Cartesian Systemic Emergence: Tackling Underspecified Notions in Incomplete Domains

Marta Franova, Yves Kodratoff
LRI, UMR8623 du CNRS & INRIA Saclay
Bât. 660, Orsay, France
e-mail: mf@lri.fr, yvekod@gmail.com

***Abstract*—Pulsation is a model for a particular evolutive improvement that handles control and prevention in intelligent systems built following this 'pulsative' model in incomplete domains. Cartesian systemic emergence presented in this paper answers the question of strategic aspects of building such systems. The main problem to tackle here is working with informal specifications while aiming at their formalization in a way that is driven by the pulsation model.**

Keywords-Cartesian Systemic Emergence; pulsation; symbiosis of information; Symbiotic Recursive Pulsative Systems; practical completeness; human reasoning mechanisms; Computational Cognitive Science.

I. INTRODUCTION

In this paper, we present Cartesian systemic emergence (CSE) that describes how to handle what we call a pulsation ([6], redefined below in Section II.C), that is a particular evolutive improvement in incomplete domains. The desired improvements have to guarantee control and prevention in the built intelligent system. By lack of control is meant here an agreement to overlook some secondary effects of the improvements.

Our answer to the question of how to achieve this goal has several facets. Namely, tackling underspecified symbiosis of information, recursion, on-purpose solution invention instead of a search in a given search space and formulating fruitful experiments during this invention. All these facets are briefly described in order to allow us to present a self-containing coherent description of CSE. These facets are related to the topics of human reasoning mechanisms, cognitive and computation models, human cognitive functions and their relationship, and even to modelling human multi-perception mechanisms.

CSE is presented here on a non-trivial toy example. However, in the field of program synthesis from formal specifications via inductive theorem proving, an ongoing research takes place [5], productively using CSE's know-how.

The purpose of this paper is four-fold:

- Propose a general method - called Cartesian Systemic Emergence - implementing strategic aspects of pulsation.
- Illustrate this method (born from our work on automating program construction [5]) on a simpler example that does not concern program construction, though it deals with a problem that many researchers may be facing.
- Show that the main problem to handle in CSE is working with informal specifications for the purpose of their formalized delimitation in coherence with the pulsation model.
- Mention the main problems and challenges addressed by CSE to various fields of Cognitive Science.

The paper is organized as follows. In Section II, we present fundamental notions necessary for understanding CSE, such as symbiosis and pulsation. We present also a difference between deductive and formal systems. Section III presents CSE. Namely, in Section III.A, we first present the rigorous mechanism that inspired CSE. Then, in Sections III.B and III.C, we present CSE on a non-trivial toy example. In Section IV we say a few words about the relevance of CSE to (Computational) Cognitive Science.

II. FUNDAMENTAL NOTIONS

In this section we are going to present the notions that will be used in the description of CSE.

A. *Symbiosis*

By *symbiosis* we understand a composition of parts that is separation-sensitive. This means that a separation of one or several parts leads to extinction or irrecoverable mutilation of the whole and all the involved parts. The most available example of symbiosis is the well-known puzzle-like picture of 'two women in one' as given in [10]. Different overlapping features show either a young or an old woman. The important point is that the features necessary to see the young or the old woman are common to both visions, say for example their 'little chin versus big nose' or their 'necklace versus lips'. If we withdraw these common features of different interpretation, the women disappear, only leaving their common coat and a decorative feather in their hair. This defines a symbiotic occurrence of both women, that is, there exist a subset of features (here, almost all of them, but this is not necessary) such that deleting them from one occurrence induces an unrecoverable loss in both occurrences. As we show in [6], in some domains, a recursive representation may capture symbiosis of information. We explain there that non-primitive recursion (known mostly through Ackermann's function) captures prevention and control in handling symbiotic information. In CSE, *symbiotic feature of information* is one of the most important features.

 ISBN: 978-1-61208-609-5

In contrast to *symbiosis*, we understand by *synergy* a composition the parts of which are not separation-sensitive. Sometimes, synergy is called also modular composition.

