Vision of Information Science Inspired by a New Representational Formalism

Lev Goldfarb
Inductive Information Systems
29 Pembroke Crescent, Fredericton, NB E3B 2V1
Canada
goldfarb@unb.ca

A recently developed formalism for structural representation—called Evolving Transformations System (ETS)—is discussed as suggesting a radically different direction for the development of computer/information science. One can gain an initial intuitive understanding of the ETS object representation by generalizing the *temporal process* of the (Peano) construction of natural numbers: replace the *single* 'structureless' unit out of which a number is built by one of several more general structural units. The new formalism points to the informational view of nature in which the *basic* object encoding is *temporal, event-based*, while the ubiquitous in science *'spatial' object instantiation* can be constructed on the basis of the former. Thus, this far-reaching structural generalization of natural numbers emerges as a universal form of both temporal and structural representation *that can be variously instantiated*, depending on the desired target medium, e.g. 3D-space, biotic, network, etc. Moreover, the ETS representation satisfies a unique and very desirable property not possessed by any other known language, scientific or spoken: its syntax and semantics are congruent.

Keywords*: new representational formalism, structural representation, class-oriented representational formalism, congruence of syntax and semantics.*

1. ON THE ROLE OF REPRESENTATION IN COMPUTER SCIENCE: THE LACK THEREOF

In an earlier paper [1], I have addressed the basic weakness in the original/conventional conception of computation "related to a gross underestimation of the role of object representation in a computational model, hence confining such models to an unrealistic (input) environment, which, in turn, leads to 'unnatural' computational models". I also pointed out that "this lack of appreciation of the role of structural object representation has been inherited from logic [where the roots of computer science are] and partly from mathematics, where, in the latter, the centuries-old tradition is to represent objects as unstructured 'points' ". Here, I first expand on the latter, more fundamental/pervasive reason integrally connected with the *particular historical development of entire mathematics*.

1.1 The Language of Mathematics

To grasp *one side* of the relevant logic of this development, it is sufficient to look carefully into the concept of set—representing, in a sense, its culmination—in which *objects are treated as entities whose internal structure is unspecified.* Basically, having fixed some (typically first-order predicate) language L, a set $\boldsymbol{S}$ is familiarly defined/specified as

$$\boldsymbol{S} = \{ x \in \boldsymbol{U} \mid P(x) \} ,$$

i.e. as a collection of all elements from some fixed (underlying) set $\boldsymbol{U}$ satisfying condition $P(x)$ which is stated in the chosen language L. What is the nature of this 'definition'? We get some help from Willard Van Orman Quine, a prominent philosopher and logician of the last century, who called sets defined in this manner "virtual" sets [2, p. 63], implying that "[t]heir use is mainly guided by two principles, the principles of *abstraction* and *extensionality*" [1]. Moreover, the two principles "do not define the term 'set', but give *contextual definitions* which allow us to eliminate or rewrite references to sets in specific contexts [in proofs].[2] . . . These definitions do not need to be taken as implying the existence of any such objects as sets." [2, p. 64] Thus, the moral is: one should not assume that the sets, which lie at the very foundations of mathematics, have any purchase on reality.

---

[1] The principle of abstraction is: $\forall y \ \left(y \in \{ x \in \boldsymbol{U} \mid P(x) \} \iff P(y) \right)$; and the principle of extensionality is: $\boldsymbol{A} = \boldsymbol{B} \iff \forall y \ \left(y \in \boldsymbol{A} \iff y \in \boldsymbol{B} \right)$.

[2] "Consider for example the proof [of Euclid's lemma] . . . involving the set $\boldsymbol{I}$ of integers of the form $kx + my$. The role of $\boldsymbol{I}$ is only to make it easier to formulate the argument, and every reference to $\boldsymbol{I}$ could be eliminated, at the cost of making formulations somewhat longer and clumsier, by using the condition defining $\boldsymbol{I}$." [2, p. 63]

In addition, it is useful to recall one elementary result from computability theory—existence of sets that are recursively enumerable but not decidable [2, p.126]—also revealing a fundamental weakness in the conventional concept of set and exposing the *basic deficiency of its specification*: this result means that the capability to list the elements of a set is, in general, insufficient for determining the set membership for an element.

