Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 1128–1133

Procedia Computer Science

www.elsevier.com/locate/procedia

2nd International Workshop on Adaptive Technology (WAT 2018)

Programming Phenomenology: Proof of Concept on Adaptivity

Francisco S. Marcondes[a], Ítalo S. Vega[b], Paulo Novais[a]

[a]Algoritmi Centre, University of Minho, Largo do Paço 4704-553, Braga, Portugal
[b]Pontifical Catholic University of São Paulo (PUCSP), Rua Marques de Paranagua nº 111, São Paulo, Brazil

Abstract

Phenomenology is the empirical study of mind and consciousness. Considering programming activity as a cognitive procedure, phenomenology may be applied to address some of its issues. A particular issue of interest is how one programming approach differs from others, *i.e.* the cognition performed when using an approach differs from the one performed using another? If they are similar the two approaches are like to be the same, otherwise essentially different. Since adaptive computing proposal is Turing-equivalent it may be discussed about the actual differences among those approaches. As an example, λ-calculus is also Turing-equivalent but since the cognition performed is different it justifies the several λ-based existing approaches. In order to accomplish such analysis, bergsonism will be used as phenomenological method to be applied in particular adaptive structure called adaptive-graph. As a result it will be argued cognition performed using adaptive-graph is different from the one performed when using non-adaptive one. Then it will generalized suggesting adaptive computing cognition shall be further explored.

Keywords: Bergsonism; Adaptive Technology; Programming Phenomenology.

1. Introduction

A known problem when considering programming approaches is the existence of a "huge number of methods and method variants, with little understood differences and artificially magnified"[8]. In other words, it is not easy to realize actual differences among several exiting approaches in order to reason about advantages, scope of use, *etc.* as occurs, as example, to the *adaptive-graph*[12] proposal. In this sense, the central question of this paper is "what changes when using the adaptive-graph?". Since phenomenology is the empirical study of mind and consciousness[22] the objective that follows is to perform a phenomenological evaluation over the adaptive-graph proposal in order to understand how does it direct one thoughts. Based on the provided understanding it may become possible to compare adaptive-graph to other approaches and reason about its proper use. Applying phenomenology to several programming approaches produces what may be called *programming phenomenology*.

* Corresponding author. E-mail: id7515@alunos.uminho.pt. Tel.: +351 912-236-927.

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.159

initial graph	$+[(v_1) \leftrightarrow (v_3)]$	$+[(v_2) \leftrightarrow (v_3)]$	$-[(v_1) \leftrightarrow (v_2)]$	$-[(v_1) \leftrightarrow (v_3)]$
$G_0 : \begin{array}{cc} & \begin{array}{cc} v_1 & v_2 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \end{array} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$	$G_1 : \begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$	$G_2 : \begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$	$G_3 : \begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$	$G_4 : \begin{array}{cc} & \begin{array}{cc} v_2 & v_3 \end{array} \\ \begin{array}{c} v_2 \\ v_3 \end{array} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$

Table 1. Considering G_0, $V(G) = \{v_1, v_2\}$ and $E(G) = \{(v_1, v_2)\}$. Supose there is a $\Delta(G)_\alpha = \{(\mathcal{A}(\psi_G(0)) = +[v_1 \leftrightarrow v_3]), (\mathcal{A}(\psi_G(1)) = -[v_1 \leftrightarrow v_2])\}$. Applying $\Delta(G)_\alpha$ over G_0 it will become G_1 after $\mathcal{A}(\psi_G(0))$ and then G_2 after $\mathcal{A}(\psi_G(1))$. In G_2, $V(G) = \{v_1, v_3\}$ and $E(G) = \{(v_1, v_3)\}$.

1.1. Theme and Scope

Adaptivity focus on possible variance levels in software functionality[18] the first level is of static program where there is no change-over at all. The second one is of parametric programs which behaves according to input data. The third level is of configurable programs whose behavior can be defined after compiling time. The topmost level is of adaptive programs[18] whose are capable to set and reset its own functionalities during run-time[19].

