we

decomposed the POE processes using one type of a simple construct called Architectural Unit (AU). Our goal was to:

- Describe the different systems using the AUs.
- Relating the three POE processes.

In a second step, we have tried to determine the criteria to discriminate the different bio-inspired systems using characteristics of both architectural units and their combination (i.e. the architectural style). Figure 1 shows the general form of the architectural unit.

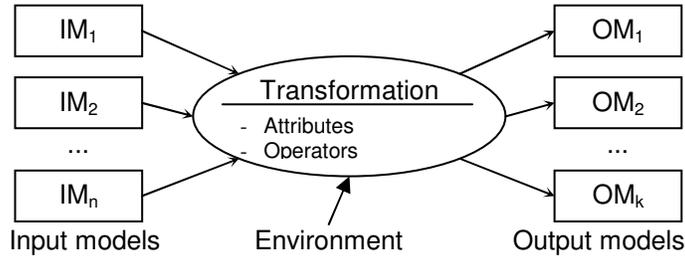


FIGURE 1: The architectural unit

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. 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.

The development AU is a specific AU that can be used in the ontogenetic process (figure 2). D is a descriptive model which guides the transformation (i.e. the genome). M is the model to transform (i.e. the innate part of a phenotype). During the ontogenesis, the output model in an iteration is the input model for the next one (i.e. M and Modified M are two consecutive states of the same model). Notice that, in the beginning, the innate part may be inexistent (this is indicated by dashed lines). Both D and M (in its current state) are used to decide which basic operation is achieved each iteration.

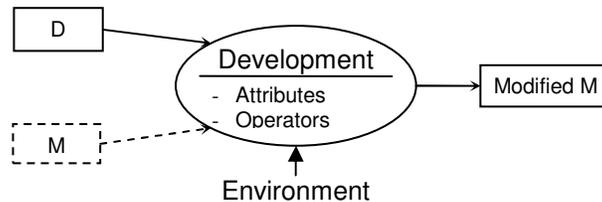


FIGURE 2: The development au

The transformation operators can be the cellular division, differentiation, death, migration, etc. The environment stimuli used in the development AU can be of various types, such as a temperature exceeding a certain limit or a wound in some part of the system. In the latter case, the transformation achieves some healing actions according to the part of the genome program (model D) dedicated to this goal. From an abstraction point of view, the development AU produces an output model M that have a higher abstraction level than the input model D . Formally, the development unit can be written using the functional notation:

$$\text{Develop}(D, M) \rightarrow M'$$

Which means that M' is obtained from M by a modification according to some description in D . M , D and M' are models. If we note the abstraction level of M by $\text{Abs}(M)$, then $\text{Abs}(M) > \text{Abs}(D)$ and $\text{Abs}(M') > \text{Abs}(D)$. M and M' have the same abstraction.

The phylogenetic 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:

$$\text{Reproduce}(RM, S) \rightarrow S'$$

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:

$$\text{Select}(SM, S) \rightarrow S'$$

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.

The epigenetic process is constructed using three AUs: the development AU, 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:

$$\text{Interpret}(P, I) \rightarrow O$$

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: $\text{Adjust}(M, P) \rightarrow P'$

P' is P modified according to some description given in M. P and P' have the same abstraction levels.

Roughly speaking, the interpretation and adjustment AU have the same expression as the development AU, however they are different when considering the abstraction of the input and output models. Whereas the development leads to higher abstraction, the adjustment maintains the same abstraction level between the output and the input models, and the interpretation can produce models that are more or less abstract.

The previous AU as well as the POE processes they compose can be combined in various ways according to the system goals. For this purpose we need other architectural units called Constructor Units (CU). Next are some of those CUs:

- Assign (M, M'): assigns the value of M' to M.
- Iterate (C, AU): executes repeatedly AU until the condition C is met.
- Block (AU1 Op AU2 Op ... Op AUn): executes AU1, AU2, ..., AUn. Where AU_i is a simple or a composed architectural unit. Op can be a comma or ||, to indicate a sequential or a concurrent execution.
- GetElem (S): Removes and returns one element from the set S, and AddElem (I, S): adds the element I to the set S.