To grasp *the other relevant side* of the development of mathematics, one should realize that during the last several centuries not merely the underlying mathematical entities have been treated as 'points' of unspecified structure, but that this initially unspecified structure is 'revealed' only during the later analysis of a chosen axiomatic system[3], e.g. vectors in a vector space. As a result, courtesy of the *previous* development of mathematics, in science, one simply *imposes* the particular *postulated axiomatic structure*, e.g. inner-product vector space, on reality.

In light of the above, the *scientific issue of object representation*, in its full scope, so far has not been—and really could not have been—the concern of mathematics, simply because this issue came to the fore only with the advent of computing, particularly during the last 40-50 years. Whose concern, then, should it presently be?

1.2 On the New Orientation of Computer Science

Now that sets don't connect us with 'the reality', and the classical mathematical axiomatic structures—following in the footsteps of the Pythagorean tradition—impress themselves on the reality, which science 'should' provide this critical *formal* link to nature? [4] I suggest that, despite the present state of affairs and for a number of reasons, it should, actually, be the concern of computer/information science. What partly mollified the issue of object representation in computer science is a false acceptance of various discrete 'representations', e.g. string, trees, graphs, as such. However, by accepting the challenge of developing genuine representational formalisms, our science will contribute, in a very fundamental way, not only to the development of mathematics, but also to all sciences, including natural sciences. *As has been the role of mathematics so far*, computer/information science would develop a radically different kind of scientific language: the formalism for structural representation. Such formalisms should eventually clarify the nature of information processing in the universe, since the representation of each object *in nature* is supposed to support the underlying pervasive form of information processing.[5] Moreover, such formalisms will also unify all sciences, by providing a single structural formal language that *can be used independently* in each of them, in contrast to the present situation with the numeric language, where we have a pyramid of sciences with physics at the bottom.

The historical situation in theoretical computer science was partly addressed in [1], and here I want to complement the latter discussion by addressing very briefly the present applied scene.

1.2.1 The Class of Objects as the Basic Underlying and Unifying Concept

I wish to suggest that, when looking for a reliable useful signpost, there is *only one (and historically a very old) link for us to follow* in order to connect with the information processing in nature—*the concept of class of structured objects*—which, I must stress, exists independently of the human mind. Although historically the latter claim appears to be one of the most debated, splitting philosophers down the middle [6], today, properly interpreted evolutionary evidence—in the form of recurring patterns (classes) both biological and prebiotic—as well as the evidence accumulated in the areas of sensation and perception [8], [9], supports this claim. Since this is a large and very old topic, here I would like to touch on its relevance to information processing as it emerges already today: from the problems arising in connection with the reality of the World Wide Web (including data mining and various search engines), as well as in connection with bioinformatics, cheminformatics, and a wide variety of real-time applications (such as computer vision, speech recognition, person identification, etc.). I suggest that what binds all these, only apparently diverse, applications together is the overall uniformity and simplicity they acquire once formulated in the framework of classes and classification.[7] Although the important role of classification in many, if not most, of these applications has already been recognised, this recognition still has not had a desirable impact on both these areas and computer science as a whole. This is mainly because the recognition has not been reinforced by the development of the *adequate representational formalisms* within which the unifying role of the class concept becomes absolutely transparent. Hence, one of the few key criteria of the adequacy of such formalism is its intrinsic capability to offer a convenient and powerful form of *class representation*.

Here are a few preliminary general remarks on the concepts of (object) class and class representation. Above all, *the concept of class must not be confused with that of a set*, where the former is a much richer concept implying a

[3] Such postulated structures have been so far relatively simple.

[4] As far as logic is concerned, ironically, the relevant fundamental (representational) inadequacy was brilliantly captured by a leading twentieth century *logician* and philosopher Bertrand Russell: "Nature herself cannot err, because she makes no statements. It is men who may fall into error, when they formulate propositions" [3, p. 311; emphasis is added]. Moreover, logical formalisms were generalized from the European languages and similar to the latter have no direct/natural interface with the 'physical' reality (unless one brings into the picture *the more fundamental* sensing mechanisms). For a longer discussion, see [4, Section 9].