In this sense, adaptivity is a Turing-equivalent[17] rule-driven computing device[15] as is λ-calculus, Post Machine, *etc.* It is noticeable that Turing Machines direct thoughts in an essentially different manner than λ-calculus does during programming activity. Such differences gave rise to programming paradigms such as imperative (based on Turing Machine formalism) and functional (based on λ-calculus formalism)[20]. In other hand Post Machines did not gave rise to any programming paradigm suggesting that it directs thoughts by a similar fashion as Turing Machine or λ-calculus. Programming paradigms suggests that in addition to formal or observable dimension of programming, there is also a cognitive meaning that must be addressed[7]. So, such analysis shall be performed to adaptivity in order to verify if it leads to a new programming paradigm or to compose another existing one.

Such is not an easy analysis since there are not many known credited instruments to performs it[8]. As been stated, one possibility to be explored is phenomenology but in computer science context it has been applied to human-computer interface and user-experience studies[5,10,13] but not as a research method. The only similar approach found was of Hiroyuki[7] but it intends to inspect the object's duration while this study intends to explore the duration of programmer's perception. By the way there is not in the scope of this paper to perform a wide range analysis nor provide a solid position over the theme; this paper aims to start a discussion within a proof of concept scope in order to be further developed. Proof of concept is the application of an idea in order to verify its feasibility. The chosen proof of concept object is the adaptive-graph proposal[12] since it is not a wide subject as a computing model yet it is a relevant computing tool to be of interest to address as it can be used as structure in many cases of data organization[21]. Roughly an adaptive-graph is graph whose incidence function may vary during run-time[12] and a graph is an ordered triple $G = (V, E, \psi)$ of vertices (V), edges (E) and incidence functions (ψ) connecting vertices and edges[2].

Adaptive-graph Definition. An adaptive-graph is a four-tuple expressed by $(V(G), E(G), \Delta(G), \mathcal{A}(\psi_G)$ where $V(G)$ is a non-empty set of vertices; $E(G)$ a non-empty set of edges which links a pair of vertices in the form $\{v_x \leftrightarrow v_y \mid v_x, v_y \in V(G)\}$ (usually depicted as an adjacency matrix); $\Delta(G)_k$ is an *ad hoc* triggered adaptation script composed by an ordered non-empty set of $\mathcal{A}(\psi_G)$, that can be 'k' adaptation scripts; $\mathcal{A}(\psi_G(j))$ is an adaptation function that can perform an edge inclusion $+[(v_x) \leftrightarrow (v_y)]$ or an edge removal $-[v_x \leftrightarrow v_y]$ (see table 1). The adaptive-graph must to hold its both symmetrical and connected properties in a $n \times n$ dimension. It means that if a new vertice is created, it must be both in matrix's line and column (holds dimension). If there is no incidence to a vertice, it shall be removed from matrix's line and column (holds dimension and connection). If there is an incidence from v_x to v_y it also exists an incidence from v_y to v_x (holds symmetry).

1.2. Problem Statement

As can be noticed from table 1 the adaptive-graph can be reduced to a set of subgraphs in a non-adaptive graph. In this sense it is difficult to reason about differences among those two proposals through mathematics. Graph as defined is a mathematical construction intended to be a static structure (it does not allow insertion or removal of nodes and edges). This means that its computational representation shall be classified as a static data structure or first level program according to software functionality variance scale[18]. Due to computer properties a computational graph must be first constituted, which means that memory address must be allocate to hold graph structure and information; this is done through an operation like "addEdge()"[21]. Since there is an "addEdge()" operation, broad interpreted operations as "removeEdge()" are natural. By such implementation an static structure (level one) becomes an adaptive one (level four); without an adaptive-graph like formalism there is no theoretical support for such implementation despite it appears to be widely used. In this sense, since an graph and adaptive-graph implementations may coincide, it becomes equally difficult to reason about those two proposals differences through computing.

Such difficulties can be expressed considering what Bergson[1] called "extension". Extension is the presence of a degree difference among existing things, as an example any object may be reduced to another one in geometric sense. In other hand when considering on object "final-cause" in Aristotle terms it becomes easier to reason among differences. Bergson goes further and considers the final-cause (or essence) as temporary, *i.e.*, an ever changing condition within a flux that can be discretized as durations. Duration is perception, the perception of a consciousness while it endures. This means that may be possible to qualitatively discuss the differences among graph and adaptive-graph through programmer's duration when using those tools. In other words it allows a qualitative analysis through phenomenology[14] since it aims to study the structures of consciousness as experienced from the first-person point of view[22]. Bergsonism is a possible choice to a phenomenological approach[9] once it is desired to address duration.

