Cartesian Systemic Emergence and its Resonance Thinking Facet: Why and How?

Marta Franova, Yves Kodratoff

Cartesian Systemic Emergence and its Resonance Thinking Facet: Why and How?

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

***Abstract*— Cartesian Systemic Emergence (CSE) is concerned with strategic aspects relative to the conception of Symbiotic Recursive Pulsative Systems intended to solve real-world problems handling control and prevention in incomplete domains. This work is performed to prepare fundamentals for designing automated tools that help to perform this complex task. This paper recalls the fundamental notions of CSE and presents the most important features of one particular way of thinking present in CSE. We call it 'Resonance Thinking'. Resonance Thinking takes care of generating and handling experiments during CSE. The work presented is related to systems design, cognitive, and computation models of human creative reasoning mechanisms as well as to the ML approach called "Ultra-Strong Learning" for computer-assisted learning of CSE and RT.**

***Keywords*— *Cartesian Systemic Emergence; Symbiotic Recursive Pulsative Systems; Resonance Thinking; systems design; implementation of human reasoning mechanisms; Ultra-Strong Learning*.**

I. INTRODUCTION

This paper presents one of the symbiotic parts, called Resonance Thinking (RT) [1], of our computer systems design theory of particular complex systems in incomplete domains. This theory of computer systems design is called Cartesian Systemic Emergence (CSE) and has been introduced in [16]. The complex systems concerned by CSE are called Symbiotic Recursive Pulsative System (SRPS, introduced in Section III.F).

The goal of CSE is to *formalize strategic aspects* of the human creation of SRPS. The originality of CSE consists in representing these strategic aspects as a *deductive-like problem-solving system* and on focusing on a formalization of *axioms* of such a deductive-like problem-solving system. Moreover, this formalization is performed in order to prepare fundamentals for *designing automated tools* that help humans, or even that are able alone to perform this complex task (as it would be convenient, for instance, for robots in space). RT is a process in CSE taking care of a generation of relevant experiments useful for pointing out the specifications for the necessary parts of the system to be constructed. Superficially, it might, therefore, be seen as a way that a system architect (see [7]) proceeds to an analysis of the system requirements in order to *decide*, which parts are necessary for constructing the corresponding system. This paper will show that CSE considers systems for which a simple analysis of system requirements is not sufficient and for which there must be a large specific experimentation phase leading to an on-purpose *invention* of the most parts of the system. Francis Bacon, in [2], calls 'experiments of Light' the experiments that have to lead to a specification of system parts and thus are related to the invention of the system axioms in contrast to the 'experiments of Fruit' that concern experiments that explore and exploit the axioms already given. RT can, therefore, be viewed as an example of a rigorous method for performing 'experiments of Light' in the context of CSE.

The purpose of this paper is five-fold:

- present the fundamental notions necessary for understanding RT and CSE;
- describe particularities of RT taking place in CSE;
- illustrate this method by a toy example, which nevertheless deals with a problem that many innovative researchers may have to face;
- suggest an application of Ultra-Strong Learning to a design of an evolving process for assisted teaching/learning CSE and RT;
- mention the main problems and challenges addressed by CSE and RT to various fields of Cognitive Science.

Since the context of CSE is rather unusual in multidisciplinary system design, in Section II, we describe this context. In Section III, we present the notions necessary for understanding the topic of CSE. Section IV presents a short description of CSE. Section V is concerned with a presentation of RT. Finally, Section VI brings forward some related work.

II. THE CONTEXT

In this paper, we deal with two paradigms in designing a problem-solving system. The first paradigm can be represented by the formula

$$\forall \text{ Problem } \exists \text{ System solves(System,Problem).} \quad \text{(P1)}$$

The second one can be represented by the formula

$$\exists \text{ System } \forall \text{ Problem solves(System,Problem).} \quad \text{(P2)}$$

(P1) states that for any problem Pb_i one can build at least one system (or a module) S_i able to solve Pb_i. (P1) leads to a library of particular heuristics or methods relevant to solving individually each of Pb_i. (P1) is a *paradigm formula*, i.e., it has no truth value. It represents only a way to proceed when designing a system. Relying on (P1), one can, therefore, design a modular system S that is a modular composition of S_i that were previously built. Paradigm (P1) is useful when one of the main goals is to guarantee a simple maintenance of the resulting systems as well as a possibility of casual collaborations of the designers for each S_i [7] [28] [39]. Most system designing approaches are thus based on this paradigm. In this paper, the systems conceived via paradigm (P1) are called P1-systems. Similarly, the solutions to a problem conceived via (P1) are called P1-solutions. The same notation is used for P2-paradigm.

(P2) states that there exists at least one system that will solve all problems. The construction of a P2-system largely differs from that of a P1-system. The use of the P2-paradigm formula has to result in a single universal system S expressing the fact that this system represents a unique way in which all problems are solved. We can mention, for instance, the efforts of the approaches to Physics that tried to put in evidence one general theory of universe known as 'Theory of everything' [23]. The fact that, presently, there are two different results for macro and micro phenomena illustrates that a by cases analysis does not necessarily lead to a P2-system design. An illustration of a result of P2-paradigm is Peano's arithmetic for natural numbers (NAT). This example also illustrates that it is worthwhile to use P2-paradigm for the construction of systems that are not complete (in Gödel's sense [21]).

Since (P1) and (P2) are concerned with different goals, they are not competitive: each of them has its own particular 'competitive advantage' (see [38]) and these 'competitive advantages' are incomparable. Namely, as said above, P1-paradigm is useful when one of the main goals is to guarantee a *simple maintenance* of resulting systems. P2-paradigm is very useful for creating systems representing solutions for real-world problems that require handling *control* and *prevention* during the system design as well as in the resulting designed system. By control, we mean here the requirement to consider all the secondary effects of the evolution in the process of the construction of a particular system so that the constructed systems need *no future maintenance* (which guarantees, in fact, a non-obsolescence of the constructed systems), as it is, for instance, the case for Peano's axioms. By prevention, we mean here a careful anticipation of possible future practical needs, opening thus a way to a *smooth extension* of a previously, practically sufficient system. This can be illustrated by a smooth extension of NAT up to complex numbers. With respect to this particular competitive advantage of (P2), namely, handling control and prevention, there is a necessity to provide a formalization for what has been, so far, only an intuitive 'know-how' for designing P2-systems. This is the goal of CSE. CSE is a particular generalization of the experience that has been acquired while creating a reasonable solution for a real-world problem of implementing automatic construction of recursive programs specified by formal specifications in incomplete domains [14]. We shall refer to it as Program Synthesis (PS), for short.

A formal specification of a program (FSP) is a particular description of *what* the desired program P has to do. This description can, formally, be written as

$$\forall\, x\, \exists\, z\ \{\ IC(x) \Rightarrow IO(x,z)\ \}. \tag{1}$$

Here x is an input vector, z is an output vector, IC is an input condition on x and IO is an input-output relation. A resulting program P is then a description of *how* it has to be done. In other words, in PS, the goal is to design a system S such that, for each formal specification FSP, the system finds a corresponding recursive program P that executes FSP. Another way of describing this is also to say that, PS aims at providing a reasonable solution for the formula

$$\exists\, S\ \forall\, FSP\ \text{'transforms'}(S,FSP), \tag{2}$$

where 'transforms' means changing a 'what' (expressed by a formal specification FSP) into a recursive 'how'. It is thus clear that there is not only a close similarity between PS (expressed by the formula (2)) and P2-paradigm but also between FSP and P1-paradigm.

Our experience with the design of a PS-system required to express formally the thinking behind the development of our PS system. This motivated us to restrict our considerations of CSE to the design of systems that require a similar design process to that of our PS-system. In the next section, we present the fundamental notions that are necessary for understanding CSE as well as for a full specification of systems concerned (namely, SRPS introduced in Section III.F).

III. Fundamental Notions

The goal of CSE is to formalize strategic aspects of human creation of *informally* specified *symbiotic deductive-like problem-solving systems* in *incomplete domains* following our *pulsation* model. This formalization is performed in order to prepare fundamentals for *designing automated tools* that help humans, or even that are able alone to perform this complex task as already above stated. In this section, we recall five terms by which this goal is expressed and that will also be used in our presentation of Resonance Thinking, namely

- informal specification,
- incompleteness,
- symbiosis,
- deductive-like problem-solving systems, and
- pulsation.

The goal of CSE is to be considered in a P2-framework, i.e., CSE aims at a formalization that is a P2-system. Therefore, all these notions, that we need to define, are symbiotically interrelated. As a consequence, each of these fundamental notions cannot be clearly described without referring to the other fundamental notions. This is why, in order to introduce such complex descriptions, we will present, at first, a rough description of their meaning independently of their aim to represent the basis of a P2-system. Such a rough description can also be used in the context of modular P1-systems.

