Cartesian Systemic Pulsation –

A Model for Evolutive Improvement of Incomplete Symbiotic Recursive Systems

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

Abstract— **Developing and using Symbiotic Recursive Systems (SRS) concerns various incompletely defined domains requiring handling prevention and control. This means that, in the innovation process, it is important to avoid decline and obsolescence as it happens with modern paradigms of innovation. This paper presents the specificities of SRS that call for a new model of evolutive improvement. Such a new model needs to handle prevention and control in SRS as well as in the improvement process. The paper presents such a model we call 'Pulsation'. Symbiotic Recursive Pulsative Systems are then SRS including the process of Pulsation.**

Keywords-pulsation; Symbiotic Recursive Pulsative Systems; systemic recursion; Ackermann's function; security; practical completeness.

I. INTRODUCTION

We work under the hypothesis that human knowledge is and hopefully will always be incomplete. This incompleteness favours innovation and discovery. One distinctive feature of modern innovation is acceptance of a rapid obsolescence and decline of the technologies or systems it develops. In many cases, however, this last feature cannot be accepted. As an example, dealing with symbiotic recursive incomplete systems needs a robustness which asks for special attention. This paper addresses this problem in presenting a new model for an evolutive development of new symbiotic recursive systems or technologies. It has been first introduced in [1] where we called it: Pulsation. Its basic features are – at each step of development –

- a kind of timelessness of previous achievements (no obsolescence);
- possibility for future new improvements (no decline);
- focus on prevention and control (particular rigor).

It is true that modern science and the philosophy of innovation tend to deal with

- synergy and modularity instead of symbiosis,
- non-recursive complexities instead of recursion, and
- constant change or mutations instead of what we call Pulsation.

While these modern science notions are extremely useful and relevant for many real-world applications, they however cannot replace symbiosis, recursion and Pulsation without harmful consequences. This means that the approaches of these two groups of notions are complementary and non-competitive.

For convenience, we shall call *first group* the one including the notions

- symbiosis
- recursion
- Pulsation

and *second group* that of including the notions

- synergy, modularity
- non-recursive complexities
- change, mutations.

This paper gives below a systemic description of the first group of notions.

In order to illustrate Pulsation in action, we present it in the framework of Symbiotic Recursive Systems (SRS). We show that these systems are particularly suited to represent potentially incomplete SRS that formalize real-world applications. An example of such a real-world application will be given.

We shall also present reasons for naming *Cartesian Intuitionism* a systemic paradigm based on the first group and *Newtonian approach* a systemic paradigm based on the second group of notions. We have called *Symbiotic Recursive Pulsative Systems* (SRPS) the systemic nucleus that is the basis of Cartesian Intuitionism. One of our goals is to introduce this notion that is, to the best of our knowledge, not referred at elsewhere.

The paper is organized as follows.

Section II specifies the notion of symbiosis as understood in this paper. Since symbiosis is related to our understanding of the notion of theory we shall introduce, with help of symbiosis, a difference between formal and deductive theory. Section III introduces recursion as a way of representing action, control and prevention. It explains what we mean by systemic recursion. Section IV introduces the

notion of oscillation as representation of one-level creation process used in Pulsation presented in Section V. Section VI presents a motivation for introducing the systemic difference between Newtonian approach and Cartesian Intuitionism. It will become clear why the latter expression is used to describe the systemic science relevant to SRPS. Section VII presents an application for which Pulsation is relevant. Section VIII relates ancient systemic thinking to Pulsation and SRPS.

II. SYMBIOSIS : 'VITAL INTERDEPENDENCE'

Symbiosis is a particular composition. In this section we shall give a definition of symbiosis as used in this paper. We shall also compare symbiosis to other kinds of composition, namely synergy and fusion.

By symbiosis we understand a separation-sensitive composition of two or several parts. This means that a separation of one or several parts leads to extinction or irrecoverable mutilation of the whole and all the involved parts, as will be illustrated below.

In contrast to this, by synergy we understand a composition of parts that is not separation-sensitive. Sometimes, synergy is called also modular composition.

In case of fusion, the resulting composition is homogenous. It does not allow to recognize the involved parts, they are blended together, such as in fusion of metals.

We shall now give several examples.

A. *Pictorial Symbiosis*

Let us consider the following version 'two-women-in-one' of 'Devinettes d'Épinal' (see also [18]):

Figure 1. 'Two-women-in-one' picture.

Several overlapping features may reveal two different human faces. In other words, this picture is a composition of two parts. The important point is that the features necessary to see a 'young' or an 'old' face are common to both visions (i.e., parts), though they may be differently interpreted. Here, the feature 'little chin' in one is interpreted as 'big nose' in the other, the ear of one is interpreted as the eye of the other, the necklace of one becomes the mouth of the other, and the couple one eyelash + small nose of one becomes the two eyelashes of the other. If we withdraw from the picture all of these common features, as shown below, then a human face recognition 'system' may still reconstruct a human face though it becomes unable to rebuilt these common features of different interpretations, the 'two-faces-in-one' picture is thus destroyed and irrecoverable. This represents a symbiotic pictorial occurrence of both faces, that is, there exist a subset of features of these two parts (here, four of these features, but this is not necessary) such that deleting them from one occurrence induces an unrecoverable loss of the picture intent.

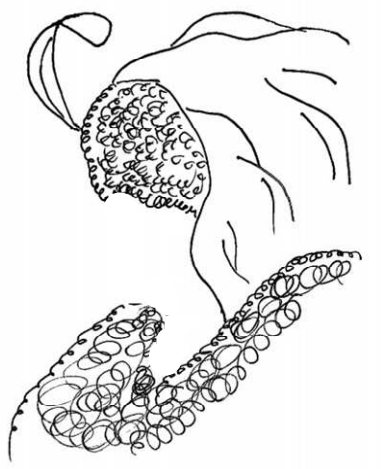

Figure 2. Mutilated 'two-women-in-one' picture.