Using the previous constructs, we can characterize the POE processes by functional expressions as follows:

Ontogenesis:

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

Which means: repeat the operation Develop on P, using G, until some condition C is met. P and G are two models representing the genome and the phenotype. Initially P doesn't exist.

Phylogenesis:

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

C is a condition, S is a set of models (for example individual genomes), FM a model describing de fitness function, and RM a model describing the characteristics of the reproduction.

Epigenesis: To deal with the learning process we use the following architecture:

```

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

```

A model M is constructed from its description D, then interpreted and the interpretation result is used to adjust its description. The process repeats until some condition C is met. Null is a null value and SC a condition used to stop the construction of the model M in each iteration of the external block. There are three parts involved in all bio-inspired systems: the processes, the structures and the environment where the system is designed to operate. 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.

<p>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.</p>
<p>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. In bio-inspired systems, various description types are used: list of symbols, L-systems [7], programs, neuron networks, rules, data modules, ... [1]. Models can be implemented in hardware or stored in some memory. All models are interpretable, the interpretation AU is able to propagate activations from input neurons to output neurons, infer rules in a rule set, executing programs, ... The other AUs interpret models in various ways: extracting items, comparing, adding, ...</p>
<p>Element/Set: The model can be a single element or a set of elements.</p>
<p>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.</p>
<p>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</p>
<p>Composition: A model can be simple or composed. A composed model can be decomposed into sub-models and transformations.</p>

TABLE 1: The model criteria set

The processes can be characterized by the functional expressions composed of models and AUs, which can be related in terms of presence/absence of some AUs, their imbrications (i.e. depth in the expression tree), and the sequencing within a block. An overall quality attribute concerning the tangling (degree of coupling) of the main system processes can be determined. Such an attribute is written: processes P1 and P2 are tightly/loosely coupled. The functional expressions are sufficiently precise and formal to support an automated analysis.

In some cases, an AU can be used recursively as a model in a higher level transformation. This allows, for example to determine the items of the AU using a higher order phylogenetic process. Another important criterion is the automatic, semi-automatic or manual support of the processes. In some systems, an ontogenetic process exists, but is carried out by a human operator.

The environment, where a bio-inspired system operates, can be characterized using two criteria: stimuli and system adaptability. If the environment changes and the system deals with this change, then we consider that the system is adaptable. A high degree of adaptability is a desirable characteristic and consequently, the adaptability is a highly discriminative criterion. Stimuli are the environment events perceived by a system.