Summarizing, discussing the differences among graph and adaptive-graph through extension approaches leads to several dim and fuzzy issues been a problem in itself. In this sense a phenomenological approach, *i.e.* a qualitative personal experience description, which addresses duration as bergsonism may be helpful.

1.3. Bergson Philosophical Method

Bergsonism is the name used by Deleuze[3] to denote Bergson's efforts in philosophy. A issue that Bergson aimed to handle is to bring precision to philosophy and to do so he proposed what he is called as a philosophical method which intends to be in philosophy as precise as scientific method is to science[1]. It aims to provide a precise characterization of the study object (reaching its essence or defining its meta-model) allowing the derivation of deductions and predictions. Objectivity is achieved through inter-subjective, in this sense an bergsonist hypothesis by be objectively validated or rebutted by peers[11]. Yet it is not a method to be followed in an algorithm fashion, but it is based on some principles that must be in sight when performing it. In this sense, both scientific and philosophical methods intends to provide a rigorous and objective refutable/refineable description of the study object through a replicable method in order to derive deductions and predictions. It shall be stated that recently some authors have been considered Bergson's philosophical method as a phenomenological approach[9] as used in this paper.

The main difference among Bergson philosophy (or in this paper stated as "phenomenology") and scientific method is intention. Science mainly concerns with behavior or structure while philosophy (particularly metaphysics) mainly concerns with primary causes that "forces" something to be[3], *i.e.*, something essence. Scientific deductions and predictions are derived from behavior or structure (extension side) mainly through reason while philosophical deductions and predictions are derived from essence (duration side) mainly through intuition. Intuition has been a misunderstanding conception since several authors opposes it to reason[1] while what truly opposes to reason is abstraction[23]. Intuition is a skill level reached when someone becomes an specialist in something to the point that explicit reasoning can be dispensed[4] since the it may was "internalized". In this sense, philosophical objectivity may be achieved recognizing ones first-person experience[22] as described by other people[11]. Considering computing artifacts, the duration approach may be also used in order to address problems such as of in this paper that can be informally stated as "is there any real difference among graph and adaptive-graph?" which extension analysis did not provide proper answers.

1.4. Phenomenological Method

Deleuze organized bergsonism into a set of rules[3] and then Marcondes[11], abducting from Bergson's and Deleuze's texts, organized them into a framework to study programming called Bergsonist Investigation Framework (BIF). This framework is composed as showed in figure 1a. In addition to these elements, it must be performed a verification of adequacy over them. A first concern to be verified is the precision of a duration claim. A duration is said to be precise if it belongs to one and just one object; if a same duration belongs to one or more object, one can say that they are the same object differing by degree from each other. Afterwards, it must be verified if the individual experience claim

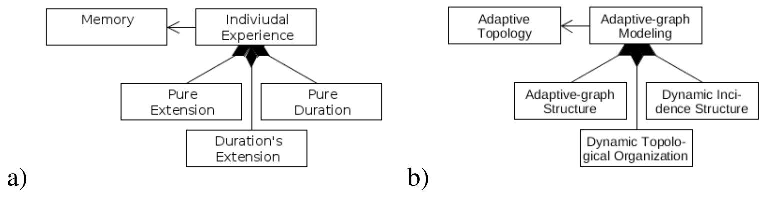

Fig. 1. a) BIF[11] representation in UML Class Diagram. *Individual experience* aims to specify what experience is intending to capture. *Memory* is the artifact resulted from the experience and the convergence point of all cognitive variations. *Pure extension* is the formal/symbolic meta-model used when expressing memory. *Duration's extension* is the symbolic mental construction that is constructed through the experience; it holds a model-diagram relation to memory and intends to accomodate reason from extension and intuition from duration. *Pure Duration* is a qualitative time that a concern bears in mind; during a duration, the mind is directed to a direction due to a concern and when a new concern interposes, it changes mind direction. b) Adaptive-graph representation on BIF.

is really an individual experience. It can be checked by verifying if the experience is a skill that can be developed in Dreyfus[4] sense. Then it must be presented the object characterization in terms of cognitive tendencies pure extension, duration's extension and pure duration, both individually and as multiplicity or nuances and verify if those tendencies are really compatible and do not present just an extension difference. Finally, it can be presented how memory is achieved due to cognition tendencies convergence and how it prolongs through several durations and encompasses them. This procedure results into a "hypothesis" that may offer some visions, possibilities and applications.