[5] Of course, if there is no "underlying" form of information processing in nature, we have no serious scientific goal to begin with.

[6] It is going back to Plato's and Aristotle's forms (e.g. [5], [6]) and also touches on the heart of biology [7].

[7] As mentioned above, I believe that this possible uniformity is not spurious but a consequence of the more ubiquitous feature of the information processing in nature.

common formative structure of its members. It is important to observe that despite the central role of the class concept in many emerging applied key areas of computer science—e.g. bio- and cheminformatics, search engines, machine learning, pattern recognition, data mining—this concept has been consistently misunderstood. There is only one main, and scientifically very 'legitimate', reason for this misunderstanding: the *concept of class cannot be both properly understood and treated within the conventional mathematical formalisms*, since the necessary generative structures are simply not accessible within the classical formalisms [10], [4]. However, I am convinced that this concept can adequately be approached within the new kinds of representational formalisms, formalisms for structural representation. Briefly, the reason has to do with the need for a new, richer generative structure that can support the concept of *class representation*—a structural concept alien to the conventional formalisms[8]—and the latter concept becomes available only within a richer underlying representational structure. Incidentally, among the conventional formalisms I count various probabilistic models, including graphical models[9] (e.g. [11, Chapter 8]).

To illustrate the strategic role of the class-oriented representational formalism—which has remained completely out of computer science sight—let me take as an example a search engine, e.g. Google. Its basic underlying idea is that of record ranking, based on some relatively simple technical principles that don't concern us here. The main observation I wish to make in connection with such general purpose or any other 'semantic' search engines is this: to be most effective, or even simply *adequate*, their organization must be *class-based* organization. What I mean by this is that (evolving) classes of relevant objects must be the main units in such (often hierarchical) organization. Unfortunately, and again due to the lack of the adequate class-oriented representational formalisms, others issues, e.g. ontologies, have come to the fore and continue to take attention away from that conspicuous lack. Why must 'semantic' search engines be class-based? A short answer is: because our perception and therefore our memory are class-based, and so the most natural, and hence most efficient way for us to communicate is via classes. Of necessity, language mechanisms had to evolve on the basis of these class-based perceptual mechanisms. Returning to Google, I am suggesting that various queries in such a general search engine should be built around classes, i.e. the *central* query should be of the following kind: it should be the class query, whose input is typically specified by one or several elements of some class and whose output is all records belonging to that class. For example, if the input query is //papers addressing 'the species problem'// [10], the expected output would be all papers in the corresponding *class* (and not in the set). The point is that even though our (spoken) languages are not ideal for specifying classes, our sensory systems—and thus our semantic capabilities—are class-based.

1.2.2 Representational Formalism = Class-Oriented Formalism for Structural Representation

The main point of this paper is this: the development of an adequate formalism for structural representation not only will radically change the entire computer science, including the state of hardware, but will also completely *redefine* our science. To repeat, the main reason why computer science has so far remained in the dark in this respect has to do with the evident fact that the role of representation has not been sufficiently recognised in the relevant subdisciplines of computer science (e.g. machine learning, pattern recognition, data mining, search engines). As was discussed above, the reasons for the latter state of affairs are deeply connected with the historical development of mathematics and with the mentioned above false acceptance of various 'discrete representation' as such (facilitated by the structure of classical computational formalisms).

As to the title of this subsection, I would like to make a general claim expressed in the title: we should approach the issue of developing representational formalisms as that of *class-oriented* formalisms for *structural representation*. In other words, we should strive for formalisms for structural representation that can support *the inductive concept of class representation*: the class representation should be *constructively defined and* should also be *inductively recoverable* (i.e. recoverable from a *small* set of class members). Although "small" number should be a function of the class complexity, the non-trivial but important stipulation for the inductive recoverability comes simply from our everyday learning experiences.