The symbiosis of parts of a system means that, if even only one of these parts is eliminated, not only the system collapses but also all the other symbiotic parts collapse as well. An informal specification of a system is a description of this system that is somewhat vague, i.e., it may be unclear what the words in this description mean exactly. Deductive-like problem-solving systems are systems that are defined exactly by their corresponding axiomatic system. Incomplete domains are domains that are insufficiently formalized in the sense that there might exist several different interpretations corresponding to the considered formalization of the domain. Pulsation is a model for a particular kind of systems' evolutive improvement.

These notions were present in our work in their informal form from the start of our research on PS. In order to achieve an explicit formulation of CSE, we had to propose here a more formalized form of these notions.

A. Informal specification

The ***informal specification*** of the kind of systems that have to be constructed is a description of each system by a sentence in which occur terms that are underspecified, i.e., they are not yet exactly defined. When considered out of a particular context, the goal expressed by an informal specification may seem impossible to attain. For instance, let us consider two examples of informal specifications for real-world problems that may, outside a particular context, seem impossible to achieve.

(g1) In Computer Science, let us consider the goal 'Automate the construction of recursive programs via inductive theorem proving'.

(g2) In Cognitive Science, consider the goal 'Construct a scientific model of the human brain that solves all the questions and problems related to the brain mental processes'.

Both these goals seem impossible if we rely on the usual scientific contexts and the usual meanings of the terms in which these specifications are expressed. The notions and techniques introduced in this paper show that considering these goals from a different point of view opens a possibility to reconsider them as reasonable and achievable long-term scientific goals.

Even though from a management point of view, there may seem to be an unsolvable problem due to the informal specifications in which occur terms that are not yet exactly defined, from a scientific innovation perspective, it is acceptable that these terms evolve during the system construction. In other words, depending on some constraints and opportunities that may arise during the construction of a system, the meaning of the terms used in the starting specification will evolve and thus, the final delimitation of terms makes a part of the solution. In other words, the initial ambiguity of terms occurring in a given informal specification is eliminated by the provided solution. The evolution of these informal terms, as well as of the design of the system will then bring also an exact specification of the context to be considered.

The evolution process of NAT shows that NAT have been used with a rigor even before their final exact axiomatization by Peano (PAD). Therefore, in order to introduce a difference between *rigor* and *exactness*, in the framework of CSE, the notion of informal specification needs to be completed by the notions of formalized and formal specification. In CSE, ***formalized specification*** is an intermediary stage on the way from an informal to a formal specification. It consists of a collection of basic not yet exact definitions and basic not yet exactly defined tools that seem plausibly pointing out a successful completion process. Some inventive steps may still be needed to complete these inexact but rigorous tools so that their use and evolution through suitable experiences leads to their final form as well as to the final form of the basic definitions. In CSE, ***formal specification*** then consists of a complete solution represented by the working system (be it in its axiomatic form or in its implemented version), both a methodology for the system functioning and the complete knowledge necessary to the system construction. These all are needed in order to be used in further evolutive improvement - if such an improvement is relevant. Note that the notion of formal specification is here different from the notion of formal specification of a program (noted FSP in Section II). FSP describes what a program has to do, while in the context of CSE, a formal specification is a complete solution to a problem-solving task.

B. Incompleteness and Practically Complete Systems

As far as the notion of *incompleteness* is concerned, from a practical point of view, we know that full reality is unknown. What we may know at a given time can be formalized by an incomplete system.

In order to illustrate the informal (thus ambiguous) 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 still 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 (or inexactly or not clearly) 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. This illustrates what we mean by an informal definition. This also means that the completion process of a definition (its emergence) needs to orientate 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 of objectives oriented towards a convenient solution of the informally specified problem. This means in practice that, in any design or completion process, the goal is to formulate experiments oriented towards the construction of relevant constraints for the intended objective as well as for the final delimitation of notions.

We know that even though PAD defining NAT is an incomplete system [21], we all use successfully and with a rigor PAD in our everyday life as well as in exact sciences. This means that a practically incomplete system can already be used by all if we all learn to 'stick' to one exact interpretation. The situation is similar to the use of different geometries in real-world considerations where the experts of EG, of hyperbolic or of elliptic geometry stick rigorously to their completion of the above described 'mutilation' of EG. This means that it is usually the practical aims that point out towards the completion of an incomplete system.

As it is pointed out in [19], p. 20, it is meaningful to work with an incomplete system. An incomplete system that is useful to exploit despite its theoretical incompleteness is called, in our work, a ***practically complete system***.

From a decision point of view, it is well-known that incompleteness constitutes a large drawback. Incompleteness, however, is not at all a drawback for the practical purpose of solving real-world problems that are asking for some kind of invention [20]. This is because, from a construction point of view, incompleteness brings freedom for technological ingeniosity, resulting in possible new technological inventions. This means that changing the decision context, whenever possible or desirable, to the constructive action context (on-purpose designed for handling incompleteness) is a way to push us to think of a model for a process of a practically useful and theoretically reasonable completion of incomplete systems. Thus, this model should represent a kind of *directed anticipation* of future extensions leading to an ideal, maybe never achievable, complete system. In other words, it is interesting, for an incomplete system, to think of a practically useful rigorous process of completion and of evolutive improvement that would lead, at least by intention, to a complete system. Of course, such an intention would need to be specified, in advance, by an informal specification as is, for instance, expressed by the goal of CSE. In Section III.E, we present a model for such a process.

Let us note that, since an informal specification of a system contains terms that are not yet exactly defined, a particular informal specification points out to a context that can be represented by an incomplete environment. CSE can then be seen not only as a construction process for a system in its informally specified initial environment but also as a fruitful strategy for a progressive, evolutive, completion of this environment.

C. *Symbiosis*

When we handle incompleteness and informal specifications in the design of a system, we need to be aware of a particular interdependence of the parts of the resulting system. This is where symbiosis moves into the systems design methodology.

By ***symbiosis*** we understand a vitally separation-sensitive composition of several parts. By *vital separation-sensitivity* of a composition, we mean that eliminating one of its parts leads to three possible penalties. It may be a complete destruction or a non-recoverable mutilation or the uselessness of the remaining parts. This means that the widely used divide and conquer strategy, as well as analysis and synthesis, are not at all suitable when creating and extending symbiotic systems. Symbiosis is therefore different from the synergy that is a mutually profitable composition of elements that are not destroyed nor mutilated by separation.

The notion of symbiosis has been inspired to us by Descartes notion of 'conceptual distinction' [10], §62, p. 214 (in French: 'distinction par la pensée' [9], p. 131) and the notion of symbiosis used, a half-century ago, to describe the failed attempt to separate algae and fungi in lichen. This attempt was, at that time, a failure, since both separated parts died. We do not know whether it is today possible to separate these two parts. However, as far as we could consult the internet, modern science understands symbiosis only as a *balance* between symbiotic parts that can only be achieved by working together. In other words, it seems to us that modern understanding is far away from our perception of the parts possibly 'dying' by their separation.

Let us consider the following pictures.

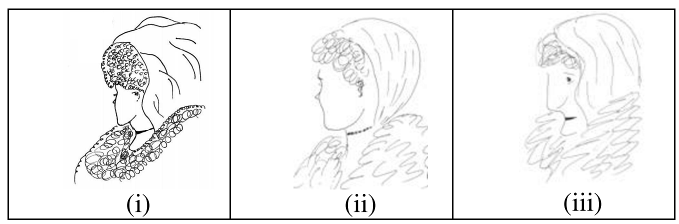

Figure 1. Example of pictorial symbiosis.

In Figure 1, a picture similar to (i) is known on the internet as 'young old woman illusion'. (i) can be seen as a symbiosis of (ii) and (iii). Indeed, if we remove from (i) the part that corresponds, for instance, to (ii), the part similar to the picture (iii), occurring previously in (i) before this step of removing, disappears as well. In other words, the result of removing one part in (i) is 'nothing'. This illustrates what we mean by 'destruction' in our definition of symbiosis.

Here, we need to point out that symbiotic parts do not necessarily need to overlap in the final symbiotic object. They may have a symbiotic, and maybe invisible, intersection that makes symbiotic their whole. From a systemic point of view, symbiosis of a system is embodied by the interdependence of all the notions and the parts of this system. For instance, a typical example of a symbiotic system is provided by PAD defining NAT. If we use the symbol ♦ for symbiotic compositions, a formal representation of the systemic definition of NAT reads:

$$\text{NAT} = 0 \blacklozenge \text{Suc} \blacklozenge \text{NAT},$$

where Suc is the successor function. If we eliminate one part from this system, for instance 0, we may no more speak of Suc nor of NAT as such.