B. Deductive and Formal systems

In our work, we introduce a difference between a deductive and a formal system. When a formal system is considered in science, its consistence is considered in terms of non existence of a proof for a formula as well as for its negation in this system. By deductive system we understand a system developed with a concrete real-world application as a model. This means that the consistence of deductive systems is proved by the existence of a concrete model. In fact, a deductive system is in our work viewed as a result of development of a relevant axiomatic system for a particular intended application. In the final stage of development, a deductive system can be viewed as a formal system, however, its completeness or incompleteness is not viewed from a theoretical point of view but from the point of view of a pragmatic evaluation. For instance, Gödel has shown the theoretical incompleteness of Natural Numbers (NAT) (i.e., the set 0, 1, 2, …). However, when we consider natural numbers as a deductive theory the intended model of which are the numbers we all *use*, we can consider NAT as practically complete. In other words, for deductive systems, we introduce the notion of *practical completeness*. Practical completeness means that we all agree on the interpretation (i.e., the model) that is considered. Usually, this is allowed when there is no ambiguity as to the exact meaning of the notions in their practical manipulations. When there is a possibility of such an ambiguity, this indicates that the developed deductive system is incomplete. We say that the notions that are ambiguous are *informally specified* (or *defined*) notions. Selecting a concrete version of the intended model is only possible when the corresponding notions have been completed through a relevant completion of the developed deductive theory.

In order to illustrate the informal character of notions in incomplete theories, let us recall that, in a geometry obtained from Euclid's geometry by eliminating the postulate of parallels, a triangle can be defined. However, in this incomplete 'theory', the sum of the triangle angles may differ from 180°. This means that the notion of triangle is incompletely defined in this particular purged (or mutilated) Euclid's geometry. In practice, it means that an informal definition covers several possible different interpretations of each 'defined' object. It illustrates what we mean by an informal definition. This means also that the completion process of a definition (its emergence) needs to orient a choice – or rather, a construction – of an interpretation that is suitable for each particular problem to be solved. Such a choice, as well as the completion process, is guided by the formalization objectives oriented towards a convenient solution of the informally specified problem. This means in practice that, in any completion process, the goal is to formulate experiments oriented towards a construction of relevant constraints for the intended objective as well as the final delimitation of notions. This example illustrates also why we consider NAT as practically complete system. In order to illustrate the 'practical completeness' of natural numbers, think how all computer driven money exchanges in the world use the same intended model of the natural numbers.

In order to guarantee a rigorous development of deductive systems that corresponds to intended real-world technological applications, we have introduced the notion of pulsation [6]. We are going now to recall its main features.

C. Pulsation

Pulsation concerns incomplete or/and evolutive systems. The usual problem solving strategy can be represented formally by the formula

$$\forall \text{ Problem } \exists \text{ Idea Leads\_to\_a\_solution(Idea,Problem)}. \quad (1)$$

Another, more ambitious, goal can be represented by the formula

$$\exists \text{ Idea } \forall \text{ Problem Leads\_to\_a\_solution(Idea,Problem)}. \quad (2)$$

From implementation point of view, (2) can be written also as the specification

$$\exists \text{system } \forall \text{problem Solves(system,problem)}. \quad (3)$$

There are two main differences between these two paradigms. The first difference is that, in (1), each problem or a class of problems related to a system can have its own solution. However, in (2), a unique, universal solution is looked for. The first paradigm leads to a library of particular heuristics, while the second paradigm results in one universal method.

The second difference is that control and prevention are implicitly present in this second paradigm. This points out towards non-primitive recursion and its simple version, the well-known Ackermann's function. In [6], we show how Ackermann's function can be constructed so that the process of its construction illustrates the meaning of control and prevention of symbiotic information. This analogy is used to define the pulsation, i.e., evolutive improvement of incomplete systems in terms of an infinite sequence of constructed recursive systems - represented by axiomatic theories - such that each system in this sequence contains all the previously constructed systems and is more complex than all the previously constructed systems. So, we have

$$T_0 \subset T_1 \subset \ldots \subset T_n \subset \ldots \quad (4)$$

and

$$T_{i+1} = T_i + A_{i+1} \quad (5)$$

where A_{i+1} is a system of axioms that extends T_i. In terms of the paradigm (2), T_{i+1} allows to solve problems of considered primitive recursive system that had no solution in the previously constructed theories.

III. Cartesian Systemic Emergence

As we said above, CSE is our answer to the following question: Knowing that we are working with informally

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-609-5

specified systems represented by formalized theories, how the process of pulsation helps us to construct these systems?