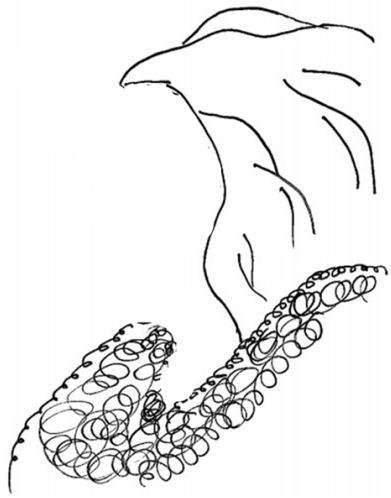

Figure 3. Irreversible mutilation of 'two-women-in-one' picture.

The feature 'hair decorated with a feather', as in Figure 2, is common to the two faces, and may evoke an incomplete female face. If we withdraw this last feature, as in Figure 3, then a human recognition system is at lost at recognizing

something really significant (a strange bird, perhaps?). While Figure 1 represents several overlapping features of two human faces, from the point of view of symbiosis, it is important to understand it as a composition of two parts (old woman, young woman). Therefore, it is relevant that eliminating one part (and not one feature) leads to the destruction of the whole and the remaining parts. Here, for instance, Figure 3 can be seen as a result of eliminating old woman. Young woman disappears as well.

B. Representational Symbiosis

A careful study of primitive notions in Euclid's geometry [20] shows that these notions are symbiotic. This means that eliminating even one notion would either render meaningless the resulting system (i.e., extinction of the resulting system) or the meaning of the resulting system would be completely different (i.e., irrecoverable mutilation).

It should be noted here that the example of Euclid's geometry illustrates well the fact that the constituents of a symbiotic system need to be handled as symbiotic in the construction process of such a system. However, after its successful final creation, the use of these notions may, in some cases, be modular. For instance, when we use the notion of point while working in Euclid's geometry, we do not need to be aware of the symbiotic dependence of this notion on other notions of the same geometry. However, such awareness was necessary for Euclid when he created this geometry. This point is further illustrated in Section II.D.

C. Intentional symbiosis

As far as intentional symbiosis is concerned, we consider it exclusively in relation to human creations. A detection of intentional processes in nature is out of the scope of this paper.

We could perceive a slight glimpse of intentional symbiosis in all the above, somewhat static, examples. Even though present, a rather procedural character of intentional symbiosis was not mentioned. In Section III., we shall give an example of the construction of Ackermann's function, where such an intention can be easily described. As far as real-world applications are concerned, usually their development is driven by synergic thinking. This synergic thinking is convenient when all the tools are available and the resulting system consists in a novel composition of these tools. When it happens that new basic tools need to be invented, symbiotic thinking opens the way to new conceptual switches enabling some breakthrough from the usual thinking. This underlines why symbiotic thinking is an asset for creating new technologies.

D. Deductive and Formal systems

The above example of representational symbiosis, namely that of Euclid's geometry, inspires us to introduce a difference between a deductive and a formal system. Indeed, when a formal system is considered in science, its consistence is considered in terms of non existence, in this system, of a proof for a formula as well as for its negation. 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 asserted 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 the set 0, 1, 2, …. However, when we consider natural numbers NAT as a deductive theory the intended model of which are the numbers we all *use*, i.e., the numbers represented by Peano's arithmetic, we can consider NAT as being practically complete. Indeed, in this practical case, we need to consider symbiotic relationship of numbers and axioms defining the addition, the multiplication, and the induction principle. 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. 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. When, on the contrary, such an ambiguity is possible this indicates that the developed deductive system is incomplete. Selecting a concrete version of the intended model will only be possible when the corresponding notions have been (at least partially) completed through a relevant completion of the developed deductive theory. By partial completion of definitions of notions, we mean definitions that guarantee practical completeness of the considered system.

In order to illustrate the informal (or incomplete) 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. Thus, deductive theories are characteristic by their origin in a concrete application and their incompleteness is not a limitation, when practical completeness only is requested.

Our model of Pulsation presented in Section V is developed with the aim to guarantee a rigorous development, or completion, of deductive systems that corresponds to intended real-world technological applications. It will become clear in Section V that Pulsation deals also with another feature of practical completeness, namely the

availability of solutions for practical problems that can be described in the theory.

The notion of symbiosis is in our work an emergent notion. This means that the specification of symbiosis presented in this section will have to be refined symbiotically with the future development of theory of SRPS. Namely, we still need to develop and present strategic and practical aspects of the creation of symbiotic systems. In [13] we call Cartesian Systemic Emergence this process. We illustrate there also one particular strategic feature on a simple example.

III. RECURSION

In this section we present a minimal, but sufficient basis for understanding *systemic recursion* for Symbiotic Recursive Pulsative Systems (SRPS). Moreover, we will illustrate how recursion represents not only actions, but also particular forms of control and prevention.

Mathematical and computational recursion handle recursion from formal or programs efficiency points of view (see [15], [14]). Recursion in these cases is a known tool and not a science. In contrast to this, systemic recursion is a *science of know-how for creating recursive systems* that are useful for real-world applications in various domains. We shall point out the main features of systemic recursion through out this paper.