D. Deductive-like Problem-solving Systems

This section shows that there is a close relation of the notions of informal specification and of incompleteness to *formal* and *deductive-like problem-solving systems* (DPSS) specified initially by informal specifications. To any deductive-like problem-solving system corresponds its underlying deductive-like theory, a part of which constitute semantic definitions, non-abstract axioms, and 'intuitive' rules, as it is, for instance, in Euclid's geometry the complete description of which contains not only axioms but also the set of semantic definitions and real-world influenced practical rules (for instance, knowing *a priori* to draw a straight line between two points) necessary for handling the knowledge implicitly containing all the knowledge related to the use of this deductive theory. With respect to the existence of an informal specification for a DPSS, in our work, we point out the following difference between a deductive and a formal system.

A ***formal system*** is an *abstract* system where the axioms are given *without* the definitions containing the semantic of the objects considered and without the semantic rules that allow performing operations in this formal system. Moreover, when a formal system is considered in Science, its consistency is considered in terms of non-existence of a proof for a formula A as well as for the negation of A in this system. In contrast to this, a deductive-like system has a real-world model semantics.

As far as deductive systems are concerned, the history of the evolution of the formal axiomatic system for NAT shows that this formal system has been developed only at the end of the nineteenth century. In the previous centuries, even though semantically influenced intuitive 'definitions', intuitive 'axioms', and intuitive 'rules' were used and were evolving, these evolving intuitive objects were handled by researchers in a rigorous and intellectually proper way [19] [24]. This evolution process could then be concluded by the formulation of the final formal system used today, namely PAD. By ***deductive system*** we thus understand a system developed with a concrete real-world application as a model. This means that, in contrast to formal systems, the consistency of deductive systems in their evolutive process of construction is proved by the existence of a concrete model. Therefore, a deductive system is in our work viewed as a result of the development of a relevant, possibly incomplete system of semantic axioms for a particular intended application interrelated with semantic definitions of objects that occur in these semantic axioms and semantic rules. In the final stage of the development, when exclusively the manipulation (and not construction or completion) purposes can be and are considered, a deductive system can be viewed as a formal system obtained by abstraction, however, its completeness is not viewed from a theoretical point of view but from the point of view of practical completeness.

E. Pulsation

Pulsation is a model for construction and evolutive improvement of incomplete, but practically complete systems that are concerned with the above-described factors of control and prevention (see Section II). In other words, pulsation provides a rigorous framework for the completion process of incomplete systems. This model relies on our particular handling of Ackermann's function. We shall recall now the features of its handling that will also be referred to later in the paper.

Let 'ack' be Ackermann's function defined, as in [40], by its standard definition, i.e.,

$$\text{ack}(0,n) = n+1 \tag{3}$$

$$\text{ack}(m+1,0) = \text{ack}(m,1) \tag{4}$$

$$\text{ack}(m+1,n+1) = \text{ack}(m,\text{ack}(m+1,n)). \tag{5}$$

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. In [18], it is shown how, for computing the value of ack(a,b), for given a and b, one can replace the standard definition of Ackermann's function by an on-purpose recursive macro, which consists of a finite sequence f_n of primitive recursive functions the "infinite limit" of which corresponds exactly to ack, i.e.,

$$\lim_{n\to\infty} f_n = \text{ack}. \tag{6}$$

This trick changes the non-primitive recursive computation of ack(a,b) for particular given values of a and b to primitive recursion and thus makes Ackermann's function suitable as a model for practical purposes of constructing systems that are not dealing with the computations on NAT but with solving real-world problems. In similarity to the infinite sequence, which is used in [18] to construct ack, the evolutive improvement (i.e., pulsation), relies on a construction of a potentially infinite sequence of systems that might, in an ideal world, be used to construct a global *'Ackermann's system'* that contains all of these systems. In our work, by pulsation we thus understand a progressive construction of a potentially infinite sequence $T_0, T_1, \ldots, T_n, T_{n+1}, \ldots$ such that

- T_0 is the initial informal specification,
- T_i, for $i > 0$, is an incomplete, but a practically complete deductive-like system,
- $T_i \subset T_{i+1}, T_i \neq T_{i+1}$ (for i = 0, 1, 2, …), and
- an infinite limit of this sequence represents an ideal, complete deductive system S.

We say here that T_{i+1} is a *practical completion* of T_i (for i = 0, 1, 2, …). We have

$$T_{i+1} = T_i \cup A_i, \qquad (7)$$

where A_i is the set of axioms corresponding to the extension of T_{i+1} from T_i.

The fourth requirement can be written as

$$\lim_{n\to\infty} T_n = S, \qquad (8)$$

where S is the ideal, complete Ackermann's system corresponding to the initially given informal specification.

The notion of Pulsation was inspired by the informal notion of 'gradual building of theories' presented in [19], p. 21, and the informal notion of 'pulsative system' presented in [12], p. 165. With respect to the constraints of handling control and prevention in the design of a DPSS as well as in a concrete DPSS itself, we have just developed and improved the formalization of these notions into our notion of Pulsation using Ackermann's function as a model. We have done so since this function models 'thinking of everything' (it is thus related to P2-paradigm and similar to the above mentioned "Theory of Everything" in Physics) as well as an effort that leads to achieving this goal.

Pulsation does not reduce to one particular step in the sequence T_0, T_1, …, T_n, $T_{n+1,}$ …. This means that pulsative systems are formalized progressively and potentially indefinitely. Thus, this new paradigm of the pulsative evolution of systems corresponds well to Bacon's understanding of Progress as a result of the work of several generations [27].

Pulsation has been introduced in [18]. Pulsation is a model that does not describe how the particular systems in the sequence T_0, T_1, …, T_n, $T_{n+1,}$ … are constructed. This is the role of Cartesian Systemic Emergence described below in Section IV.

F. *Putting it together*

Let us consider an example of a symbiotic system for which we put symbiosis together with Pulsation and P2-paradigm. Thus, let us consider the above informal specification (g2) from Cognitive Science. Now, we may ask whether a solution might not be something like

$$\lim_{n\to\infty} Brain_n = Brain, \qquad (9)$$

where

$$Brain_n = Left\_Brain_n \blacklozenge Right\_Brain_n \blacklozenge RNK_n \;\&$$
$$Left\_Brain_n = Brain_{n-1} \blacklozenge RNK1_n \;\&$$
$$Right\_Brain_n = Brain_{n-1} \blacklozenge RNK2_n$$

Here, RNK_n is a new knowledge related to symbiotic composition of left and right brains in the n-th pulsation step, $RNK1_n$ and $RNK2_n$ are relevant new knowledge extending, by the process of practical completion, the previous knowledge respectively about $Left\_Brain_{n-1}$ and $Right\_Brain_{n-1}$. This means that, instead of studying the brain as a synergy of two elements (left and right brain) [3], it might be interesting to explore the potential of this new symbiotic paradigm. Note also that (9) represents a recursive model of the brain, which explicitly indicates its evolutive character. This example also illustrates that, from a systemic point of view, symbiosis of a recursive system is embodied by the interdependence (explicit or implicit) of all notions and parts of this system.

Above, we have introduced the fundamental notions that contribute to understanding CSE and ST. These notions allow us to introduce, in multidisciplinary complex systems design, the notion of *Symbiotic Recursive Pulsative Systems* (SRPS) that are, by definition, systems that are implicitly or explicitly symbiotic, which are recursive either by systemic recursion or by the process of evolutive improvement via Pulsation, and that are pulsative, whenever the model of Pulsation (together with the notion of practical completeness) is used in their design. In order to point out that SRPS are developed via paradigm (P2), we may call them P2-SRPS.

We thus come to Cartesian Systemic Emergence and its facet 'Resonance Thinking' that takes care of generating and exploitation of some experiences in this process of CSE.

IV. Cartesian Systemic Emergence

The above-introduced vocabulary allows us now to say that the goal of CSE is to *formalize strategic aspects* of human creation of P2-SRPS in order to allow a collaboration in P2-SRPS-projects and to develop automated tools that help human in this task or even that perform this task of P2-SRPS-creation themselves. As presented in [16], the main features of CSE are as follows:

- It works with an informally specified goal.
- It handles incompleteness.
- Takes into account symbiosis and pulsation.
- Generates experiences.
- 'Oscillates' between the paradigms (P1) and (P2) in order to reach a solution described by (P2).

There are four fundamental symbiotic facets of CSE:

(a) Pulsative Thinking, i.e., taking care of security, control, and prevention.
(b) Metamorphic Thinking, i.e., taking care of resulting epistemological equivalence between P2-paradigm and particular CSE-handling P1-paradigm.
(c) Symbiotic Thinking, i.e., taking care of the construction of a symbiotic system.
(d) Resonance Thinking, i.e., taking care of generating and handling experiments.