4. NATURAL PROCESSES AND EVOLUTION

In the previous 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 ))
            )
        )
    )
```

When we analyze these expressions, we can deduce some similarities. While the ontogenetic process develops a phenotype Ph using a low level model (i.e. the genome), the phylogenetic process evolves a set of individuals S using a fitness function FM and non-deterministic operations. If we consider abstractly S , without looking to the individuals, as a phenotype, we notice that the phylogenesis develops a phenotype using a particular genome FM . We can perceive this fact, when considering an ant colony, where each individual evolves by an ontogenetic process and the colony evolves by a phylogenetic process.

Now, we can also consider the colony as one individual having a respiratory system (achieved by winged ants), a defense system (fighter ants), a reproduction system (the colony queen), etc. In this case the previous phylogenetic process becomes an ontogenetic one.

At the opposite each biological organism can be seen as a collection of cells that regenerates continuously and consequently the ontogenesis become a particular form of a phylogenesis.

On another hand, epigenesis provides an individual M with various behavioral properties that are easily alterable. Thus we can see the epigenesis as a particular ontogenesis that develops properties that are easily alterable, the genome being the society knowledge, where the individual lives, and which adjusts the description D of M .

In the same way, the ontogenesis is an epigenesis that provides individual with stable properties that are hard to change. We also notice a form of phylogenesis in the learning processes where individuals (i.e. possible solutions to a given problem) are enhanced iteratively to get a suitable solution.

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 3 we summarize our vision of the relationship between the three biological processes.

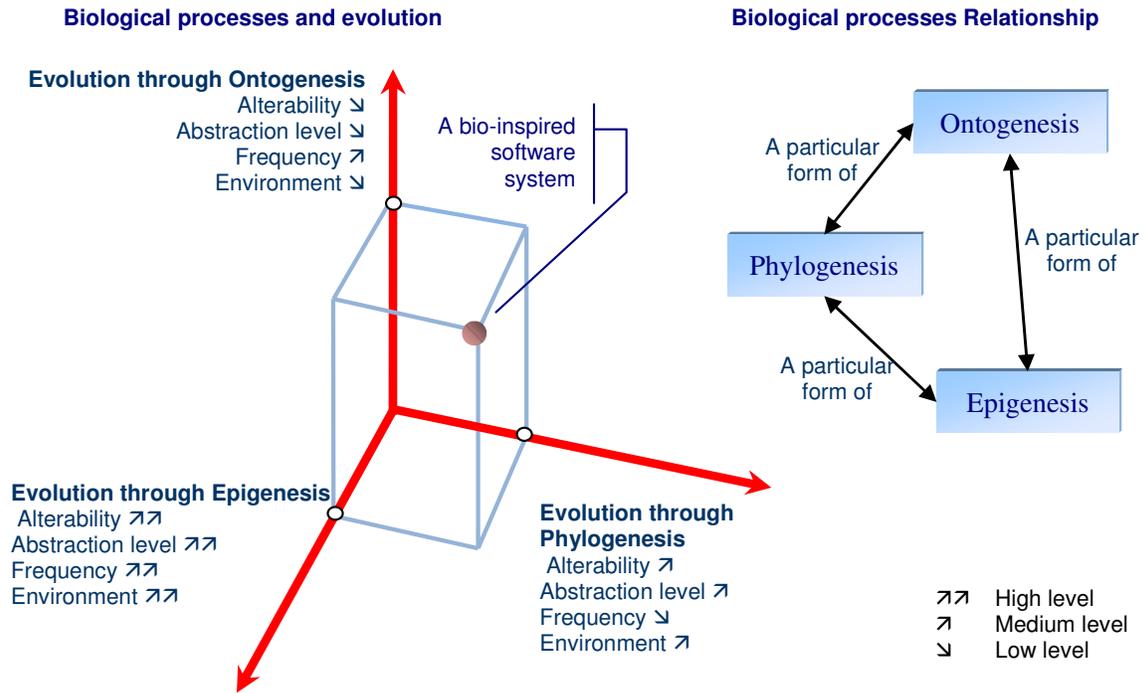


FIGURE 3: Evolution and biological processes relationship

5. CHARACTERIZING BIO-INSPIRED APPROACHES

We have characterized several classical and new bio-inspired approaches using the criteria set proposed. In the following, we present three promising approaches.

Mage is the first fully bio-inspired implementation of the ontogenetic process in software systems [9]. The ontogenetic process is supported by a dedicated execution system. The functional expression is $\text{Iterate}(C, \text{Assign}(P, \text{Develop}(G, P)))$ where G is the genome and P the phenotype. G describes an individual using gene as granularity. Genes are rule-like elements grouped in chromosomes. The genome is highly alterable but in the current state of the work it is manually changed. P is described using universal components and its granularity is the component, its characteristics and the connectors. P can be highly altered by the ontogenetic process. The ontogenetic process is controlled by genes using external events and conditions on component and gene states.

An application of an epigenetic process for software adaptation has been proposed in Rainbow project, at Carnegie Mellon University [4]. Although authors use terms such as self-healing, it is indeed a learning approach. Whose functional expression is the same as that of epigenesis in section 3. The adjustment AU uses an architectural description of a running system (AD), constraints and a model of indicators (collected during its interpretation) to dynamically and automatically alter the system. AD and constraints are called $Style$ and are alterable manually. The granularity is the component and connectors, and the AD is described using a graph with annotations. The running system, which can be dynamically and automatically altered, has components and connectors as granularity.