The classical computational formalisms offer hardly any clues as to the nature of the representational formalism, since they do not work with any legitimate 'representational entities'. For example, a string (or a graph) does not contain enough representational information to allow the inductive recovery of the corresponding grammar: even under reasonable restrictions, there are just too many grammars that contain a chosen small finite set of strings. The latter is due to a simple fact that within a string (or a graph) there is simply *not sufficient information about the string's (or graph's) formative history*. It is precisely this reason why the concept of grammar cannot adequately capture the concept of class: indeed a formal language is defined as a *set* of strings—again, set—so that the class concept could not have emerged within the conventional computational paradigms.[11] Thus, under closer scrutiny it

---

[8] For example, within the vector space formalism, the class is usually delineated by several decision surfaces, which explains why the concept of class representation has not been introduced in this formalism: it cannot be justified on formal grounds.

[9] The 'structural' ingredient in such models is quite rigid/trivial and is completely auxiliary to the basic numeric model.

[10] Note that although a spoken language is not a sufficiently precise form of query representation—e.g. in the above query, "addressing" vs. "discussing", where the former *can* imply a requirement for more substantial discussion—having adopted a representational formalism, for example ETS, it is not difficult to built a special purpose interface for a reasonably reliable conversion of the spoken language query into the chosen formal representation.

[11] Moreover, even from the classical mathematical point of view, there is no *natural topological structure* that can be associated with the set of all strings over a finite alphabet: for example, why should one restrict oneself to single-letter deletion/insertion operations for defining the topology?

becomes clear that the string 'formalism' is not a legitimate representational formalism, but has been constructed ad hoc, on the basis of a written language, which is a very unreliable source for the representational formalism. In fact, it is quite clear that the spoken language has been built on top of the basic perceptual mechanisms, and it is the latter that should serve us as the corresponding source.

The next important question is this: Which further assumption would 'guarantee' that we are on the right track towards developing a *representational* formalism? Although there are, of course, no absolute guarantees, but there is an accumulated philosophical wisdom—the tradition from Heraclites to Hegel, Bergson, and especially Whitehead, Collingwood, and Whyte—which may serve as an adequate guide: I am referring to the view of an object as a process (a structural process to be sure) supported now by the foundations of physics. In other words, we need a formalism in which *objects are viewed and treated as structured processes.* This is, perhaps, the main point to keep in mind. Finally, adding to the latter as a (very reliable) starting point the Peano process of constructing natural numbers, and generalizing its basic successor operation to a structured event, we arrive at the basic idea of the Evolving Transformations System (ETS) formalism, outlined in the next section.

2. OUTLINE OF BASIC ETS CONCEPTS

First of all, it is useful to keep in mind that although its main entities, 'structs', may have a *superficial* resemblance to graphs, the structure of the ETS formalism has absolutely no analogues to compare it with. Of necessity, in what follows, I will have to restrict myself to informal descriptions, while the formal definitions can be found in [12], Parts II and III, which is the principal exposition of the formalism. As was mentioned in the last paragraph, the most important point to keep in mind is that each 'object' in the formalism is both viewed and represented as a *(temporal) structural process*, 'struct' (see Fig. 2), which is an interconnected (temporal) sequence of structured *primitive events, or primitive transformations* introduced next.

2.1 Primitive Transformations, or Primitives

The most basic concept is that of a **primitive event**, or primitive transformation, or simply **primitive**, examples of which are depicted in Fig. 1. Each such primitive event transforms/interrupts the regular flow of several adjacent *primal* processes (of undisclosed structure[12]) specific to this event (the lines just above an event in Fig. 2), and, as a result of the event, new primal processes continue to 'flow' (the lines just below an event in Fig. 2) until, again, some event intervenes. Although the corresponding formal concept is relatively intricate, in another, more important sense, it is simple, since, as will be discussed below, it carries *identical* semantic and syntactic loads.

Again, a **primitive** stands for a *fixed kind of micro-event* [13] (site of interaction of processes) responsible for transforming **initial processes** (top), into **terminal processes** (bottom); see also Fig. 2, where both kinds are shown as lines connecting the primitives. The formal structure of the event is such that it does not depend on the concrete initial (or concrete terminal) processes, as long as each of the processes involved belongs to the corresponding (fixed) class of processes. At this, initial (or 0^{th}), stage of representation[14], the structure of each initial and terminal processes is suppressed, as is the internal structure of the event itself, and what's being captured by the formal structure is the 'external' structure of the event.