In [1], we show that these four rules are inspired by Descartes' method [8] considered in symbiotic environments.

As we can realize while trying to give an *exact* description of the old-young lady picture given in Figure 1.i, a description of one part in a symbiotic composition (such as 'old lady' in Figure 1.iii) is not a simple task. Indeed, while considering Figure 1.i, an exact description of the old lady (as in Figure 1.iii) would imperatively require explicit references to the young lady (as in Figure 1.ii). Therefore, in this paper, we do not intend yet to provide a complete description of Resonance Thinking, since we need first to describe more in details Metamorphic Thinking (MT) and Symbiotic Thinking (ST).

The next section presents the main ideas necessary for a rough understanding of RT.

V. RESONANCE THINKING

Experiences represent an important part of each real-world complex system designing process. For P1-paradigm, with respect to the possibility of a heuristics library representing a set of partial solutions for different types of problems, these experiences are mostly clever combinations of the already acquired knowledge. However, by the requirement of a universal, 'unique', character of the solution for P2-paradigm, it is necessary to perform experiences that are relevant to the invention process of the knowledge that is not yet available and that is not a combination of the available knowledge. In other words, since CSE aims at a construction of a deductive-like problem-solving P2-system, the ultimate goal of RT is to generate experiences that suggest the essential symbiotic parts (understood as primitive notions) of the desired P2-system. As a hint, we could here recall that Francis Bacon, called 'experiments of Light' such experiences.

In Section II, we explained what is the difference between the use of (P2) and (P1). RT exploits the idea of 'oscillating' between (P2) and (P1) in order to come to a global P2-solution by studying particularly chosen P1-experiments. In order to be fruitful and justified, a switch from (P2) to (P1), i.e., an 'oscillation' step, generally has to rely on the above mentioned symbiotic part of CSE, namely MT. Roughly speaking, MT takes a care of a rigorous, epistemologically and pragmatically justified transformation of paradigm (P2) into the context of paradigm (P1). Such an epistemic justification relies on particular features of 'case analysis' used while developing a P2-system. As we have said above in Section II, not all kinds of 'case analysis' allow coming from P1-systems to a universal P2-system. Therefore, in our future work, we aim to explain in detail what makes our 'case analysis' reach the desired goal. However, we need to present ST first, since all the symbiotic parts of CSE have to be included in such an explanation.

In other words, MT provides a switch from (P2) to (P1) that is useful in order to generate experiments leading, within the P1-framework, to some hints and inspiration for solving finding a P2-solution. This means that the goal of RT in the construction of a P2-system is twofold:

- to hint at the parts of the future P2-system,
- to provide a guarantee that the suggested parts are symbiotically interrelated.

The hints and inspirations resulting from an achievement of the first goal represent temporary underspecified constraints that enlarge the already existing set of underspecified constraints.

We shall present RT and its basic notions with the help of a toy example used, in [16], for illustration of CSE. In comparison to examples provided by PS-framework, this example is simpler and could illustrate many other scientific fields than PS-research does. The example problem presented here concerns conveying a new original scientific knowledge in such a way that its essential content and creative potential are preserved by the next generations. This is not a trivial problem as already pointed out in the past [2] [8]. Our experience confirms that, for new knowledge that concerns the creation and a smooth extension of symbiotic recursive systems, this problem remains relevant until now.

A. *Specification of a toy example*

In this section we present the context of our example illustrating RT.

Let us suppose that René is a founder of a novel scientific theory with a high pulsative potential. Referring back to the founders' unhappy past experience, he (our 'René') needs to ask himself: "How to build some 'works' able to convey the full complexity of my new theory while simultaneously preventing a degradation of its pulsative potential?" In a more formal way, René must solve a problem informally specified as:

$$\begin{gathered}\exists \text{works } \forall \text{disciple conveys(René, works) \&} \\ \text{conveys(works,disciple)} \Rightarrow \\ \text{essential\_of(René) = essential\_of(disciple).}\end{gathered} \tag{10}$$

Note that this problem has the same logical structure as P2-paradigm. Specification (10) is an informal specification. As said above, this means that the notions that appear in (10) are not defined in a rigorous way. They are only specified in an informal way in terms of some non-formal criteria (i.e., a kind of underspecified constraints). This means that a solution 'works' for (10) has to emerge simultaneously with suitable formalizations (thus, the final definitions) of notions that occur in (10). In the following, we shall denote by D_t the set of (initially underspecified) sentences specifying 'to convey' and by D_e the set of (initially underspecified) sentences specifying 'essential_of'. These sets evolve in the process of CSE and Resonance Thinking towards a more rigorous final form. For the presentation simplicity, we do not involve such an evolution in our notation.

B. *Resonance Thinking in Action*

In order to solve (10), we perform a particular switch to a framework of experiments described by the formula

$$\forall \text{disciple}\ \exists \text{works conveys(René, works)}\ \& \\ \text{conveys(works,disciple)} \Rightarrow \\ \text{essential\_of(René)} = \text{essential\_of(disciple)}. \qquad (11)$$

This formula represents P1-paradigm. Above, we have explained that there is a difference between the use of (P2) and (P1). This obviously applies to their instances (10) and (11). Above we have explained also that the goal of RT in a construction of P2-system is twofold, namely to hint at the parts of the future P2-system and to provide a guarantee that the suggested parts are symbiotically interrelated.

As far as the first goal is concerned, in order to generate such inspiring experiments hinting the parts of the future system, while considering (11), from the set of all disciples, we chose a finite number of disciples $d_0, d_1, \ldots, d_n$ that seem highly different from each other so that each of them seems *a priori* to need a different 'works'. We shall call *representatives* these disciples. In other words, our experience shows us that various challenging experiments are needed to obtain some inspiration contributing to a solution of (10) in the framework of the P2-paradigm. Note that we are working here in the context of real-world situations (the semantics of which is well-known) where we consider the representatives for which it is meaningful to suppose that some solutions can or should be found in the framework of the P1-paradigm. This is related to the notion of practical completeness of the resulting system. In other words, representatives describe situations that are difficult but not *a priori* unsolvable ones. Note also that we order these disciples in a numbered sequence just for the presentation purposes. This will be useful when describing recursive procedures that handle this finite set of disciples.

Recall that the two operators 'conveys' and 'essential_of' are here informally specified only by some set of sentences that represent informal descriptions (i.e., underspecified constraints) relative to these notions. Thus, we shall replace these notions by their informal descriptions. Above, we have denoted by D_t the set of sentences specifying 'to convey' and by D_e the set of sentences specifying 'essence_of'. Therefore, (11) writes as

$$\forall \text{disciple}\ \exists \text{works}\ \{\ D_t(\text{René, works})\ \& \\ D_t(\text{works,disciple}) \Rightarrow D_e(\text{René}) = D_e(\text{disciple})\ \}. \qquad (12)$$

Let us consider (12) for each particular d_i, i.e.,

$$\exists \text{works}\ \{\ D_t(\text{René, works}) \\ \&\ D_t(\text{works}, d_i) \Rightarrow D_e(\text{René}) = D_e(d_i)\ \}. \qquad (13)$$

For a moment, let us suppose that a solution for (13) is found for each d_i while, during this 'search', oscillating between paradigms (P1) and (P2). In this case, the oscillation means that while, in (13), we are working in the context of (P1), we seek for solutions that are not the results of clever heuristics but are the results of trying to capture 'parts' of a general method that might be a basis for a P2-system. This means that we 'switch' mentally from (P1) back to (P2). In other words, while seeking a P1-solution we, in fact, aim at a P2-solution. Practically, this manifest by the fact that one is aware of the danger that comes from the attraction of clever solutions. Clever P1-solutions are usually a barrier to a discovery or invention of a unified way of solving problems, which is aimed at by (P2). The actually obtained (almost P2-like but in reality a) P1-solution consists of a concrete value w_i for 'works' and of less informal descriptors $D_{t,i}$ and $D_{e,i}$. We shall note $Sol_i = \{\ w_i, D_{t,i}, D_{e,i}\ \}$. Due to a careful oscillation between paradigms (P1) and (P2), the descriptors $D_{t,i}$, $D_{e,i}$ and w_i refine 'works' and the operators 'to convey' and 'essential_of' in (10). These resulting refinements have to 'resonate' with the framework of paradigm (P2). By their resonating we mean that, during the experimentation process, we need to feel that they might, probably after some 'judicious adaptations', be applied also to other instances of 'disciple'. In other words, the parts suggested by d_i have to be symbiotically compatible with the parts suggested by d_j (for i, j $i \neq j$). Therefore, while the first step of RT (taking into the account the first goal of RT) lies in a careful choice of representatives leading thus to specific experiments, the second step of RT takes care of the second goal, namely it generates experiences that have to provide a guarantee that the suggested parts of the future system are symbiotically interrelated. These steps are interrelated in the sense that when a symbiotic interrelationship is incompatible among some parts suggested by d_i and d_j (for i, j $i \neq j$), new representatives are chosen and the consideration of the failure representative d_f (i.e., a representative suggesting incompatible parts) is postponed until more experience allows to suggest another solution Sol_f for d_f.