In [15], the authors proposed an object adaptive model where classes, attributes, relationships and the behaviors are described as metamodels. Users of a system are able to change the metamodels at runtime in order to deal with new requirements and enhancements. The metamodels are stored in a database and the running system is considered as a metamodel processor that takes into account changes as soon as they become available. This particular approach can be characterized by the functional expression $\text{Iterate}(C, \text{Interpret}(MD, I))$. Where MD is the set of metamodels that are executed using a data I . The metamodel granularity is variable since the user can change any part (attribute, class, method, ...) within the metamodel.

6. 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 [11]. This Poetic taxonomy is itself 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.

Biologists use two types of taxonomy. The first is the phylogenetic taxonomy which relies upon ancestral relations among individuals in order to divide them up. The second is the phenetic taxonomy which relies upon observed behavior and characteristics in order to divide them up. While the two approaches are useful in biology, the phylogenetic taxonomy is useless in the context of bio-inspired systems, since it does not add any value to their study or enhancement. Finally, our approach is partially inspired by the model driven architecture concepts [8].

7. CONCLUSION

Deriving a framework to position and relating software systems would be of great value for their development and maintenance. Unfortunately little has been done in this direction. In this paper, first, we proposed an approach that characterizes software systems using POE processes and set of criteria on the three dimensions: structure, process, and environment. The originality of this approach lies in the fact that it can characterize a wide range of systems independently of their hybridization degree. And second, we proposed our own vision of the relationship between the three biological processes. This relationship was deduced through the comparison of the functional description of the POE processes

As a perspective, we will consider other criteria and compare our approach with other taxonomies used inside bio-inspired sub-disciplines, such as characterizing soft computing systems [13]. In the same way, we will consider the POE processes relationship in the context of a development methodology that deals with the multiple aspects of software evolution.

REFERENCES.

- [1] C. Adami, Introduction to Artificial Life, Springer-Verlag New York, Inc., 1998
- [2] J. Ferber, Les systèmes multi-agents, vers une intelligence collective, InterEditions, Paris, 1995.
- [3] S. Forrest, "Genetic algorithms", ACM Computing Surveys, Vol. 28, issue 1, March 1996, pp. 77-80.
- [4] D. Garlan and B. Schmerl, "Model-Based Adaptation for Self-Healing Systems", WOSS'02, Charleston, USA, Nov. 2002, pp 27-32.
- [5] S. F. Gilbert, Developmental biology, Seventh Edition, Sinauer associates Inc. Publishers, March 2003.
- [6] A. K. Jain, Artificial Neural Networks: A Tutorial, IEEE Computer, March 1996, pp 31-44.
- [7] A. Lindenmayer and P. Prusinkiewicz, Developmental Models of Multicellular Organisms: A Computer Graphics Perspective, Addison-Wesley, 1988.
- [8] S. J. Mellor & al, MDA Distilled: Principles of Model Driven Architecture, Addison Wesley, March 2004
- [9] D. M & al, "The MAGE Ontogenetic Model: Towards autonomously-developed software", 17th Int. Conf. on Software & Systems Engineering and their Applications, Paris, November 2004.
- [10] M. Ridley, Evolution, Second edition, Blackwell scientific publications Ltd, Oxford, 1996.

- [11] M. Sipper & al, "A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems", IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, April 1997, pp 83-97.
- [12] G. Tempesti et al., "A POEtic Architecture for Bio-Inspired Hardware", 8th Int. Conf. on the Simulation and Synthesis of Living Systems, Sydney, Australia, Dec. 2002, MIT Press, Cambridge, 2002, pp 111-115.
- [13] A. Tettamanzi and M. Tomassini, Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems, Springer, Berlin, 2001.
- [14] <http://www.poetictissue.org>.
- [15] J. w. Yoder et R. Johnson, Implementing business rules with adaptive object models : Business rules approach, Prentice Hall, 2002
- [16] D. Meslati, L. Souici, S. Ghouli, Classification of software and hardware bio-inspired systems, AICCSA-06, 8-10 Mars 2006, Dubai/Sharjah, pp 221