This section is divided as follows. Firstly, in Section III.A, we shall present the mechanism of CM-formula construction that will introduce a vocabulary necessary for understanding the last two parts. Since this mechanism has been previously presented [4], our presentation here will be based on the suitability of drawing an analogy with CSE, in the same way as we perceive it while building our program synthesis system. We shall present it while insisting on parts that are the most important for working with incomplete theories. Secondly, in Section III.B, we shall present a specification of a concrete situation: A toy, though non-trivial, example. Thirdly, in Section III.C, we shall provide a description of CSE on this toy example.

A. CM-formula construction for CSE

In order to show that CSE is rigorous even without its formal description at the present state of our work, we shall present now its more formalized version in the context of inductive theorem proving (ITP). Indeed, ITP handles recursive notions. In other words, an illustration of ITP framework allows to mirror the emergence work also with informal specifications where the notions involved need to be recursively defined.

Let us proceed to this formalized version known as CM-formula construction and used in the framework of the development of a concrete Symbiotic Recursive Pulsative System (SRPS), as illustrated in [5].

For simplicity, let us suppose that, in an axiomatic theory A, F is a binary recursive predicate and t_1 and t_2 are two terms in A. We restrict here ourselves to primitive recursive theories only. CM-formula construction concerns either proving that $F(t_1,t_2)$ is true or finding conditions under which $F(t_1,t_2)$ might be true. In both cases, for simplicity, we shall speak of proving $F(t_1,t_2)$. CM-formula construction has been initially developed for program synthesis via inductive theorem proving. Thus, it is supposed that some existentially quantified variables may occur in $F(t_1,t_2)$. The mechanism works however also for the cases when there is no existentially quantified variable. In our description we shall point out the differences between these two cases.

Note that when the predicate F is defined in A, its definition expresses the relationship between (or constraints on) the variables x and y so that F(x,y) is true. Informally we may say that the definition of F expresses everything that is needed for F(x,y) to be true. CM-formula construction is based on this simple understanding. On the other hand, in program synthesis, usually $F(t_1,t_2)$ contains existentially quantified variables. This means that a simple unfolding of $F(t_1,t_2)$ using the definition of F and the functions involved in t_1 and t_2 does not lead to explicit values for these existentially quantified variables. The knack here consists in introducing a new type of arguments in the atomic formula to be proven. We call them *pivotal arguments*, since focusing on them enables to suitably handle existentially quantified variables and enables to decompose complex problems (such as strategic aspects of a proof) to conceptually simpler problems while remaining in the context of construction of possibly missing information (such as conditions or new axioms). Among the most usual problems generated is a transformation of a term into another, possibly finding sufficient conditions for this transformation. These pivotal arguments are denoted by ξ (or ξ' etc.) in the following.

Once a pivotal argument has been chosen (first step of the procedure), it replaces, in a purely syntactical way, one of the arguments of the given formula. In this presentation, let us suppose that we have chosen to work with $F(t_1,\xi)$, the second argument being chosen as the pivotal one. In an artificial, but custom-made manner, we state $C = \{\xi \mid F(t_1,\xi)$ is true$\}$. Implicitly, this can be viewed as a desire to find ξ such that $F(t_1,\xi)$ is true. Except the syntactical similarity with the formula to be proven, in this step, there is no semantic consideration postulating that $F(t_1,\xi)$ is true. It simply represents a 'quite-precise' purpose of trying to go from $F(t_1,\xi)$ to $F(t_1,t_2)$ while preserving the truth of $F(t_1,\xi)$.

In the second step, we unfold $F(t_1,\xi)$ using the definition of F and of the functions involved in the formulation of the term t_1. Given the axioms defining F and the functions occurring in t_1, we are thus able to obtain a set C_1 expressing the conditions on the set $\{ \xi \}$ for which $F(t_1,\xi)$ is true. In other words, calling 'cond' these conditions and C_1 the set of the ξ such that cond(ξ) is true, we define C_1 by $C_1 = \{\xi \mid \text{cond}(\xi)\}$. This implicitly means that C_1 is a constructed solution space in which we have to look for a solution of our initial desire to find ξ such that $F(t_1,\xi)$ is true.

We can also say that, with the help of the given axioms, we build a 'cond' such that the formula

$$\forall\xi \in C_1, F(t_1,\xi) \text{ is true.} \qquad (6).$$

The third step relies on the fact that F is recursive and thus a recursive call in its definition suggests that an available induction hypothesis is available to prove (6).