Since all of nature is composed of various temporal processes, examples of the above events are all around us: e.g. an elementary particle collision; formation of a two-cell blastula from a single cell (initial process is the original cell and the terminal processes are the resulting two cells); a change in the position of your leg; the event associated with the effect on the listener's memory of the sentence "Alice and Bob had a baby" (initial processes are related to Alice and Bob and the terminal processes to Alice, Bob, and the baby).

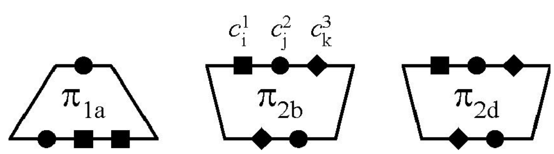

FIGURE 1: Pictorial depiction of three primitives. The first subscript stands for the class of primitives sharing the same structure, e.g. π_{2b} and π_{2d}. **Initial classes of processes** are shown as solid shapes on the top, while **terminal classes** are those on the bottom of each event. The only concrete processes—i.e. the elements of these classes—labelled in the figure are the initial processes of π_{2b}, with label $b = < c^1_i, c^2_j, c^3_k >$, where c^s_t is the t^{th} process in the primal class C_s, $s = 1, 2, 3$.

2.2 Structs

The second basic ETS concept is that of a **struct** [15] formed by a (temporal) sequence of primitives, as shown in Fig. 2.

[12] However, see Section 2.7.
[13] The internal structure of such event is undisclosed. However, see Section 2.7.
[14] In this paper, I discuss *almost* exclusively a single-stage version of ETS (for the multi-stage version, see [12], Part IV).
[15] Or **level 0 struct**, where 0 refers to the **initial representational level**.

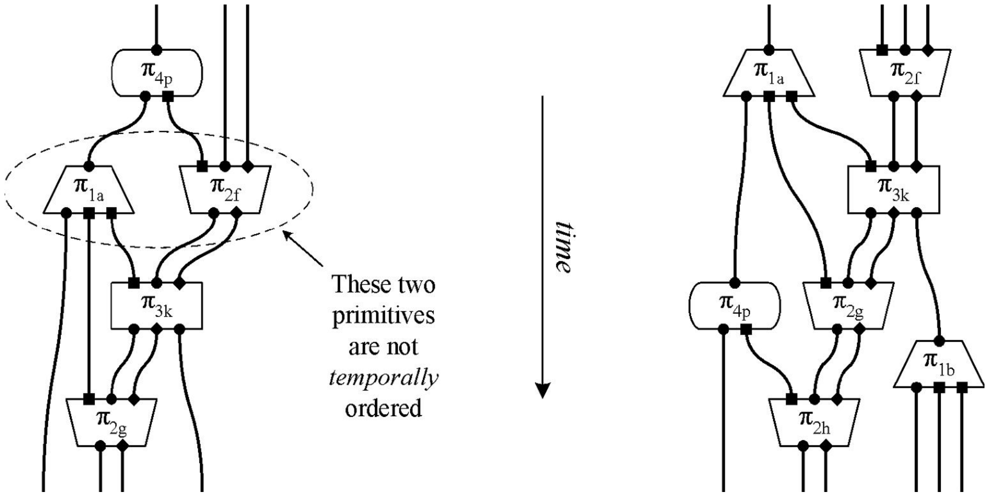

FIGURE 2: Two structs

It is easy to see now how the Peano construction of natural numbers (Fig. 3) was generalized to the construction of structs: the single 'structureless' unit, out of which a number is built, was replaced by one of several structural ones, i.e. by ETS primitives. An immediate and important consequence of the multiplicity of units is that we can now see *which unit was attached and when*. Hence, the resulting (object) representation, for the first time, embodies both temporal and structural information in the form of a formative, or generative, object history recorded

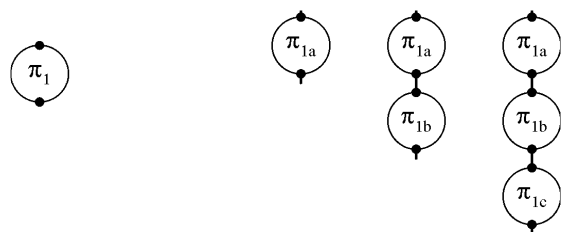

FIGURE 3: The single primitive involved in the ETS representation of natural numbers (left) and three structs representing the numbers 1, 2, and 3.

as a series of (structured) events. Consequently, from both formal and applied points of view, the concept of struct can justifiably be thought of as *the true structural generalization of the numeric representation.* At the same time, one should not confuse the present far-reaching structural generalization with such historical generalizations as complex numbers, quaternions, etc., since all of them are still numeric-based.