Let us proceed now to the second goal of RT, i.e., providing a guarantee that the suggested parts are symbiotically interrelated. It is important to note here that RT relies heavily on 'resonance' as defined by: "a quality that makes something personally meaningful or important to someone" (e.g., as in Merriam-Webster Dictionary). The second step of RT thus involves the ability to create and explore personally meaningful or important relations in the process of generating and handling experiments.

We have seen that the first goal is approached via a choice of relevant representatives, i.e., a *choice* of relevant experiments. The second goal is approached via a particular process of *generating* and *handling* complementary experiments. We are going to describe it in the framework of René's example.

At this stage, we suppose that (13) for d_0 is already solved. Sol_0 represents a 'temporary' solution for d_0. By 'temporary' we mean that this solution will still have to be approved or modified by RT. Procedurally, the part of generating experiences of RT is based on two procedures for which we cannot yet provide a detailed description (thus, making explicit also 'handling experiments' part of RT), as they rely also on other symbiotic facets of CSE not introduced yet (namely, MT and ST mentioned above). We shall, therefore, concentrate on explaining the role of these procedures. The first procedure will be called *topological* *symbiosis* (noted *ts*) and it is also a primitive operation for the second procedure. The second procedure is called *complementary topological symbiosis* (noted *cts*). Both of these procedures require creativity in developing symbiotic systems. They are therefore to be handled, for the time being, by a creative human person. The following description of the role of *ts* and *cts* illustrate some of the challenges that *ts* and *cts* have to tackle.

1) *On symbiosis in Resonance Thinking*

We need to point out here two particular features of *ts*. The first one concerns the character of possible "mutilations" performed by *ts* and the second one concerns its goal.

In Section III.C, we have presented an example of a pictorial symbiosis of two different women. One woman is young, the other is old. The resulting symbiosis is a face that can be seen simultaneously as a young and an old woman. The original two pictures of women have to be 'mutilated' so that the resulting symbiotic picture is a convincing illusion. For instance, an eye of the old woman and an ear of the young woman overlap in the symbiotic picture. As for the opposite ages of the women on the initial pictures, they are 'merged', since the symbiotic picture is at the same 'old' and 'young'.

A systemic symbiosis manifests itself not so much as 'merging' contradictory facets of the considered system (as 'merging' two opposites, namely young and old in the above mentioned pictorial symbiosis), but as constructing an emergent vitally separation-sensitive interdependence (i.e., symbiosis) of suggested parts of the system. Thus, while the first goal of *ts* refers to a seemingly useless 'mutilation' of suggested parts of the system, the second goal of *ts* expresses the importance of such a mutilation in a search of relevant vitally separation-sensitive interdependence (i.e., symbiosis) of suggested parts of the system. Note that these goals are feasible thanks to the fact that the suggested parts are informally specified, thus 'temporary' and evolving, till the end of the process of the system construction. By 'temporary' we mean that these parts will still have to be approved or modified by CSE and RT.

2) *On generating experiments in Resonance Thinking*

We are going to describe *ts* and *cts* in the framework of René's example. At this stage, we suppose that (13) for d_0 is already solved. This provides the solution Sol_0 for d_0. Sol_0 represents a 'temporary' solution for d_0. Similarly, for other disciples d_1, …, d_n, we will obtain Sol_1, Sol_2, and so on. We assume here that the solutions are obtained in a particular 'linear' way, one after another. This 'linear' way looks as follows.

Once Sol_0 is constructed, a 'temporary' solution Sol_1 for d_1 is constructed ('temporary' in the same way as Sol_0 is a 'temporary' solution for d_0, i.e., they will have to be still approved or modified by CSE and RT). Note that both these constructions may lead to new experiences and thus, *they may modify the initial environment* by refining the informal notions of our definition (10) of our problem. For the sake of simplicity, we do not describe explicitly below this evolution of the environment, though we take it into account by calling it a 'feedback' when we use it.

Now, let us suppose that we have solved the problem for d_1. Before starting solving the problem for the next one, we try to take into account the informal notions present in (10). This try amounts to an attempt to 'merge' the solutions Sol_0 and Sol_1 using topological symbiosis *ts*, i.e., we try to achieve their symbiotic composition that resonates (as explained above) with the informal specification (10). We shall denote this process by $ts(Sol_0,Sol_1)$.

If solving $ts(S_0,S_1)$ fails, i.e., we cannot find relevant refinements, we keep in mind the feedback obtained while constructing Sol_0 and Sol_1, as well as the failure reasons of $ts(S_0,S_1)$. This failed step will have to be redone later while relying on some inspirations that may arise while finding the solutions for the next disciples. If this process fails, the problem will have to be considered as a challenge for one of the next pulsation steps.

If the process $ts(Sol_0,Sol_1)$ succeeds, both solutions are temporarily approved. Then, keeping in mind all the feedback obtained, a solution of (13) for d_2 is constructed. One might suppose that this process may continue linearly as suggested by its beginning, as we just have seen. However, recall that we work in an environment that requires control and prevention. This means that generating complementary experiments for the topological symbiosis of solutions constructed is necessary. We call *complementary topological symbiosis* (noted *cts*) this procedure for generating new experiments.

Roughly speaking, *cts* is a particular generation process (defined with the help of *ts*) for creating experiments. The goal of these complementary experiments is to provide inspirations for further refinement of underspecified notions and constraints. Similarly to the computation of ack (see [17]), in the process of generating experiments (via *ts*) for Sol_m and Sol_n, i.e., while 'computing' $cts(Sol_m,Sol_n)$, the operation $ts(Sol_i,Sol_j)$ for other solutions Sol_i and Sol_j has to be performed several times.

Let us denote by $ts_1(Sol_i,Sol_j)$ the solution of the first computation, by $ts_2(Sol_i,Sol_j)$ for the second computation, and so on. It is important to point out that $ts_p(Sol_i,Sol_j)$ and $ts_q(Sol_i,Sol_j)$ in this sequence of computations may carry two different feedbacks. Indeed, each inner step of *cts* (i.e., evaluating $cts(Sol_m,Sol_n)$), may bring new refinements, constraints as well as it may point out to missing knowledge or second-order notions and procedures. The procedures *ts* and *cts* have to ensure that not only reasonable and achievable solutions are obtained but that a possibility of future evolutions are guaranteed while properly handling prevention and control.

We do not present here an algorithm for *cts* since an automated execution of *cts* is not yet solved. However, based on our large experience, already now we may express that an important requirement of this procedure is that, for a computation of $cts(Sol_m,Sol_n)$, it must contain considering $ts(Sol_m,Sol_n)$ in randomly generated environments provided by the computation of $cts(Sol_p,Sol_q)$, where ($p < m$ and $q < n$) or ($p = m$ and $q < n$) or ($p < n$ and $q = m$) and $ts(Sol_p,Sol_q)$

has to be performed several times for some p and q. In other words, *cts* is an environments generation process that has to be inspired by the computation trace of Ackermann's function [17]. Recall that, in RT, *ts* and *cts* are, in our case, presently performed by a human mind. This means that human mind can rely on relevant creativity in order to decrease the number of repetitions. In consequence, even though *ts* and *cts* are not simple, CSE and RT are not overwhelming tasks for human performers. However, they may be overwhelming for a human observer even in this simplified form.

VI. DISCUSSION/RELATED WORK

Usually, in the design of systems, there exists a clear distinction in the roles of a system architect/analyst, of a system designer and of system integrator (see [7]). This is not the case in the design of SRPS systems. Namely, these particular systems design requires 'one mind' for performing these tasks simultaneously (i.e., symbiotically). Moreover, collaboration on the design of SRPS systems is a scientific work that can itself be characterized as a complex emergent system [41]. In Section VI.A, we illustrate some negative consequences of an attempt to simplify and replace a symbiotic process by a process that should simulate a synergic collaboration. In Section VI.B, we describe our study of an interesting relationship between mental processes of Resonance Thinking and the so-called Ultra Strong Learning. Section VI.C brings some insights on a possible influence of CSE on some research topics in the field of Cognitive Science. Sections VI.D and VI.E present a comparison of our research, General System Theory and Multidisciplinary Systems Design, respectively.