A. Preliminary definitions and notions

It is known that recursion is a particular way to represent, by a finite set of rules, potentially infinite systems and processes (actions and creations). These rules are expressed in terms of basic action or creation operators called constructors.

In Mathematics and Computer Science such constructors are usually known, available or easily attainable in standard know-how. In contrast to this, know-how of systemic recursion lies in a progressive invention of on-purpose constructors of a goal system in dependence with progressive invention of a *formal specification* of the goal system. Indeed, in systemic recursion, we start to build a system from an *informal specification* of the goal system. The notion of Pulsation presented in Section V is important for understanding the systemic emergence of a formal system from its informal specification.

By informal specification of a real-world application we mean that it is not yet a formalized description. It is, for instance, the case for technological visions or some pragmatic formulations of technological needs. An informal specification is usually not formulated by a mathematician or by a computer scientist. It is formulated by a visionary person or by experts expressing a need for some new solving tools in their domain. Informal specification describes a rather vague 'what' of the intended application, tool or system.

When an informal specification of a technological vision is known, the 'how' of its implementation is usually not available and may even be unknown or impossible in standard know-how. Therefore, in systemic recursion, we speak of creation or even emergence rather than of development. Since not all the constructors of the intended system are known at the beginning, a long period of preliminary research of a sufficient set of relevant constructors always precedes the implementation of an experimental prototype. The final development might even require some complementary inventions.

The goal of systemic recursion is to pass from an informal specification to a satisfactory formal specification of the goal together with the relevant on-purpose knowledge and know-how, i.e., the 'how' or 'procedural science' of the actual development. A formal specification expresses all the knowledge necessary for a final implementation of the 'what' that has been at first only informally specified.

This means that, in systemic recursion, a formal specification is an agreed upon compromise between the visionary and the developers of the considered informal specification. This compromise is built up progressively. It cannot be created in advance. This is because, in systemic recursion, research is pluridisciplinary in the sense that it crosses the traditional boundaries between disciplines and it progressively constructs its own on-purpose knowledge, know-how and boundaries. This on-purpose systemic knowledge is symbiotic.

In the following section we shall show how recursive actions may be considered as a way to represent not only actions but also a particular kind of control.

B. Representation of a particular control

In this section we shall present an example that illustrates how recursion captures in itself, by symbiotic dependency, all the secondary effects of a simple recursive procedure computation. Let us point out that we emphasize here symbiotic information, not symbiotic computation.

Let us consider the following simple problem. On a sufficiently big table consider a stack of blocks a, b, c, d and e as shown in Figure 4.

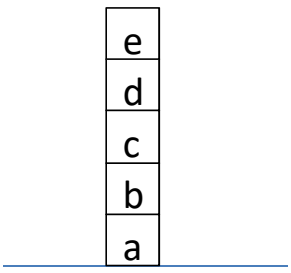

Figure 4. A stack of blocks.

We say that a block m is clear if there is no other block on m. (In Figure 4. block e is clear.) There can be at most one block on the top of the other. If n is on the top of m we say that n is top of m written as: n = top(m). Let us consider the following procedure makeclear:

```
makeclear(x) =
   if x is clear then end
   else
      if     top(x) is clear
      then   put(top(x)) on table
      else   first makeclear(top(x))
             and
             then put(top(x)) on table
```

It can easily be checked that makeclear(b) results not only in clearing the block b but also in the situation where blocks c, d and e are clear and on the table. This means that the procedure makeclear contains in its description not only its direct effects (such as: the block b is cleared) but also the full description of all the secondary effects of any action performed. In Figure 5 these secondary effects are that the blocks c, d and e are on the table.

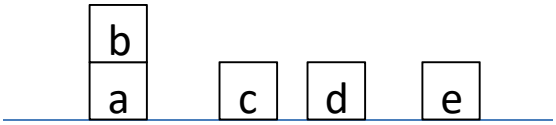

Figure 5. The environment after clearing b.

For some primitive recursive procedures the secondary effects do not modify the environment, but this should not be a barrier for a general perception of primitive recursive procedures as invisible procedural 'seeds' containing symbiotically related the effects (i.e., the results of the computations) and the secondary effects (i.e., the consequences of the computation of a particular value). Therefore, implementing recursive procedures is interesting in all the environments where the control over the secondary effects is important.

It may be not straightforward to see in what sense we speak of symbiotically related effects and secondary effects. Let us recall that symbiosis of information means here that if we take away one piece of information, the global information becomes distorted, or mutilated.

Let us consider therefore the instruction makeclear(b). This means that, step-wise, all blocks above b have to be put on the table leading to the result that the blocks c, d and e are on the table. Let us suppose that we take away the information that the block d is on table. However, in the progression of the procedure, if we call makeclear(b) first, we have to call makeclear(c), then makeclear(d). Since top(d) is e and it is clear, we may put e on the table. d = top(c) is now clear, but we cannot put it on the table since this information has, by our assumption, been withdrawn. This means that the whole process is stuck and we then cannot put block c on the table. So, the resulting information is mutilated since it does not express all the potential of the procedure makeclear. Moreover, there is no evidence that the information that the block e is on the table is related to the procedure makeclear. It could, in principle, be related to some other procedures.

The above procedure makeclear is an example of primitive recursion. A recursion that is not primitive goes even further in representing symbiotically information that concerns control, rigor and reproducibility. Ackermann's function is the simplest example of a non-primitive recursive function. Therefore, it is a suitable representative for explaining how non-primitive recursion modelizes a particular kind of Pulsation in SRPS.

In the following part we shall give a formalized presentation of the Pulsation starting by a presentation of a construction procedure that generates Ackermann's function. It will become clear how this construction and the notion of Pulsation are linked together.