2. Adaptive Graph Hyphothesis

Claim. *An* adaptive-graph modeling *can be considered as a convergence of the* adaptive-graph structure*,* dynamic topological organization *and* dynamic incidence structure *that results into an* adaptive topology *(see figure 1b).*

Tendencies. *Adaptive-graph Structure* is e vertice-edge relation in the form $G = (V, E, \Delta, \mathcal{A}(\psi))$ which can be represented by any representation convention. This can be considered *pure extension* since: 1) there is a degree difference among one representation and another when using a same convention; 2) there is a degree difference among conventions in a way that one can be reduced to another; 3) there is a degree difference when a same convention is used to represented two different objects such graphs and adaptive-graphs. Applying such formalism there is a direction to a *dynamic topological organization* state of mind, *i.e.* several possibilities and situations of topology variation interposes to mind in order to be modeled. This can be considered *duration's extension* since: 1) it directs and is directed by the adaptive-graph structure; and 2) performs the actualization of the mental construction according to duration interposition according to each concern to be handled. Each concern that interposes to mind makes it to change its duration and to focus in one or more $\mathcal{A}(\psi_G)$ variation that can be able to handle it. This results into an update of mental construction that will reflect into the adaptive-graph structure. It is in this sense that *dynamic incidence structure* can be considered *pure duration*. The mental construction holds in *memory* and actualizes the topology of the adaptive-graph (*i.e.* the *adaptive topology*). This is memory since it prolongs through durations encompasses nuances and consolidates all presented tendencies. Those tendencies are proper since they can diverge to be analyzed and also converge reconstituting the *individual experience* of *adaptive-graph modeling* providing a natural bind and cannot be reduced to each other as would occur a degree difference were present.

Precision. A graph is a mathematical structure whose underlies several applications and technologies. Most of those graph-based applications and technologies put emphasis the behavior performed over the graph structure, as an example, an finite automaton may be described by a graph structure in a way that its primary concern is on state transition and not on its topological structure. All those similar cases shall not present the same duration as of graph since they use it as a realization mean. In other hand, some of those applications and technologies emphasis topology as a primary concern. One example is network topology whose structure can be considered and described as a graph structure. Theses similar cases present a degree difference from each other performing *nuances*. According to the example, a network topology may require more information than a plain graph but "removing" all this, its essential concern is directed to an incidence structure among machines (or nodes or vertices). Compared to a graph whose essence is

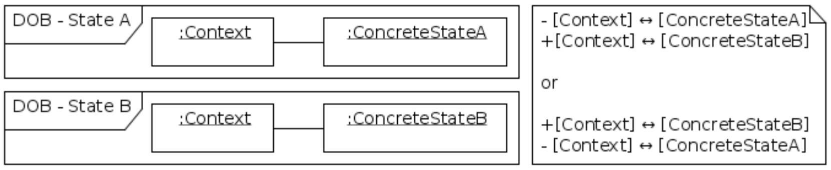
DOB - State A
:Context
:ConcreteStateA
DOB - State B
:Context
:ConcreteStateB
- [Context] ↔ [ConcreteStateA]
+[Context] ↔ [ConcreteStateB]
or
+[Context] ↔ [ConcreteStateB]
- [Context] ↔ [ConcreteStateA]

Fig. 2. Consider a UML Statechart organized through GoF's *State Pattern*[6] with two states, ConcreteStateA and ConcreteStateB, and a transition from the first to the second one. Using UML object diagram it is just possible to express two detached diagrams each to a particular object's configuration like in this example depicted as "DOB - State A" and "DOB - State B". When applying the adaptive-graph approach, it becomes possible to model how can be the structural transition from configuration "DOB - State A" to "DOB - State B" (depicted in the "note").