2.3 Two Representational Contexts: Subjective and Objective (Syntax and Semantics)

There are two *basic* contexts within which the above concept of object's formative history can appear which are correlated: objective subjective. On the subjective, or an agent's, side, a struct is the recorded sequence of sensory micro-events *during the agent's sensory interaction with the target object*. Such interaction must rely, of course, on the agent's own arsenal of primitives. On the objective, or agent-independent, side, the corresponding struct can be viewed as the representation of the sequence of events that were actually part of *the object's formation and evolution*.

The essential point to observe is that, in ETS, both modes of object representation are captured 'uniformly', i.e. formally identically, which appears to be a novel and desirable feature of a representational formalism: *no other known language, scientific or spoken possesses this property*. The reason why I believe this to be the essential feature of a representational formalism has to do with the considerations I presented in [13][16], in connection with the well-known Searle's Chinese Room argument.[17] In particular, drawing a more general scientific conclusion from the Searle's argument, I suggested that we will not succeed in either understanding or modelling biological information processing unless we come to grips with a representational formalism in which syntax and semantics are congruent: only in the latter case we will not be faced with a serious problem raised by Searle in connection with the artificial intelligence agenda.

[16] I quote freely from that paper.
[17] In this argument (see e.g. [14]), John Searle points out that in any known computational formalism—and, I can add, *in any known language or any formalism in science*—the syntax is not related to the semantics. Moreover, since all present day computers are "purely syntactic machines", he claims, they cannot, in principle, be 'intelligent' in the sense we or other biological species are: they cannot relate to what their programs are executing.

Briefly, a more direct, 'technical' reason for the necessity of the syntax/semantics congruence emerges when we consider some of the consequences of relying on a *representational* formalism that does not satisfy the congruence of syntax and semantics (**CSS**) property. Basically, in such formalism, the *structure* of syntactic constructions is unrelated to the *structure* of semantic constructs.

Indeed, *taking the formal structure of the representational formalism seriously*, I assume that the term '**syntax**' refers to the *underlying formal* (generative) *structure*[18] of the agent's (or internal) representational formalism, while the term '**semantics**' refers to the underlying structure of the agent's (external) environment. Although the latter designation is not standard, I believe it is simple, more precise, and quite productive, as will become clear below.

Now, let us assume that we built an agent functioning on the basis of the representational formalism which *does not* satisfy the above CSS property. It is understood, of course, that our agent's architecture must include a 'sensory' mechanism responsible for realizing a mapping $\boldsymbol{f}$ that maps each admissible external object $\boldsymbol{o}$ into its representation $\underline{\boldsymbol{o}}$ in the agent's 'mind'[19]. Next, let us consider some object $\underline{\boldsymbol{o}}^*$ (internally) constructed (and *not directly sensed)* by the agent in accordance with its internal syntax. What can we say about a *possible* external counterpart $\boldsymbol{o}^*$ of this internally constructed object $\underline{\boldsymbol{o}}^*$? The insurmountable difficulties one is facing when trying to address this question are directly related to the postulated structure of the agent's representational formalism. Indeed, the lack of the CSS property does not allow one to produce a *semantically* meaningful 'reverse' mapping [20] $\boldsymbol{g}$, $\boldsymbol{g}$: $\underline{\boldsymbol{o}}^* \mapsto \boldsymbol{o}^*$, since in order to be able to produce such a mapping the (syntactic) structure of the agent's representational formalism must be correlated with the (semantic) structure of its environment. In other words, although the two structures don't have to be *exactly* identical, their underlying formal (axiomatic/computational) structures must be of the same type: in the latter case, the attempted above two-way correspondence is possible and meaningful: *on both sides of the divide, the objects are composed in the same manner, prescribed by the common formal, or 'computational', structure*.