C. A Construction of Ackermann's Function

The idea to modelize Pulsation by Ackermann's function comes from the understanding of how this function may be constructed. The practical use of this function becomes then exploitable by a 'simplification' of the computation of its values using the knowledge of its construction process.

Let 'ack' be Ackermann's function defined, as in [19], by its standard definition, i.e.,

$$ack(0,n) = n+1$$
$$ack(m+1,0) = ack(m,1)$$
$$ack(m+1,n+1) = ack(m,ack(m+1,n)).$$

We shall show here how this function can be constructed.

Since ack is a non-primitive recursive function, by definition of non-primitive recursion, it is a particular composition of an infinite sequence of primitive recursive functions. We shall thus define a function ack' as a particular composition of an infinite sequence of primitive recursive functions and it will become clear why the definitions for ack and for ack' are identical.

By definition, each primitive recursive function f is a composition of a finite number of primitive recursive functions and of f itself. Let us therefore construct such an infinite sequence of primitive recursive functions f_0, f_1, f_2, …, f_n, f_{n+1}, …. We define

$$f_0(n) = n+1$$
$$f_{i+1}(n+1) = f_i(f_{i+1}(n))$$

for each i from 0, 1, 2 …. We are thus able to define a new function ack' as follows: ack'(0,n) = $f_0(n)$ and ack'(m+1,n+1) = $f_{m+1}(n+1)$. Note that ack'(m+1,0) is not yet defined. Since we want ack' to be a non-primitive recursive function, we need to guarantee that it cannot be reduced to any of f_i. In order to do so we shall simply perform a

diagonalization [15] on this infinite sequence of functions by defining

$$f_{i+1}(0) = f_i(1).$$

In other words, we define

$$ack'(m+1,0) = f_m(1).$$

By this construction, we see that f_{i+1} is more complex than f_i for each i. It is obvious that

$$ack'(m,n) = ack(m, n) = f_m(n).$$

This construction is at the same time a guarantee that ack is not primitive recursive, since it is indeed a composition of an infinite sequence of primitive recursive functions each of them more complex than those before it and ack cannot be reduced to any one of them. As a by-product, we have thus simplified also the standard presentation of the non-primitive character of ack which is usually done by a proof by a projection of Ackermann's function ack into a sequence of primitive recursive functions $a_m(n) = ack(m,n)$ and showing that ack grows more rapidly than any of these primitive recursive function (see [19]). The difference thus lies in our use of an indirect construction (instead of a projection) and relying on a progressive diagonalization. To our best knowledge, this construction with a use of progressive diagonalization was not presented so far. Note that the notion of Pulsation that refers to this construction of Ackermann's function has no relation to measures of the computation complexity of a function, such as Ritchie's hierarchy [17]. Complexity and efficiency are thus out of the scope of this paper.

D. Disentangling Ackermann's Function

The above construction of Ackermann's function shows immediately that the computation of its values, for given m and n, using non-primitive recursive definition can be 'simplified' - or, rather, replaced - by a definition of m primitive recursive functions obtained by a suitable macro-procedure.

Our recursive macro-procedure will simply compute, step by step, each of the values $f_{i+1}(0)$ in advance and will define the whole f_{i+1} with this already computed value. This may not lead to a fast computation but we are not concerned now with computational efficiency of this way of proceeding, only by its practical feasibility and reproducibility.

We define a macro-procedure, we call ack_macro. It uses the standard LISP procedures add_to_file and load_file. The procedure add_to_file(*text*,*F*) adds the *text* at the end of file *F*. The procedure load_file(*F*) loads file *F* in order to make computable the functions written in this file. We create at the start an auxiliary file F that stores the functions f_i generated by ack_macro. Our macro-procedure ack_macro(m,n) uses the infinite sequence of functions defined above as being representative of Ackermann's function.

```
Step 1:
   text:= { f0(n) = n+1 }
Step 2:
   Create file F (empty at start) and
           add_to_file(text,F)
   load_file(F)
Step 3:
   i:=0
   aux:= compute the value of fi(1)
Step 4:
   text := { fi+1(0)= aux
           and fi+1(n+1)= fi(fi+1(n))}
   add_to_file(text,F)
   load_file(F)
   aux := compute the value of fi+1(1)
   i:= i+1
   if i < m
   then Go to Step 4
   else stop
```

Figure 6. A macro-procedure for computing particular values of ack

ack_macro(m,n) is now completed and file F collects the definitions of m primitive recursive functions. We are now able to compute $ack(m,n) = f_m(n)$, where the definition of f_i from file F is used for all i = 0, 1, …, m.

On internet, we can find several programs that compute Ackermann's function values faster and much further than our presented macro-procedure. The problem with those programs is that they are based on the knowledge that Ackermann's function can be represented as a generalized exponentiation function. The advantage of our presentation lies in its suitability for practical purposes in the following sense. As we shall see with the notion of Pulsation, complex real-world problems may require modelization by non-primitive Ackermann's like programs that will not be reducible to an arithmetic generalized exponentiation. In other words, it is useful to consider non-primitive recursion that is not defined for natural numbers only but which is defined for all practically useful and exploitable recursive systems.

E. Prevention and Control in Recursion

We have seen above, in the example of program makeclear, that primitive recursion captures the effects (the value of the computation) and the secondary effects (the consequences of the computation that are in fact the intermediary values generated by the same procedure). We have also seen that the non-primitive recursive Ackermann's function is obtained using a diagonalization procedure. This diagonalization brings forward complementary information about the process of this symbiotic information in recursion. Since diagonalization is a meta-level procedure, we understand this complementary information as a kind of meta-level prevention from primitive recursion reducibility. While lack of control is accepting to ignore some secondary

effects of the computation, lack of prevention is accepting to ignore some secondary effects of secondary effects of computation. They are taken into account in advance (by diagonalization and generalization) and thus they are related to prevention.