A. On Simplification and Delegation

As said above, CSE and RT are concerned with the human creation of symbiotic systems. One may, therefore, wonder whether this process of creation cannot be described in a simple way, so that the usual process of delegation and synergic collaboration might be used. That such an initiative is not at all reasonable in the case of symbiotic objects may easily be illustrated by a study of the creation of Figure 1.i. Namely, it is not difficult to foresee problems if we charge several skilled painters that are not familiar with this picture to come out with a solution for the problem: Create a picture that makes some people to see exclusively a young woman, some people exclusively an old woman and some people both the women. Of course, Figure 1.i. is a convenient solution in this case. While this illustration is short, it does not rely on a scientific study. Such scientific study should justify the hypothesis that

- the process of creation of practically complete symbiotic systems cannot be simplified, and
- the usual synergic delegation cannot be used.

In order to give an illustration that is, in our opinion, scientifically admissible, let us consider, side by side,

- the above mentioned standard definition of Ackermann's function; we have used the name ack for it, and
- an unusual definition of Ackermann's function; we shall use the name ak for it.

Thus, we have:

$$\mathrm{ack}(0,n) = n+1 \tag{14}$$

$$\mathrm{ack}(m+1,0) = \mathrm{ack}(m,1) \tag{15}$$

$$\mathrm{ack}(m+1,n+1) = \mathrm{ack}(m,\mathrm{ack}(m+1,n)) \tag{16}$$

and

$$\mathrm{ak}(x,0) = \mathrm{sf}(x) \tag{17}$$

$$\mathrm{ak}(x,y+1) = \mathrm{ak}(x,y) + \mathrm{sf3}(x,y), \tag{18}$$

where sf is defined by

$$\mathrm{sf}(0) = 1 \tag{19}$$

$$\mathrm{sf}(a1+1) = \mathrm{sf}(a1) + \mathrm{sf1}(a1). \tag{20}$$

Here, sf1 is defined by

$$\mathrm{sf1}(0) = 1 \tag{21}$$

$$\mathrm{sf1}(b+1) = \mathrm{sf2}(b,\mathrm{sf}(b) + \mathrm{sf1}(b)) \tag{22}$$

and, sf2 is defined by

$$\mathrm{sf2}(0,y) = 1 \tag{23}$$

$$\mathrm{sf2}(a+1,0) = 1 + \mathrm{sf2}(a,1) \tag{24}$$

$$\mathrm{sf2}(a+1,b+1) = \mathrm{sf2}(a+1,b) + \mathrm{sf2}(a,b+\mathrm{sf2}(a+1,b)) - 1 \tag{25}$$

Finally, sf3 is defined by

$$\mathrm{sf3}(0,y) = 1 \tag{26}$$

$$\mathrm{sf3}(a+1,y) = \mathrm{sf2}(a,\mathrm{ak}(a+1,y)) \tag{27}$$

We have shown, in [13], in a constructive way, that ak is computationally equivalent to ack, i.e., it implements the program ack, even though it does so in a different form. Thus, both ack and ak are non-primitive recursive programs. While ack is defined recursively with respect to the first argument (and by-cases with respect to the second argument), ak is defined recursively exclusively with respect to the second argument. We shall call ack-form the definition of ack in terms of (14), (15) and (16). We shall call ak-form the definition of ak in terms of (17) and (18). Similarly, sf-form stands for the definition of sf by (19) and (20), sf1-form stands for (21) and (22), sf2-form stands for (23), (24) and (25), and finally, sf3-form stands for (26) and (27).

Pragmatically speaking, we have said, in [18], that Ackermann's function can be seen as 'thinking of everything' for a given problem. While some may argue that this is impossible because of the incompleteness of reality, our model of pulsation (see Section III.E) illustrates that this can reasonably be done if we accept to work in the framework of a potentially infinite sequence of practically complete systems.

Let us consider ack-form and ak-form from the management point of view of simplification and delegation. If we consider a name for a program as a person charged to perform the task of this program, the first and second argument as the right hand and left hand, respectively, we may notice that

- ack uses both hands and executes the task alone;
- ak uses just left hand, delegates completely to sf the computation of (17) and, as for the computation (18), ak together with sf3 work on it.

From the point of view of standard management, we may note that ak uses only the left hand – being thus able to accept other tasks of company employing him – and that he also delegates a full case (17) to sf. Therefore, his move – namely, to delegate fully one task and to collaborate with sf3 on the second case – are highly appreciated. Using the standard criteria of management ak is considered as a better manager. We may say even that ak seems much clever than ack since he has only two lines to describe his work, whereas for ack we count three lines. Moreover, ak uses a description of his work (i.e., ak-form) that looks very 'nice' from the perspective of the form of primitive recursive functions. Let us recall that primitive recursive functions are a composition of a finite set of functions.

Before continuing further, let us recall that it is not unusual that we feel at ease with forms that look familiar at first sight (since they look 'nice') and then we do not look too much at details [26]. We shall not skip the details in this case, and we shall go deep inside.

First, let us denote by BC_{ak} the set of members involved while solving the base step of ak-form, i.e., in (17). We have BC_{ak} = {sf, sf1, sf2}. In this set, sf is, from the management point of view, on the same level as sf1, since sf, in (20), collaborates with sf1 and sf1, in (22), collaborates with sf. Then, we have that sf1 is on a higher level that sf2, since sf1 partly delegates, in (22), to sf2. We can note that sf-form and sf1-form look 'nice' from the perspective of the form of primitive recursive functions. sf2 works alone with the work that is charged to him by his superiors.

Let us denote GC_{ak} the set of members involved while solving the general step of ak-form, i.e., in (18). We have GC_{ak} = {ak, sf3, sf2}. We can also note that ak collaborates with sf3, see (18). Finally, sf3's work is not recursive, in (27), he calls ak for help and then charges sf2 to do the final work. Since sf2 occurs in both sets BC_{ak} and GC_{ak}, since it is given a work by his hierarchic superiors, i.e., it is a simple auxiliary, it might logically seem that his work is very insignificant. However, if we compare sf2-form and the original ack-form (i.e., (14), (15) and (16)), we can see that sf2-form is as 'difficult' to appreciate - from the simplicity and 'niceness' point of view – as ack-form. In the language of mathematics, sf2 seems a non-primitive recursive program as it is the case for ack. Since there is a computational equivalence of ack and ak [13], sf2 must be the program that guarantees this non-primitive recursive character of ak. While we have shown [15] that the non-primitive recursion character of ack can be proved in a simple constructive way, in order to prove the non-primitive recursive character of sf2, one needs to return to [40] to seek inspiration for a proof by contradiction. Indeed, a proof by contradiction is usually presented to prove the non-recursive character of ack. And then, since ak uses sf2 in its computation, ak is obviously a non-primitive recursive program, since a primitive recursive program is, by definition of primitive recursion, a composition of a finite set of primitive recursive programs. Finally, while ack is computed in the base step, i.e., in (14), by a primitive recursive program n+1, ak computes its base step relying on non-primitive recursion of sf2.

In other words, from the point of view of simplification and delegation the program ak, with all his necessary auxiliaries sf, sf1, sf2 and sf3, gives an illusion of simplification and delegation while decreasing the computational efficiency of ack. In other words, even though ack works alone, he is more efficient than ak with his collaborators and auxiliaries.

Our illustration in this section shows that a synergic collaboration is unsuitable for the projects of the creation of SRPS. These projects require 'symbiotic management' and 'symbiotic collaborations'. These topics will be addressed in our future work.

B. Topological Symbiosis and IML

In the process of RT, performing topological symbiosis gives a rise to several problems that are also met in the field of Inductive Machine Learning (IML), as illustrated below.

In this research field, it is interesting to make the difference between several 'levels' of learning and Michie [32] provided three criteria for the evaluation of a degree of 'value' for Machine Learning (ML) results. He classified them as weak, strong and ultra-strong criteria. For him, the weak criterion identifies the classical case where the machine learner produces improved predictive performance with increasing amounts of data. The importance of this 'weak' ML is illustrated by the large success of what is nowadays called "Artificial Intelligence."

The two other criteria may nevertheless be the root for even more powerful techniques of learning.

'Strong' ML generates also some kind of symbolic explanations enabling the human persons receiving the results of a strong ML system to understand the why of these results provided the machine. This differentiates two subfields of ML, namely one based on numerical computations and one based on symbolic lines of reasoning.

Finally, Michie's Ultra-Strong Learning (U-SL) implies the existence of a kind of collaboration (not necessarily a symbiotic one) taking place in between a machine and a human in a way such that, at first, the ML system teaches some valid information to his human user, as may do IML applied to a specific training data set. In order to reach the ultra-strong level, human performance (on the same training data) has to be proven becoming more efficient than the one obtained by human studying the training data alone.

This last requirement asks for a computer system to perform three complementary abilities, as is shown in [33] [34].

The first one is to *generate pieces of programs* that are 'immediately' understandable to a human being. Since the manipulated programming language is Prolog, Muggleton's experiments have been carried on people who underwent at least two terms of Prolog teaching.