GRAPH	ADAPTIVE GRAPH	GRAPH	ADAPTIVE GRAPH
Queue	Multipriority queue	Neural Network	Multipattern Neural Network
Tree	Prunable Tree	Data Base	Multischema Data Base
Multi-linked	Separation of Concerns	Computer Network	Variable Topology Network

Table 2. Some possibilities considering well-known graph-based artifacts through adaptive-graph.

static incidence structure since during its modeling it does not arouse to mind interposing topologies; *adaptive-graph* essence is *dynamic incidence structure* since several topological variations possibilities interpose.

3. Results and Discussion

There are several adaptive-graph application possibilities already in use. An explicit use of adaptive-graph was to underlies an adaptive Virtual Network Embedding proposal to logically remove undesirable nodes of substrate network due to service specification[12] in order to reduce the processing time when computing virtual routes (a NP-hard problem)[12]. Actually any structure which presents topological variation is liked to be considered an adaptive-graph implementation since in strict theoretical terms, a graph topology is not allowed to change-over. In other words, adaptive-graph provides a theoretical foundation to structures whose topology changes over time. As an example, through the adaptive-graph conception it is possible to describe the transition among views of an UML Object Diagram as presented in figure 2. In this sense, it can be said that adaptive-graph formalism is applied at least in two situations: 1) solve problems whose data can be benefited by a dynamic topological organization; and 2) support structures that can be described as adaptive-graph based or like structures.

Taking up the initial question of this paper ("what changes when using the adaptive-graph?"), approaching some known graph-based artifacts considered through adaptive-graph yields to a reinterpretation or to a different perspective. Considering as example a Neural Network, the possibility to insert and remove neurons during run-time provide to the network the ability of restructures itself becoming a possibly Multipattern Neural Network. Considering a Tree structure through adaptive-graph conception it may provide the ability of pruning itself, this is useful considering some tree-based heuristics. Some of those possibilities are depicted in table 2.

Except for Variable Topology Network that was already performed[12] all those other are "mental experiments" based most on intuition based on those artifacts and presented adaptive-graph. An attempt that worth to be performed is apply adaptive-graph in a situation that there are no dynamic topological changes need. Doing so it is possible to notice that mind naturally conceives an adaptive-graph whose vertices and edges do not need to change, *i.e.*, a non-adaptive-graph. This means the several diagrammatic expressions that use the graph concept as underlying theory as UML diagrams[24], network structures[12], automata[16], digraph[12], *etc.* when considered through adaptive-graph conception may become an adaptive artifact.

Summarizing, through bergsonism it is possible to realize key differences among graph and adaptive-graph in a way that becomes possible to say the *adaptive-graph conception is essentially different from a non-adaptive graph* since it yields to diverse mind direction in problems that involve topological structures. Also, adaptive-graph suits to the class of structural problems with dynamic incidence structure (pure duration) that can be solved by a dynamic topological organization effort (duration's extension) that can be reduced to an adaptive-graph structure (pure extension). The main difference among graph and adaptive-graph is the mind focus during modeling which is quite subtle but with evident implications: on graph mind focuses on relations and on adaptive-graph on *changing* relations.

Since such phenomenological claim was state it can be validated, reviewed, refined or rebutted by further phenomenological studies over this same subject. A validation or rebuttal procedure may be apply the adaptive-graph in many situations and realize if it does or does not direct mind in a similar fashion that was presented in this paper. The proposed phenomenological method itself, since it was defined, may also be object of further critic and development.

4. Conclusion

This paper started with an informal research question stated as "what changes when using the adaptive-graph?". The justification to such question is the known difficulty to proper understand differences among methods and method variations. The motivation is to discuss if adaptivity leads to a distinct (as does λ-calculus) or compose an existing (as does Post Machines) programing paradigm. Since this is an wide enterprise and there is no wide recognized approach procedure to be performed, this paper stepped-back and focuses the research method applied as proof of concept in order to verify its feasibility.

The proposed method, based on Bergson's phenomenology, intends to divert symbolic difficulties in the extension side and focuses on personal experience in the duration side. In other words, instead of discussing structural or behavioral difference among things the proposal is to perform a qualitative discussion during their experience of use. In order to reduce complexity, to perform the proof of concept it was chosen a particular case of adaptivity called adaptive-graph. The proposed method was applied over the adaptive graph showing up feasible. This means this same method may be applied to other programming artifacts resulting into a *programming phenomenology*.