It is interesting to note that some scientists may intuitively 'feel' that Ackermann's function provides a model of human thinking of 'everything' for a particular situation. The above mentioned makeclear program shows that this intuition can be presented in terms of symbiosis of the information included in a particular situation. Note that the above macro-procedure (Figure 6) only simplifies the computation of thinking of 'everything'. In order to illustrate this simplification of the computation we may mention that, as it can be checked, the computational trace of the value for ack(3,2) using standard definition shows (see [8]) that the value ack(1,1) is computed twenty-two times for obtaining the result of ack(3,2). This is not the case for $f_3(2)$ simplified computation. It is however necessary to understand that the overall complexity of this situation remains the same since, in order to be able to 'simplify' (i.e., to define the above macro-procedure), we already need to have available Ackermann's function equivalent sequence of f_i. In other words, the principle and effectiveness of 'thinking of everything' are globally unaffected. The simplification concerns only focusing on one particular local level defined by the two values a and b instantiating Ackerman's variables. Of course, the macro-procedure is general, but for a and b given, it generates only the finite sequence of primitive functions $f_0, f_1, \ldots, f_a$.

This makes explicit that 'thinking of everything' keeps its order of complexity after applying our simplification. Systems requiring a simultaneous handling of prevention and control factors such as information security systems or strategic planning in flexible environments are practical examples of a problem requesting to think of 'everything' (see [16], [10]).

F. Systemic recursion

In the previous sections, we have presented what can be seen as a recursive structure. In a recursive structure there is an obvious so-called base step element (for example, 0 in NAT). We shall speak about systemic recursion when a system is defined or constructed recursively, but there is no such an obvious 'base step element'. It is mostly the case when all the parts of the system may themselves be considered as symbiotic systems. In other words, in systemic recursion, symbiotic constructors are also systems, as will be illustrated by the following example – a simple one though it illustrates the complexity of systemic recursion:

Consider a method M that is recursively defined in terms of a finite number of complex symbiotically dependent procedures $R_1, R_2, \ldots, R_n$. Then, a systemic equation for such a method can be represented as follows:

$$M = R_1 + R_2 + \ldots + R_n + M.$$

Of course, rules $R_1, R_2, \ldots, R_n$ may themselves be recursive procedures calling M as well. This emphasizes the difference between systemic recursion and linear-like, tree-like or network-like representations.

The above construction of Ackermann's function and our particular disentangling of its computation by a primitive recursive macro-procedure allow us to consider it as a model for a particular kind of Pulsation. The notion of oscillation, defined in the next section, provides an informal background for the notion of Pulsation as described in Section V.

IV. OSCILLATION

In scientific fields, an obvious basic paradigm, for a given problem, is looking for ideas that possibly lead to a solution. This behavior reflects the belief that the following formula is valid

$\forall$ Problem $\exists$ Idea Leads_to_a_solution(Idea,Problem).

We shall call this formulation: "first paradigm."

However, another and rather unusual (except in Physics) paradigm is to find an idea that provides a solution for all problems. We shall show how Ackermann's function provides a model for this second paradigm. Similarly to Ackermann's function, in a sense, it is a kind of 'thinking of everything'. First, however, let us express this paradigm by the formula

$\exists$ Idea $\forall$ Problem Leads_to_a_solution(Idea,Problem).

We shall call this formulation: "second paradigm."

The difference between these two formulas lies in the fact that, in this second case, the 'Idea' obtained is unique, while in the first formula each problem can use its own Idea.

We call **oscillation** this approach of *symbiotic* switching between the two above paradigms. It corresponds to a representation of a one-level creation process.

The oscillation may be performed in the following way. We start to consider a large variety of problems for which we try to find an idea for a general solution to all of them. This solution needs to be open to the need of a further improvement. We shall, in the next section, introduce Pulsation as being a particular kind of such an improvement,

V. PULSATION

The above sections will help us explaining how Ackermann's function enables us to formally specify the notion of **Pulsation**, i.e., a particular kind of 'evolutive improvement'. This is interesting not only from the point of view of building particular deductive theories for unknown domains but also for understanding the difference between

revolution, innovation and evolutive improvement in this building process.

Let us consider a potentially infinitely incomplete theory. In unknown environments this may be seen as a framework for potentially infinitely incomplete theories. Building a deductive theory becomes then a process of *suitable completions* of a particular initial theory T_0. We shall say that this theory T_0 is **practically complete** when it formalizes solutions of practical problems that have been met so far. It implicitly means also, as mentioned in Section II.D, that a non ambiguous specific model is available. Since the theory is potentially incomplete, sooner or later we shall meet a problem that cannot be solved in the framework of T_0. In the vocabulary of scientific discovery we may say that we need a conceptual switch (a new axiom or a set of axioms) that *completes* T_0. Note that we speak here about *completion* and

- not about a *revolution* - which would mean in a sense rejecting T_0
- not about a *innovation* - which may simply amount to a particular reformulation of T_0, not necessarily coherent with T_0.