The second one (called *'Step A'* by Muggleton) is that the program, while it is running in order to answer a question, is able to generate new Prolog clauses. This ability is called "*predicate invention*" since each Prolog program is built with a concatenation of such predicates. The topic of predicate invention has been a basic problem for many Prolog specialists since the beginning of this research field. Fulfilling this condition amounts to achieve Michie's condition for a "strong machine learning" system. Muggleton's approach to this problem can be summarized as follows [33]: the system makes use of a controlled pattern matching of a higher order knowledge, provided in the form of meta-knowledge handled by a meta-interpreter. New knowledge is obtained by proving that a meta-goal is valid on a selected set of true examples. Note that this procedure is not submitted to our constraints of symbiosis and pulsation.

In our presentation, Muggleton's Step A. can be seen as a partial instance of what we call here *ts*, the role of which is to create new relationships induced from the data.

The third needed ability is stated below, in Muggleton's step B.

Muggleton's step B. Once new rules are found during Step A., [34] makes use of these rules in order to *select a set of significant examples*. This selection could work in a random way, generating a random mixture of examples illustrating both the old rules and the new generated one. In Muggleton's context, these examples are actually generated in such an order as to constitute the 'background knowledge' provided to a human learner. Thus, some selection among the possibly generated rules has to be done in order to be sure to obtain a 'significant' background knowledge.

In our presentation, Muggleton's step B. can be seen as a partial instance of what we call here *cts*. The goal of *cts*, similarly to step B., is to select a set of examples in order to complete or to enlarge the new knowledge initially generated by *ts*. However, in difference with step B., *cts* generates random examples because this provides a greater probability of generation of new (useful or missing) knowledge. This is coherent with our choice of a set of disciples for which solving (13) (see Section V) is rather difficult. We have mentioned above that this necessarily leads to a need for greater creativity and thus leads more efficiently to practical completeness of resulting system, here 'works' as in formula (10). At this stage, we already can acknowledge that Muggleton's steps A. and B. might be used as an inspirational model for programming the main procedural features of *ts* and *cts*.

As far as further steps in Muggleton's approach are concerned, his purpose is to get a set of rules such that their knowledge will improve a human's programming behavior. His success in this task, as explained in [34], proves that Muggleton's work is a success in implementing the very first ML program fulfilling the 'extra-strong learning requirements'.

Muggleton's successful trend of Machine Learning research opens us to some hope that teaching human-based creation of incompletely specified SRPS may benefit of the U-SL attitude, as the two following examples suggest.

Example 1, relative to symbiosis among the components of a system. In Section V.B, we have seen that symbiosis is better defined by its "vitally separation-sensitive interdependence" among the components, as illustrated by symbiotic Peano's primitive notions and axioms. As far as we know, teaching the recognition and handling of separation-sensitive interdependent systems, a skill necessary to creative programmers, does not exist yet. The research presented here provides a few clues of how it could be formalized. A tight collaboration with specialists in Cognitive Sciences should enable us to provide a large enough battery of symbiotic and non-symbiotic systems so that, mimicking US-L, we could unravel the deep features of systemic symbiosis, a necessary, if not sufficient, condition to safely handle creativity.

Example 2, relative to Oscillation. Oscillation has been introduced in Section V, where we underlined the difference between (P1) and (P2) problems. While exposing the "René's disciples" example, we used the switch between these two problems by replacing formula (10) by (11). Understanding the nature of a switch from a "$\forall\ \exists$" problem to a "$\exists\ \forall$" one is not easy, and its justification, as said above, requires epistemic considerations. We think that a strategy *à la* U-SL may constitute a tool favoring the understanding of the importance of the shift proposed here.

C. 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. Moreover, scientific creation, as a particular human invention [22], becomes a highly economically interesting topic when it can be turned into an implementable science. CSE does try to build an implementable theory of SRPS scientific creation. In consequence, the four fundamental facets of CSE bring several stimulating challenges to Cognitive Science (CS). In this paper we would like to mention modular model of the brain [3] [31], frame problem [3] and conceptual blending [11].

Bermudez's work, as cited above, seems to imply that CS is somewhat wary of non-modular processing. One of the reasons is that non-modular processing very quickly meets frame problem-like difficulties. We have seen that, during Resonance Thinking, the human brain is rather at ease with the identifications needed to handle the frame-problem. Why it is so? Is it because there is a particular kind of internal representation the human mind is able to construct? Alternately, is it because our mind includes mechanisms that are presently out of the scope of the current modular approach to our mind architecture [3]? Moreover, performing CSE includes a symbiosis of form, a meaning, a representation formalism, mechanisms and, importantly, includes also reaching a human agreement via conceptual coherence [35] and real-world exploitation. Does it mean that a modular approach to mind architecture should be revised? Could it be possible that some kind of symbiotic approach might be better suited even though it is more complex?

In [3], Bermúdez pointed out the influence of Computer Science on the development of CS Paradigms. CSE, 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 CS. One cause may be that symbiotic thinking is considered as not achievable in CS. For instance, John Medina claims in [31] 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 physically 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 possible misinterpretations of external observations. In particular, the observation of symbiotic modules in action may meet difficulties 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." ([25], 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.

In complement to considering symbiosis as a reasonable paradigm in CS, we have tried to find some concepts of CS that resonate with CSE-thinking. We have found some similarities between Resonance Thinking and Conceptual Blending (CB) as presented in [11]. On a high-level of abstraction, RT and CB seem similar, since they are both concerned with the construction of meaning and they both involve 'merging'. Of course, they also show some differences at this high-level because RT is consciously performed, while CB is considered as taking place outside consciousness and is not available to introspection (as in [11], p. 33). We believe that this unconscious feature of CB disappears if people work in domains where rigor, justification and reproducibility of results are essential. Incidentally, let us point out that Fauconnier and Turner's illustrations, in [11], do not fulfill these stipulations.

At a lower-level, RT seems to us more complex than CB. Let us mention several features of RT that contrast CB.

- CB is highly nondeterministic while, in RT, the solution is specified in advance, even-though informally. Thus, RT performs what could be called a 'goal-oriented symbiosis'. While handling the generated experiments, RT focuses on what resonates as contributing to a universal solution, as in René's example, 'works' in (10).
- RT involves solving underspecified constraints due to the presence of incompleteness and an informal specification.
- RT not only handles a given data input (experiments) but it also generates complementary data (temporary facts, feedback, new experiments).
- CB is performed on mental spaces, i.e., small conceptual pockets constructed for purposes of local understanding and action ([11], p. 40). In RT, there are no small conceptual packets since global understanding is required even in considerations that may seem local.
- CB usually works with two mental spaces. RT, via topological symbiosis, works with three inputs (two experiments and one goal) and the solutions obtained are temporary until other experiments confirm the output.
- Fauconnier and Turner [11], in relation to CB, claim that researchers are unaware of how they are thinking. RT is a description of our way of thinking relevant to the creation of SRPS.
- In the case of CB, the effects of some unconscious imaginative work are captured by consciousness, but the operations that produce it are not ([11], p. 58). As said above, RT (and CSE) is a description of our way of thinking that is relevant to SRPS creation. This means that we are consciously aware of the informal specifications of the operations performed by our mind.

It follows that CSE might well be part of a challenge for CS. This will be achieved by developing CS 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 CS will lead soon or later to a fruitful empowerment of CS. We hope that the models presented in the present paper might be of help in such a difficult task.

We are aware that our description of the cognitive tasks involved in CSE does not provide a clear idea of whether it is possible (or reasonable) to find a way to break down the cognitive tasks that are performed into more determinate tasks. We describe what humans do or what they have to do without specifying how these tasks are performed by our brain. We thus believe that research on these topics in the field of CSE in particular and its comparison with scientific creativity in general (i.e., a comparison with scientists' creative thinking in several scientific domains) might bring new conceptual and procedural switches not only in Computer and Cognitive Sciences, but also in other human activities.

D. Comparison with works on General System Theory

The next type of related work concerns complex systems modeling [4] [29] [30]. This falls to the domain of General System Theory (GST), where by a General System is understood "the representation of an active phenomenon perceived as identifiable by its projects in an active environment, in which it functions and it transforms teleologically" [30], p. 40. Similarly to our use of paradigm (P2), GST is conceived in the logics of an open unique global system. However, while GST responds actively to the problem of modeling and *observing* real-world phenomena, CSE builds a unified 'theory' of human *creation* of particular complex systems. It might, therefore, be seen as a kind of meta-theory of General System Theory used not in its standard observation mode but in its new 'creation' mode. Moreover, while, in GST, the conjunctive logic (i.e., conjunctive modular composition) is applied to already existing entities (i.e., the objects that exist already in their independent form; see [30], p. 33-41), in our work, symbiotic composition concerns 'objects' that start to exist only once the process of symbiotic composition is achieved.