This study shows that adaptive and non-adaptive graphs direct mind in different ways suggesting they are essentially distinct artifacts. As a supposition this can be generalized to adaptivity in general, which been true may result into a distinct programming paradigm. In this sense, this method must to be enlarged to enfold adaptivity as a whole.

Acknowledgements

This paper has been supported by COMPETE: POCI-01-0145-FEDER-0070 43 and FCT Fundação para a Ciência e Tecnologia - Project UID/CEC/ 00319/2013.

References

1. Bergson, H. (2013). *La pensée et le mouvant*. Presses Électroniques de France.
2. Bondy, J. A. Murty U. S. R. (1982). *Graph Theory with Applications*. North-Holland.
3. Deleuze, G. (1988). *Bergsonism*. Zone Books.
4. Dreyfus, H. L. (2002). *A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian non-representationalist Cognitive Science*. n/a.
5. Frauenberger, C. Good, J. Bright, J. K.(2010). Phenomenology, a framework for participatory design. *PDC '10 Proceedings*. Sydney, Australia.
6. Gamma, E. Helm, R. Johnson, R. Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.
7. Hiroyuki, M. (2003). From reflection to interaction. *CRPIT '03 Computers and Philosophy*, v. 37 . Australian Computer Society.
8. Jacobson, I. Meyer, B Soley, R. (2009). The SEMAT Initiative: A Call for Action. *Dr.Dobb's*: http://goo.gl/wkiX2K
9. Kelly, M. (2010). *Bergson and Phenomenology*. Springer.
10. Mantis, H. M. Laaksolahti, J. Hk, K. (2014). My Self and You. *ACM Transactions on Computer-Human Interaction*, v. 21 i. 4.
11. Marcondes, F. S. (2015). *Abordagem bergsonista para o estudo da programação*. Ad: Vega, I. S. Catholic University of São Paulo, Master T.
12. Marcondes, F. S. Molina, M. A. T. (2017). Uma proposta de Topologia Adaptativa. *Memórias do WTA 2017*. EPUSP, São Paulo.
13. Marcos, L. Flores, F. Martnez, J. J. (2010). Lecturing about the phenomenology of databases. *ITiCSE '10 Proceedings*. Ankara, Turkey.
14. Nakayama, Y. (1994). "Phenomenology" and qualitative research methods. *Sei Roka Kango Daigaku kiyÅ*. 20:22-34.
15. Neto, J. J. (2002). Adaptive Rule-Driven Devices - General Formulation and Case Study. *Lecture Notes in Computer Science*, v. 2494. Springer.
16. Neto, J. J. (2003). Autômatos em engenharia de Computação. *I Sem. de la Sociedad Chotana de Ciencias y la Red Mundial de Cient. Peruanos*.
17. Neto, J. J. (2007). Um Levantamento da Evolução da Adaptatividade e da Tecnologia Adaptativa. *Revista IEEE América Latina*, v. 5, n. 7.
18. Neto, J. J. (2009). Adaptatividade: Generalização Conceitual. 3^o *Workshop de Tecnologia Adaptativa*. São Paulo, EPUSP.
19. Neto, J. J. (2009). Um Glossário sobre Adaptatividade. 3^o *Workshop de Tecnologia Adaptativa*. São Paulo, EPUSP.
20. Ramos, M. V. Neto, J. J. Vega, I. S. (2009). *Linguagens Formais: Teoria, Modelagem e Implementacao*. Bookman.
21. Sedgewick, R. Wayne, K. (2011). *Algorithms*. Addison-Wesley Professional, 4th edition.
22. Smith, D. W. (2016). Phenomenology. In: *The Stanford Encyclopedia of Philosophy (Winter 2016 Edition)*, Edward N. Zalta (ed.).
23. Teixeira, L. (2001). *A Doutrina dos Modos de Percepção e o Conceito de Abstração*. UNESP.
24. Vega, Í. Camargo, C. E. P. e Marcondes, F. S. (2015). Elaboração de especificações adaptativas. *Memórias do IX WTA*. EPUSP, São Paulo