This completion T_1 has to contain T_0 and thus it must be coherent with T_0. However, since a new conceptual switch guarantees that T_1 is more powerful than T_0, we consider this particular kind of completion as a suitable model for one step of *improvement* in our search for suitable completions. Since we consider here a potentially infinitely incomplete theory, we can then see **Pulsation** as an infinite sequence of theories T_0, T_1, …, T_n, … . In this sequence, T_{i+1} completes and is coherent with T_i for all i = 0, 1, 2, …

We have seen that, in the infinite sequence from which Ackermann's function is built, the function f_1 relies on (is coherent with) f_0, and f_{i+1} relies on f_i for each i. We can therefore see that Ackermann's function really provides a model for evolutive improvement (or progress in Bacon's sense [2]) and we understand it as being different from revolution and innovation.

Let us now come back to our notion of Pulsation. We have seen that, in the informally specified notion of oscillation, we switch coherently between two paradigms. In our interpretation, the second paradigm, i.e.,

∃ Idea ∀ Problem Leads_to_a_solution(Idea,Problem)

represents the idea of Ackermann's function and the first paradigm, i.e.,

∀ Problem ∃ Idea Leads_to_a_solution(Idea,Problem).

represents particular primitive recursive functions from which Ackermann's function is constructed. In the definition of Ackermann's function we have seen that

$$f_{i+1}(0) = f_i(1).$$

Analogously, we shall state that the sequence of completing theories can be written as:

$$T_{i+1} = T_i + A_{i+1},$$

where A_{i+1} is an axiom or a set of axioms representing the conceptual switch that enables solving the problem unsolvable in T_i and solvable in T_{i+1}.

Let us stress the fact that by Pulsation we understand an infinite sequence of theories T_0, T_1, …, T_n, T_{n+1}, … with the just above mentioned property: It does not reduce to one particular step in this sequence. This means that pulsative systems are formalized progressively and potentially indefinitely.

We have seen above that Ackermann's function is also a model for symbiotic consideration of prevention and control.

Let us return therefore to the construction of Ackermann's function. We could see that, with respect to our requirement to obtain a non-primitive recursive function, f_0 must be defined in a way that guarantees the non-primitive recursion of the final composition of the constructed infinite sequence. Indeed, if f_0 were a constant, for instance 3 (which would mean that $f_0(n) = 3$ for all n), the resulting infinite composition would also be the constant 3. This means that, even though f_0 is the first function of this infinite construction, since it must be defined as a symbiotic part of the final composition, prevention and control factors must already be present in this function.

We can thus see that Ackermann's function provides in fact a model for the Pulsation that intends and guarantees symbiotic handling prevention and control already from the start. In other words, prevention and control are present already in T_0.

VI. PULSATION AND CARTESIAN INTUITIONISM

In the previous part, we have introduced the notions of symbiosis, systemic recursion and Pulsation.

We have seen that symbiosis is different from compositions that are not separation-sensitive. Usually, systems that are not separation-sensitive are considered as modular also in the case of interdependency. In modular interdependent systems, the parts, when separated, preserve their essential properties. This is not the case for symbiotic parts of a symbiotic system.

We have also seen that recursive systems are different from systems that allow linear-like, tree-like or network-like representation.

It is somewhat obvious that the paradigm of Pulsation is different from what is understood as a process of evolution that tends to preserve only strongest 'individuals'. A similar kind of evolution can be recognized in self-organized systems. Edward de Bono [3], one of recognized experts of practically exploitable creativity and innovation, has characterized such a self-organizing system by the fact that "an idea may be logical and even obvious in hindsight but invisible to logic" of externally organized systems.

Conversely, in externally organized systems, any idea which is logical in hindsight must be accessible to logic in the first place. This last assertion is true for the classical systems but not for our SRPS. Pulsative systems are externally organized by human creators and developers. However, in pulsative systems no conceptual switch can be considered as logical in hindsight. This follows from the fact that the axiom (or system of axioms) A_{i+1} that extends a theory T_i is logically independent from the previously constructed theories. In consequence, A_{i+1} cannot be logically explained in T_i.

Moreover, the main problems of Pulsation are construction of systems and development of completion-like procedures for these systems. The decision procedures are secondary and dependent on the developed construction and completion-like procedures. In Pulsation, all 'individuals' collaborate symbiotically towards one goal that is informally specified from the start.

This means that SRPS are complementary to non-recursive systems that are usually considered in science or business. In order to capture the essential difference between the paradigms implicitly present in standard science and the complementary SRPS, we call these paradigms Newtonian and Cartesian Intuitionism, respectively.

The main difference between Newtonian and Cartesian paradigms is easily perceptible from comments pronounced by Newton and Descartes themselves.

In a letter to Robert Hooke, Newton wrote: "If I have seen further (than you and Descartes) it is by standing upon the shoulders of Giants."

Newtonian science can be seen as established on logic of sequential 'observational' research. In a little more systemic way, we can thus describe the Newtonian way by a sequence of advancements built one upon the other from a 'beginning' until an 'end'. We say that this research is observational since, at each step of advancement, it does not require that the previous results are fully recreated. They are only observed externally and adopted as true. This is a model of what means "standing upon the shoulders of Giants."

Descartes wrote his first rule in the *Discourse on the Method of Rightly Conducting the Reason, and Seeking Truth in the Sciences* [4] in a following way: "The first was never to accept anything for true which I did not obviously know to be such; that is to say, carefully to avoid precipitancy and prejudice, and to comprise nothing more in my judgement than what was presented to my mind so clearly and distinctly as to exclude all ground of doubt."

This formulation could be looked upon as being similar to Newton's except that Newton expresses the utmost confidence in the 'giants' while Descartes wants to check, or rather re-create everything by himself before accepting a new knowledge. Indeed, Descartes always recreated all the knowledge useful to him when it had been previously obtained by someone else. This means that, if ever standing on 'giants' shoulders' took place, it had to be very carefully checked in order justify any extension of it.