E. Comparison with works on Multidisciplinary Systems Design

The last type of related work considered here concerns the systems design in general. It is thus somewhat related to the field called System Engineering (SEng) and Whole Systems Design, even though these fields do not use the same language and representations and the latter studies social and economical systems oriented towards sustainable solutions rather than developing problem-solving systems in general. The careful study of [7] [28], and [39] shows that SEng, even for complex systems, relies on modular compositions. However, even a complex communication of system modules, or their synergy (as in Whole Systems Design [5] [6]) is not a symbiosis. In order to have a symbiotic composition of the parts of the system it is necessary that these parts are defined in terms of the other parts of the system or even in terms of the system. Moreover these approaches work with specifications that are not informal and they do not consider prevention and control in the sense understood in our work. Therefore, our work on CSE and ST differs from these approaches to systems design. Note that CSE does not improve these approaches but it *enlarges* the class of possible approaches to Multidisciplinary Systems Design. Each of the mentioned approaches, by their competitive advantages, plays an important role for the progress in the field of Multidisciplinary Systems Design.

VII. CONCLUSION

The goal of Cartesian Systemic Emergence is to develop an implementable systems design theory for Symbiotic Recursive Pulsative Systems. In this paper, we have introduced one of its symbiotic features, namely Resonance Thinking. RT takes care of generating and handling experiments during the creation process of symbiotic systems specified, at the start, by an informal specification. RT is very complex, since it has to deal with the requirements of control and prevention, as well as with the process of 'shrinking' the incompleteness in accordance with the pulsation model.

Presently, our goal is not to apprehend all the conscious details of the operations performed by RT and CSE. Our present goal is to specify what enters into the 'game' of RT (and CSE) and what the 'winning strategies' are in order to conceive all the rules of 'the full game' of CSE. In other words, presently, we aim to develop a 'prosthesis' that can be implemented and used during CSE. We are convinced that apprehending human operations first by relevant informal specifications is halfway to a reasonable implementable solution. Thus, we believe that, even in its presently incomplete version, CSE brings forward thinking mechanisms that are essential for exploration, creation of possibilities, anticipation, resonance, blending, on-purpose creating of informally specified tools, invention, discovery, and so on.

Finally, recall that it is largely accepted that inspiration seems to take place anytime, such as while walking (e.g., Poincaré's case [36]), showering or during a pause playing the violin (e.g., Einstein's case). It is usually also accepted that some sort of unconscious incubation precedes this inspiration. Since we differentiate 'unconscious' and 'non-verbal', such an incubation does not take place during RT. Furthermore, contrary to Popper's opinion [37] that "there is no such a thing as a logical method of having new ideas, or a logical reconstruction of this process," RT is a systemic method for generating new and relevant ideas. Of course, its 'logical reconstruction' is not trivial, as is illustrated by this paper. CSE, with its four symbiotic facets, nevertheless seems to be a good start for a 'Cartesian reconstruction' of this process.

REFERENCES

[1] Y. Kodratoff and M. Franova, Resonance Thinking and Inductive Machine Learning, in S. Sendra Compte (eds.), Proc. of The Fourteenth International Conference on Systems, ICONS 2019, ISBN: 978-1-61208-696-5, 2019, pp. 7-13.

[2] F. Bacon, The Great Instauration, Complete Works of Francis Bacon, Kindle Format, Minerva Classics, 2013.

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

[4] L. von Bertalanffy, General Systems Theory, George Braziller, 1969.

[5] J. L. Blizzard and L. Klotz, A framework for sustainable whole systems design, Design Studies 33(5), 2012, pp. 456–479.

[6] F. Charnley and M. Lemon, Exploring the process of whole system design, Design Studies 32(2), 2011, pp. 156-179.

[7] J. A. Crowder, Multidisciplinary Systems Engineering: Architecting the Design Process, Springer, 2018.

[8] R. Descartes, Discourse on the method (Discours de la méthode pour bien conduire sa raison et chercher la vérité dans les sciences), in R. Descartes, Œuvres philosophiques (3 vol.). Edition de F. Alquié. T. 1, Classiques Garnier, Bordas, 1988, pp. 567-650.

[9] R. Descartes, Principles of Philosophy (Les principes de la philosophie), in R. Descartes, Œuvres philosophiques (3 vol.). Edition de F. Alquié. T. 3, Classiques Garnier, Bordas, 1989, pp. 87-525.

[10] R. Descartes, Principles of Philosophy, in R. Descartes, translated by J. Cottingham, R.Stoothoff, D. Murdoch: Philosophical Writings of Descartes, vol. 1, Cambridge University Press, 2006, pp. 177-292.

[11] G. Fauconnier and M. Turner, The Way We Think: Conceptual Blending And The Mind's Hidden Complexities, Basic Books, 2003.

[12] V. Filkorn, Character of the contemporary science and its methods (Povaha súcasnej vedy a jej metódy), Vydav. Slovenskej Akadémie Vied, 1998.

[13] M. Franova, A construction of a definition recursive with respect to the second variable for the Ackermann's function, Rap. de Recherche No.1511, L.R.I., Univ. de Paris-Sud, Orsay, France, 2009.

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

[15] M. Franova and 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.

[16] M. Franova and 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, 2018, pp. 1-6.

[17] M. Franova, Trace of computation for ack(3,2), https://sites.google.com/site/martafranovacnrs/trace-of-computation-for-ack-3-2, retrieved 05/14/2020.

[18] M. Franova and Y. Kodratoff, Cartesian Systemic Pulsation – A Model for Evolutive Improvement of Incomplete Symbiotic Recursive Systems, International Journal On Advances in Intelligent Systems, vol 11, no 1&2, 2018, pp. 35-45.

[19] J. Gatial and M. Hejný, Construction of planimetry (Stavba planimetrie), Slovenske Pedagog. Nakladatelstvo, Bratislava, 1973.

[20] J. Y. Girard, Domain of the sign or the bankruptcy of reductionism, (Le champ du signe ou la faillite du réductionnisme), in T. Marchaisse, (dir.): Le théorème de Gödel, Seuil, 1989, pp. 145-171.

[21] K. Gödel, Some metamathematical results on completeness and consistency, On formally undecidable propositions of Principia Mathematica and related systems I, and 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, 1967, pp. 592-618.

[22] J. Hadamard, An Essay on the Psychology of Invention in the Mathematical Field, Read Books, 2007.

[23] S. Hawking, The Theory of Everything: The Origin and Fate of the Universe, Jaico Publishing House, 2008.

[24] M. Hejny, The roots of causal thinking and Greek mathematicians (Korene kauzálneho myslenia a grécki matematici), in S. Znam, L. Bukovsky, M. Hejny, J. Hvorecky, B. Riecan: Pohlad do dejín matematiky, ALFA-SNTL, 1986, pp. 11-83.

[25] C. G. Jung, Symbols of transformation, Princenton University Press, 1956.

[26] D. Kahneman, Thinking, Fast and Slow, Allen Lane, 2011.

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

[28] H. Kopetz, Simplicity Is Complex: Foundations of Cyber-physical System Design, Springer, 2019.

[29] J. L. Le Moigne, Theory of the general system (La théorie du système général, théorie de la modélisation), P.U.F, 1984.

[30] J. L. Le Moigne, Complex systems modeling (La modélisation des systèmes complexes), Dunod, 1999.

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

[32] D. Michie, Machine learning in the next five years, Proceedings of the third European working session on learning, Pitman, 1988, pp. 107-122.

[33] S. Muggleton, D. Lin, and A. Tamaddoni-Nezhad, Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited, Machine Learning 100; 2015, pp. 49-73.

[34] S. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold, Ultra-Strong Machine Learning: comprehensibility of programs learned with ILP, Machine Learning 107, 2018, pp. 1119-1140.

[35] G. L. Murphy and D.L. Medin, The Role Theories in Conceptual Coherence, Psychological Review 92(3), 1985, pp. 289-316.

[36] H. Poincaré, Invention in Mathematics (L'invention mathématique), in J. Hadamard, Essai sur la psychologie de l'invention dans le domaine mathématique, Editions Jacques Gabay, 1993, pp. 139-151.

[37] K. Popper, The logic of scientific discovery, Harper, 1968.

[38] V. K. Ranjith, Business Models and Competitive Advantage, Procedia Economics and Finance 37, Elsevier, 2016, pp. 203 – 207.

[39] C. S. Wasson, System Engineering Analysis, Design, and Development: Concepts, Principles, and Practices, Wiley-Blackwell, 2015.

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

[41] H.P. Zwirn, Complex systems : Mathematics and biology, (Les systèmes complexes : Mathématiques et biologie), Odile Jacob, 2006.