In the third step, using the conditions in C_1 obtained in the second step, the induction hypothesis is applied. Thus, we build a form of ξ such that $F(t_1,\xi)$ is related to $F(t_1,t_2)$ by using the induction hypothesis. This simply means that ξ will be expressed in terms of involved operators. For the sake of clarity, let us call ξ_C the result of applying the induction hypothesis to C_1 resulting in its subset $C_2 = \{ \xi_C \mid \text{cond}_2(\xi_C)\}$. As we just said, ξ_C is expressed in terms of involved operators related to the given problem. Thus, it is no more as abstract as it was case in the first step. C_2 is thus such that $F(t_1,\xi_C)$ is true.

In the fourth step, we proceed to prove that t_2 belongs to C_2. If t_2 does not contain existential quantifiers, this is done by verifying $\text{cond}_2(t_2)$. If t_2 contains existentially quantified variables, this is achieved by a detour. We try to solve the problem $\text{cond}_2(\xi_C) \Rightarrow \exists\sigma\ (\xi_C = \sigma t_2)$, where σ has to provide a suitable instantiation for the existentially quantified variables in t_2. This solution may be recursive. With such an obtained σ we have then to prove $F(t_1,\sigma t_2)$. In other words, we have to prove that ξ_C and t_2 can be made identical (modulo substitution) when $\text{cond}_2(\xi_C)$ holds. If we succeed doing so, the proof is completed. If not, we need to start an additional step.

In this fifth step, a new lemma $cond_2(\xi_C) \Rightarrow \exists\sigma\ (\xi_C = \sigma t_2)$ with an appropriate quantification of the involved variables is generated. This lemma can be seen as a new experience to be performed. In some cases, this may lead to generating missing subroutines (as illustrated in [3]). An infinite sequence of 'failure formulas', i.e., lemmas or missing axioms, may be generated. It is therefore important that the generated sequence may be generalized either by using appropriate tools, some of which are still to be built, or relying on human ingenuity. The generalized formula logically covers the infinite sequence of lemmas or missing axioms and it thus fills the gap that cannot be overcome by a purely deductive formal approach to theorem proving or decision procedures. In the case of generation of missing axioms, the process of the initial theory completion is performed in coherence with the above described model of evolutive improvement of incomplete theories, i.e., pulsation. Be it achieved with or without human interaction, the resulting system is logically coherent by construction.

B. Specification of a toy example

In an ideal world, the transmission of information is a transitive relation. In other words, if x, y and z are variables for some conveyors of a given message, we can formally write

$$\text{conveys}(x,y)\ \&\ \text{conveys}(y,z) \Rightarrow \text{conveys}(x,z). \qquad (7)$$

However, in the well-known children 'phone' play where a first child conveys a sentence to the second, and so on, the last child comes out usually with a sentence that has (almost) nothing to do with the initial message. In a more serious context, Francis Bacon has described several centuries ago how a bad transmission of work of Ancients not only mutilated their work, but also the original fertility (which may be seen as an intention for pulsation mentioned above) has been lost in such a transmission.

In our previous works, we have shown that symbiotic systems are very difficult to describe in the process of their construction. Indeed such a description goes back and forth connecting maybe yet non-existing parts together with already existing parts. This 'circular inductive' behavior has been properly described by Descartes in [2], p. 797. This is the reason why we speak of Cartesian systemic emergence. It concerns the development of symbiotic systems. In tribute to Descartes, who certainly faced this situation too, let us consider the following problem.

Let us suppose that René is a founder of a novel scientific field with a high pulsative potential. Referring back to the bad founders' experience in the past (mentioned by Bacon), he needs to ask himself: How to build some 'works' able to convey the full symbiotic complexity while protecting timelessly the pulsative potential of the field? In a more formal way, René must solve the problem

$$\exists\text{works}\ \forall\text{disciple conveys(René, works)}\ \&\ \text{conveys(works,disciple)} \Rightarrow \text{essential\_of(René)} = \text{essential\_of(disciple)} \qquad (8)$$

Note that this problem has the same logical structure as the second paradigm presented in the form (3), i.e.