Descartes justifies these possible extensions by stating that they have to lead to some obvious truth obtained by what he called 'intuition'. He describes what this 'intuition' is in his *Rules for the direction of the mind* (*Regulae ad directionem ingenii* [5]). When examining his definition from a recursive systemic point of view, we get hints that intuition, for Descartes, is a symbiotic, possibly recursive, composition. The process by which these hints are shown to be reasonable is complex and explained in detail in [8].

The same thing is expressed by Descartes in a little more complicated way by saying that "beginnings … can be persuaded well only by the knowledge of all the things that follow later; and that these things which follow cannot be understood well, if we do not remember all those that precede them." [4], p. 797. Note that our description of a Pulsation, in Section V. above, looks like an explicatory paraphrasing of Descartes' way of speech. From a more formal systemic point of view, we may state that the demarcation of a notion is not the initial stage, as it is the case in the Newtonian paradigm, but the final stage of its formation.

We thus see that Descartes' work, as we present it, contains a basis for SRPS systemic research. We introduce therefore Cartesian Intuitionism as a paradigm complementary to the Newtonian one. For us, Cartesian Intuitionism is nothing but a systemic science relevant to SRPS. In [9] we provide a more detailed comparison between Cartesian Intuitionism and Newtonian approach in the framework of Program Synthesis (PS). PS is a basic problem to be solved in the technological vision described in the next section.

VII. A PULSATIVE TECHNOLOGICAL VISION

In the previous parts, we have introduced the basic notions for a rough understanding of SRPS. Further work is necessary to provide a deep understanding of systemic emergence, i.e., the 'how' behind Pulsation. Symbiosis and recursion of parts of a system are reason why SRPS cannot be well understood externally. It is necessary to study them in the framework of a concrete creational referential system. In Computer Science, automation of recursive Programs Synthesis in Incomplete Domains via Inductive Theorem Proving (PSIDITP) already provides a usable experimental creational referential system. A formalization of this problem as well as a description of one particular approach built on systemic science of SRPS can be found in [9]. This approach is called Constructive Matching Methodology (*CMM*).

Let us make precise here what we call a methodology in a technological framework: Given a non-trivial goal, its solution relies on a fully formalized 'algorithmic' description of all problems that arise in achieving this goal. In this context, this special description is what is called a methodology (for the solutions of these problems). In other words, a methodology is a full 'know-how' for successfully achieving the given goal.

From the point of view of Pulsation presented here, it is interesting to note that the goal of *CMM* is to build a program synthesis system ('Idea') providing a 'Solution' to the problem of program construction in incomplete theories.

We thus globally work with the second paradigm. However, in our everyday research (which means to acquire fruitful experiences enabling to build relevant knowledge), we work locally with the first paradigm while keeping in mind the second paradigm. This means that we mentally oscillate between two paradigms. The second paradigm presents a global vision and the direction of the solution we seek and, to make this goal achievable, we perform our everyday work in the framework of the first paradigm following nevertheless the direction imposed by the second paradigm. It is important to note that we are still at the level 0 of pulsative development of *CMM*. In other words, we work on defining a powerful primitive recursive f_0 with respect to the overall goal of resulting non-primitive recursive SRPS for *CMM*. This means that level 0 has already required several decades of research and many useful results not known in PSIDITP were obtained so far. A full bibliography of these results can be found in [12]. We have to underline here that, obviously, an informal version of the Pulsation model was used from the start of our research. Recently only we formalized it enough to be presented in [1]. Our experimental implementation in [7] reflects this pulsative feature of our research.

In the long term, an expected success of the mentioned approach to PSIDITP provides fundamentals for a pulsative technological vision that may roughly be described by three contributions of PSIDITP. Indeed, PSIDITP seems to be a way how robots, in the future, will be able to

- formalize recursively unknown domains (e.g., in space research) handling perfectly control, rigor and evolutive improvement;
- perform experiments necessary for finding such suitable formalizations;
- program themselves autonomously with the help of the formalizations found.

Formalizing an unknown domain is a progressive exploration aimed at acquiring experiences – through experiments – that lead to facts enabling some progress in the formalization of this domain.

Of course, a successful achievement of this technological vision will require other tools than the ones presented in [9], [11]. New tools developed in Machine Learning, Big Data, Computational Creativity will certainly be also necessary. It is even quite possible that some of the necessary tools will appear in the future, born from pluridisciplinary cooperation and from yet unknown scientific fields developed in the course of research in PSIDITP.

VIII. PULSATION, SRPS AND ANCIENT'S SYSTEMS

The above presented construction and role of Ackermann's function as a model for infinite pulsation provides a very good sieve through which it is interesting to study or revise the systemic foundations of Ancient civilizations. Eternity, Timelessness and Progress are three essential themes upon which grow these ancient foundations (see [6], [2]). In modern interpretations these notions are still embedded within philosophical opinions. In our opinion, the pulsation model provides another point of view, a more mathematically oriented one.

We use a feature central to Ackermann's function, namely its representations by a specific infinite sequence of different and non-trivial functions, which constitutes a computable representation of *eternity*. Each of these functions plays an important role throughout the progressive growth of the sequence.

Timelessness might be represented by the fact that each f_i contains in a sense all the previous f_j (for $j < i$) and thus there is no obsolescence.

Finally, the fact that f_{i+1} represents a sort of a conceptual switch extending the potential power and action of f_i and of all previous f_j (for $j < i$) is a rigorous representation of *progress*, though perhaps a bit limited one.

This means that Ackermann's function seems to be a very good start for a more mathematically rigorous model of the three ancient symbiotic notions of Eternity, Timelessness and Progress.