The above argument also suggests that the syntax of a natural language, at least as it has been understood so far, cannot be indicative of the syntax of the basic/underlying (biological) representational formalism.

In light of the above, there are reasons for believing that any CSS formalism must be a formalism for *'structural'*, rather than numeric, object representation. Basically, to produce a numeric representation, of necessity, one must dismantle the original *structural* information[21] in order to encode it numerically, thus leaving the missing (from the vector representation) structural/semantic information to be supplied by the human mind. The same mind, of course, also had to be involved in the original 'dismantling'. In other words, the 'semantic' information appears to be synonymous with the (properly formalized) 'structural', or 'relational', information.

In ETS, the formal *structure* of the agent's representational formalism (syntax) is identical with formal structure of the agent's environment (semantics), since there is no *structural* distinction between the 'physical' (actual) object representation and the agent's representation, i.e. both are structs. This is because in the formalism, by its very design, the principles of object 'construction' in nature, i.e. 'true' object representation, are exactly the same as those of object representation by an agent. Of course, what makes this structural identity possible is the chosen *event-based form of structural object representation*, associated with the formative object 'history'.

To take an example, consider a flower: its 'actual representation' is formed by the events in its full developmental history, while its representation in an agent's 'mind' is formed by the perceptual events associated with the agent's exploration of this flower. If the agent's sensory mapping is at all 'reasonable'—i.e. if for some fixed representational stage of the actual flower, each *sensory primitive event* captures some structural aspect of the struct representing the actual flower—then there must be a 'reasonable' correlation between the two representations (starting from a particular stage of actual flower representation).

From a pragmatic point, the identity of the two representational formal *structures* (internal and external) is a necessary but not sufficient condition for the CSS property: an arbitrary sensory mapping (an unlikely case) *can* botch the correspondence between the two representations.

Finally, I would like to come back, briefly, to the point made at the end of Section 1.1 and the beginning of Section 1.2 and to emphasize once more the potential leadership role computer science can play by developing an (event-based) representational formalism. Indeed, although during the last one hundred years there has been much work

[18] As has been the accepted practice in mathematics during the second half of the last century, the generative 'rules', the rules for constructing (syntactically valid) formal objects in an axiomatic system must rely on the axioms only. E.g. in the case of the vector space representational formalism, i.e. when representing a handwritten character ***B*** by a vector, one cannot rely on any non-linear—and therefore external to the (linear) axiomatic structure of the vector space—relationships among the coordinates of the vector representing ***B***. Thus, although we can represent various ***B***'s in a vector space, we have no syntactically valid way of deciding which of the (arbitrary) chosen vectors represents a character ***B***.

[19] In the ETS formalism, $\boldsymbol{f}$ preserves some additional structure.

[20] The inverse of $\boldsymbol{f}$ may not exist, since $\boldsymbol{f}$ might well be not an injective mapping.

[21] E.g. non-numeric relationships between various part of ***B.***

done in philosophy, linguistics, psychology, and other fields on various event-based approaches—not only no concrete representational proposals have been forthcoming, but a tremendous controversy persists to date regarding even the possibility of a uniform event-based approach (see e.g. [15] – [17]).

2.4 Level 0 Classes

The third, auxiliary, concept is that of a (level 0) **constraint**—which is a more involved concept not discussed here, see Part III of [12]. It is a formal *specification of a family of structs* sharing structural 'components' in the form of similar substructs. Moving on, the fourth basic concept is that of a **class** of structs**,** which can possibly be multi-levelled. A single-level (or 1-level, or level 0) [22] class is defined via a single-level, or level 0, **class generating system**, which specifies a stepwise mode of construction of the class elements. Each (non-deterministic) **step** by such a system—which always follows a (possible) step by the environment[23]—is specified by the corresponding *set* of (level 0) constraints. During such step in the piecemeal construction process, the struct that is being attached to the part of the class element that has been assembled so far (see Fig. 4) must satisfy one of the constraints specified for this step.