$$\exists\text{system}\ \forall\text{problem Solves(system,problem).}$$

Description (8) requires some precisions. First of all, the notions that appear here are not defined in a rigorous way. They are only specified in an informal way in terms of some non-formal criteria. For instance, we know that 'to convey' in this description has to be transitive. However, at present, we do not know whether René, while trying to solve (8), is not forced to adopt some compromises concerning the final delimitation of this notion in this particular context aiming at conveying the full symbiotic complexity while protecting timelessly the pulsative potential of the field. In other words, a solution for (8) has to emerge simultaneously with suitable formalizations (thus, the final definitions) of notions that occur in (8). We shall say that all the notions in (8) are of *evolutive* and *flexible* character. In order to distinguish formal descriptions from descriptions containing evolutive and flexible notions, we call the latter *informal specifications*. CSE concerns thus informal specifications. It will become clear that emergence here cannot be seen as a sophisticated scanning through a given, in advance selected, search space. A solution for (8) does not result from a decision procedure. It is a result of formulating some significant relevant experiments aiming at obtaining simultaneously a concrete value for 'works' as well as a final delimitation of notions occurring in (8). This final delimitation will contain all the compromises adopted during the emergence. This means that solving (8) depends heavily on the ability of the developed mechanisms to create relevant experiments. We have shown in section III.A that CM-formula construction can be seen as a such relevant experiences activator.

C. CSE through CM-formula construction

Let us show now how CM-formula construction is applied to solving (8).

Since we look here for a concrete instance of 'works' that verifies (8), we replace 'works' by a pivotal argument ξ. We thus change the original status of 'works' from unknown variable to pivotal argument. This means underlining the importance of this pivotal argument for creating relevant experiences that may lead to the construction of a suitable solution for 'works'.

We thus obtain

$$\text{conveys(René},\xi)\ \&\ \text{conveys}(\xi,\text{disciple}) \Rightarrow \text{essential\_of(René)} = \text{essential\_of(disciple)} \qquad (9)$$

Note that presently, in (9), the predicate 'conveys' as well as the function 'essential_of' are evolutive and flexible, as opposed to the formal operators handled in CM-formula construction. The two operators 'conveys' and 'essential_of' are here specified only informally by some set of sentences that represent the most general constraints that concern these notions. This set of sentences is not a set of formal definitions, as takes place with the formal operators handled in CM-formula construction. This means that instead of the evaluation of these operators, as we have seen in CM-formula construction, we shall replace these notions by their informal descriptions. We shall denote by $Descript_t$ the set of sentences specifying 'to convey' and by $Descript_e$ the set of sentences specifying 'essence_of'. Then, in a purely artificial way we shall write

$$Descript_t(René,\xi) \ \& \ Descript_t(\xi,disciple) \Rightarrow Descript_e(René) = Descript_e(disciple) \quad (10)$$

Note that we need now to start to 'move' things in a way similar to CM-formula construction. In other words, we have to start to create some relevant experiences that would enable the emergence of less informal descriptions of our operators and more concrete information about ξ. Since René is here a constant of the problem, we shall focus on the universally quantified variable 'disciple'. In order to create relevant experiences, we shall take a concrete example of 'disciple', say d_0, and try to find a particular informal solution for the problem

$$Descript_t(René,\xi) \ \& \ Descript_t(\xi, d0) \Rightarrow Descript_e(René) = Descript_e(d0) \quad (11)$$

The peculiarity of the informal solution we are looking for lies in the fact that we have to resolve the constraints expressed by this problem, while keeping in mind that the solution we are looking for must be applicable (thus relevant) to our general problem (8). This means that we consciously work here simultaneously on the above mentioned two paradigms (1) and (2). This is necessary to guarantee that we are working in the framework of a pulsation model. In other words, we try to find some refinement of the operators specified by their descriptors in (11) so that the resulting refinements can be, without loss of generality, applied also to other instances of 'disciple'. Our goal is thus to determine what are the basic problems (BP) that have to be solved while keeping our goal of solving (11) for d_0 as well as for other instances of 'disciple'. These basic problems will not only give more information about ξ that is looked for but also ξ will be expressed in terms of the solutions obtained for these BP. Thus, more concrete information about the final 'works' will be found. Note that it may happen, as it is the case in CM-formula construction, that new problems are discovered while solving BP.