This kind of thinking leads us to detect possible roots of our SRPS approach in ancient philosophy: It could quite be a particular "déjà-vu" of what has been understood in Ancient times as Universal Mathematics. In this case, it follows that a systematic study of the systemic links between all these ancient systems of thought and SRPS might bring many new ideas and technological visions for modern Science in general and for the development of secure dynamic evolutive systems in particular.

IX. CONCLUSION AND FUTURE WORK

Recognizing the symbiotic character of systems is vitally important, namely for security reasons and for preserving the essential properties of systems that are designed for showing no decline. We have shown that recursion is able to handle symbiotic information in systems and participates in their rigorous control and prevention. For real-world applications it seems therefore useful not only to recognize but also to develop symbiotic systems whenever conceptual switches are needed. With respect to the incompleteness of human knowledge, such a development must be strategically planned in order to avoid hindering future systems evolutions. We have therefore defined a non-trivial model, called Pulsation, for creation of symbiotic recursive systems. This model shows the essential features expected for a smooth evolution of human knowledge and for the design of ambitious real-world applications, namely

- possibility of infinite evolution (i.e., no decline)
- possible coherence of new results with previously developed systems, i.e., no rejection,
- rigorous security and prevention handling (i.e., no accidents).

In this way, the paper substantially completes and refines our definitions of Cartesian Intuitionism and Symbiotic Recursive Pulsative Systems introduced in our previous work.

We have mentioned in the paper the pragmatic and structural character of the notions developed. We have also mentioned a real-world application in which these notions are embodied. Their dynamics, i.e., their algorithmic descriptions started to be described in [13], where we provide a simple illustration a particular feature of Pulsation. Next, we plan to extend and generalize our experience acquired mostly in the process of the application mentioned in this paper.

Because of their rich potential, we foresee that Cartesian Intuitionism and Symbiotic Recursive Pulsative Systems will play an important role in the innovation process without needing to compete with standard Newtonian paradigms. However, once the complementary and non-competitive character of symbiotic systems is well understood, together with their rich potential, Cartesian Intuitionism and Symbiotic Recursive Pulsative Systems will certainly be highly exploited regardless of the particular ways of thinking they require.

ACKNOWLEDGMENT

We thank to Michèle Sebag and Yannis Manoussakis for a moral support.

REFERENCES

[1] M. Franova, Y. Kodratoff, "A Model of Pulsation for Evolutive Formalizing Incomplete Intelligent Systems," in L. van Moergestel, G. Goncalves, S. Kim, C. Leon, (eds), INTELLI 2017, The Sixth International Conference on Intelligent Systems and Applications, ISBN: 978-1-61208-576-0, 2017, pp. 1 - 6.

[2] F. Bacon, The Advancement of Learning, Kessinger Publishing Company, 1994.

[3] E. de Bono, Serious Creativity: Using the Power of Lateral Thinking to Create New Ideas, HarperCollins, 1992.

[4] R. Descartes, Philosophican works, Oeuvres philosophiques (3 vol.), Edition de F. Alquié. T. 1, Classiques Garnier, Bordas, 1988.

[5] R. Descartes, *Rules for the direction of the mind,* "Regulae ad directionem ingenii," in R. Descartes: Oeuvres complètes (11 vol.), Edition Adam et Tannery. T. 10, Vrin, Paris, pp. 359-469, 1996.

[6] I. Franco, Rites and beliefs of Eternity (Rites et croyances d'éternité), Pygmalion - Gérard Watelet, 1993.

[7] M. Franova, Precomas 0.3 User Guide, Rapport de Recherche No.524, L.R.I., Université de Paris-Sud, Orsay, France, October, 1989.

[8] M. Franova, Formal Creativity, Method and Use – Conception of Complex "Informatics" Systems and Epistemological Patent (Créativité Formelle : Méthode et Pratique - Conception des systèmes « informatiques » complexes et Brevet Épistémologique), Publibook, 2008.

[9] 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.

[10] M. Franova, D. Hutter, and Y. Kodratoff, "Algorithmic Conceptualization of Tools for Proving by Induction 'Unwinding' Theorems – A Case Study," Rap. de Rech. N° 1587, L.R.I., Université de Paris-Sud, France, Mai 2016.

[11] 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.

[12] M. Franova: List of publications (retrieved 2018.05.25) https://sites.google.com/site/martafranovacnrs/publications .

[13] M. Franova, Y. Kodratoff, "Cartesian Systemic Emergence - Tackling Underspecified Notions in Incomplete Domains," in. O. Chernavskaya, K. Miwa (eds.), Proc. of COGNITIVE 2018 : The Tenth International Conference on Advanced Cognitive Technologies and Applications, ISBN: 978-1-61208-609-5, pp. 1-6, 2018.

[14] D. Gries, The Science of Programming, Springer-Verlag, New York, 1981.

[15] S. C. Kleene, Introduction to Meta-Mathematics, North-Holland, 1980.

[16] H. Mantel, A Uniform Framework for the Formal Specification and Verification of Information Flow Security, PhD thesis, Universitty of Saarlandes, 2003.

[17] R. W. Ritchie, "Classes of predictably computable functions," Trans. Amer. Math. Soc. 106, pp. 139-173, 1963.

[18] E. W. Weisstein, Young Girl-Old Woman Illusion, From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/YoungGirl-OldWomanIllusion.html (retrieved 2018.05.25).

[19] A. Yasuhara, Recursive Function Theory and Logic, Academic Press, New York, 1971.

[20] Euclid, The Thirteen Books of the Elements, vol. I, II, III, Dover Publications Inc., Édition, 1956.