The process described so far is however only a beginning. Since solutions for BP are still informal, we continue creating experiences for other instances of 'disciple'. Solving, even though incompletely, these BP in a coherent manner for many highly varied instances of 'disciple' is a good start for beginning an informal development of T_0 in pulsation sequence T_0, T_1, T_2, …

For some instances of 'disciple' it may happen that new problems (NP) are recognized. Thus, a coherent solution with the solutions for BP has to be built. It may happen that the informal solutions found for BP are shown unsuitable for an extension with respect to NP. Thus, based on this constructive feedback, new solutions for BP ∪ NP need to be constructed. The flexibility of informal formulations of BP and NP allows, in general, to generate a new reformulation that covers a real possibility of coherently solving BP ∪ NP. Let us denote by ALLNP the set of all problems generated for all considered experimental instances of 'disciple'. The above process is thus repeated on many instances of 'disciple' so that the finalized solution for BP ∪ ALLNP represents the basis for an implementation of the solution expressed in the axiomatic framework of T_0. Of course, since we cannot examine all concrete instances of 'disciple', T_0 is potentially incomplete. This means that the notions constructed for T_0 are prone to a modification. However, with respect to the pulsative model of their development in the construction process, all these formalized notions are flexible for further improvement. In other words, the pulsative model of development, as defined here, guarantees the flexibility of the notions. The process is both flexible and rigorous.

Note that this process indeed illustrates how, during the construction process, we consider simultaneously the two above paradigms (1) and (2). In this emergence process, we simultaneously follow the general goal specified by the second paradigm (2) and we work with experiences in the framework of the first paradigm (1).

We call *oscillation* this particular feature of CSE in the pulsation model. Oscillation is thus considered at a local level for a particular T_i, while pulsation is a model that covers an infinite sequence of theories T_j verifying the above mentioned conditions ($i, j \in NAT$).

IV. CSE AND COMPUTATIONAL COGNITIVE SCIENCE

Cartesian systemic emergence seems to us heavily related to the topic of human reasoning mechanisms, cognitive and computation models, human cognitive functions and their relationship, and even to modeling human multi-perception mechanisms.

In [1], Bermúdez pointed out the influence of Computer Science on the development of Cognitive Science Paradigms. Cartesian Systemic Emergence, as an example of symbiotic thinking (i.e., simultaneously focusing mentally on several different topics), represents a way of thinking which, as far as we know, is not studied in Cognitive Science. One cause may be that symbiotic thinking is considered as not achievable in Cognitive Science. For instance, John Medina claims in [9] that our brain *is not conceived* to handle simultaneously several different topics. We may agree that it may be impossible for a non-trained person to perform two different physical challenging tasks. We believe, however, that this opinion, when generalized to mental processes, is born from existing brain synergic models (that are thus non symbiotic) as well as from some misinterpretations of external observations.

In particular, the observation of symbiotic modules in action may have problems with comprehending the emergence of a solution in an active performance. At least, its explication is bound to seem obscure and a clear (but inexact) presentation of its functioning tends to explain the modules roles once their interaction is completed, as if they were independent of each other, i.e., using a synergic model.

The problem of spotting symbiotic interaction, in itself, is therefore hard to tackle. This difficulty becomes obvious when psychoanalysis describes harmful relationships of the sick person with his/her self. A solution to the problems seems to become possible when, as suggested by famous psychoanalyst C. G. Jung, a symbiotic solution starts to be built following the rule that: "… it is as much a vital necessity for the unconscious to be joined to the conscious as it is for the latter not to lose contact with the unconscious." ([8], section 457, p. 298). We could use a similar way of speech to express the fact that two modules of an emerging system should not 'lose contact' one with the other.

It follows that Cartesian systemic emergence might well be part of a challenge for Cognitive Science. This will be achieved by developing Cognitive Science models that capture all the essential characteristics of CSE, by finding methods and tools to study the emergence process in an active performance and developing on-purpose computational models for this particular way of thinking. Even though the topic is challenging, we are convinced that a strong desire or need to solve problems that CSE suggests to Cognitive Science will lead soon or later to a fruitful empowerment of Cognitive Science. We hope that the models presented in the present paper might be of help in such a difficult task.

V. CONCLUSION

In this paper we have introduced and exemplified a general method called Cartesian Systemic Emergence (CSE) that implements strategic aspects of pulsation (introduced previously in [6]). We have shown that the main problem to handle in CSE is working with informal specifications for the purpose of their formalized delimitation in coherence with the pulsation model. We have illustrated that, in this delimitation context, CSE is very convenient as an experiences activator. We have also mentioned the main problems and challenges addressed by CSE to various fields of Cognitive Science.

We have been led to CSE in two complementary ways. Firstly, CSE is a by-product of our research on Program Synthesis from their formal specifications as presented by Manna and Waldinger in [11]. Secondly, our efforts in this domain, so to say, 'forced' us to handle the problem of automating recursive programs synthesis in incomplete domains [4], which, in turn, led us to recognize the necessity of CSE. In other words, this paper is a result of our formalization of the method that guided, from the start, our program synthesis research. However, we have also recognized the great potential of this method to model and solve other real-world problems. CSE thus seems a know-how that might show its importance in all incomplete domains where solving problems requires rigor, as well as pragmatic considerations and experimentations.

Cognitive Science already interacted with computer science in a topic that somewhat 'looks like' ours, namely the so called Emergent Computing that was defined in the beginning of the nineties by Stephanie Forrest [12] by "Emergent computation is proposed in the study of self-organization, collective and cooperative behavior." The comprehensive review of Xiao, Zhang and Huang [13] provides different views on how to define it. None of these definitions alludes to any symbiosis imposed on Emergent Computing systems, a theme central to our research. It is still an open problem, and an interesting research field, to decide whether human brain does or not use symbiotic modules for thinking, at least for deep thinking, such as the one of a mathematician trying to prove a still unproved theorem, where the proof the theorem demands completion of some currently available incomplete specification of what is to be proven. At any rate, the views presented in this paper constitute a challenge for Cognitive Science, and we cannot presently do more than hoping it will be a fruitful challenge.

ACKNOWLEDGMENT

We thank to Michèle Sebag and Yannis Manoussakis for a moral support. Anonymous referees of the submitted version provided a very useful feedback.

REFERENCES

[1] J. L. Bermúdez, Cognitive Science: An Introduction to the Science of the Mind, Cambridge University Press, 2014.

[2] R. Descartes, Complete Works (Oeuvres philosophiques), Édition de F. Alquié, Vol. 1, Bordas, 1988.

[3] M. Franova, "A Construction of Several Definitions Recursive over the Variable under the Exponent for the Exponent function," Rapport de Recherche No.1519, L.R.I., Université de Paris-Sud, Orsay, France, June, 2009.

[4] M. Franova, "Cartesian versus Newtonian Paradigms for Recursive Program Synthesis," International Journal on Advances in Systems and Measurements, vol. 7, no 3&4, pp. 209-222, 2014.

[5] M. Franova and Y. Kodratoff, "Cartesian Handling Informal Specifications in Incomplete Frameworks," Proc. INTELLI 2016, The Fifth International Conference on Intelligent Systems and Applications, pp. 100-107, 2016.

[6] M. Franova and Y. Kodratoff, "A Model of Pulsation for Evolutive Formalizing Incomplete Intelligent Systems," Proc INTELLI 2017, The Sixth International Conference on Intelligent Systems and Applications, pp. 1 - 6, 2017.

[7] K. Gödel, "Some metamathematical results on completeness and consistency" in J. van Heijenoort: From Frege to Godel, A source book in mathematical logic, 1879-1931, Harvard University Press, Cambridge, Massachusets, 592-618, 1967.

[8] C. G. Jung, Symbols of transformation, Princeton University Press, 1956.

[9] J. Medina, Brain Rules, Pear Press, 2008.

[10] E. W. Weisstein, Young Girl-Old Woman Illusion; From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/YoungGirl-OldWomanIllusion.html . (accessed January 2, 2018)

[11] Z. Manna and R.Waldinger, "A Deductive Approach to Program Synthesis", ACM Transactions on Programming Languages and Systems, Vol. 2., No.1, January, 1980, 90-121.

[12] Stephanie Forrest, "Emergent Computation: Self-Organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks" 1991, Physica D, Vol. 42, Nos. 1–3, pp.1–11.

[13] Xiao R., Zhang Y. and Huang Z., "Emergent computation of complex systems: a comprehensive review", Int. J. Bio-Inspired Computation, Vol. 7, No. 2, 2015, pp.126–139.