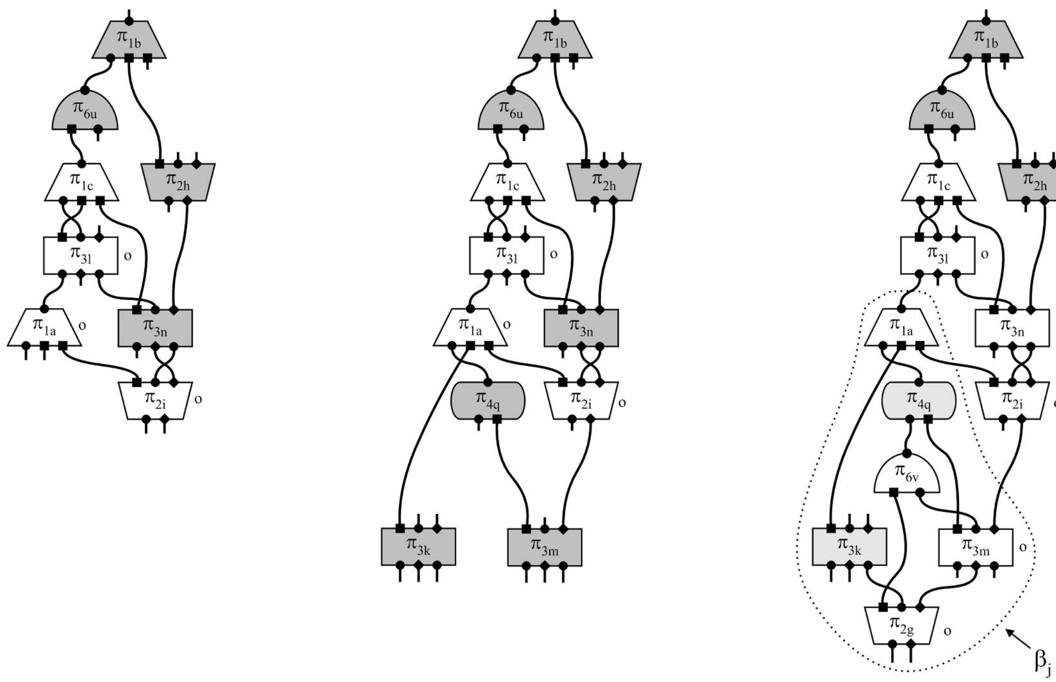

FIGURE 4: Illustration of a generic two-step "unit" in the construction of a (level 0) class element: a step by the environment (resulting in the addition of the bottom three shaded primitives in the second struct) is followed by a step made by the class generating system (substruct β_j partly overlaying the second struct is added to it). The grey primitives are those contributed by the environment and currently not 'absorbed' by the class generating system (via overlaying). An *'o'* next to a primitive in a struct σ signifies that this, *open*, primitive is an *allowable* 'interface' primitive, on which the struct to be attached to σ is allowed to overlap with σ.

As shown in Fig. 4, it is assumed that each 'class' step can be preceded by a step executed by the 'environment', i.e. by some other class generating system *interacting* with the present one. Thus, quite appropriately and realistically, a change in the environment (i.e. in some of its classes) may *change the class elements and hence the class itself, without any change in the class generating system*. Such concept of class admits the effects of the environment in a 'natural' manner.

2.5 Level 1 Structs

Next, suppose that an agent has already learned several level 0 classes, which together form the current **level 0 class setting**. Subsequently, when representing an object, the agent has an access to a more refined form of object representation than a plain level 0 struct: the agent can now detect if the corresponding level 0 struct is, in fact, composed of several *familiar* level 0 class elements, as shown in Fig. 5. This leads to the concept of **the next level (level 1) struct**, which provides *extra, i.e. 'higher level', representational information*, as compared to the underlying level 0 struct itself, since this class-based partition is not part of the struct. (The nature and origin of such partition should get clarifies in the next subsection.)

[22] Note the alternative terminology: "single", or "1" in font of "level" refer to the number of levels involved, while "level 0" refers to the 'name' of the level, which is the number of levels -1.
[23] I.e. by one of several other generating systems.

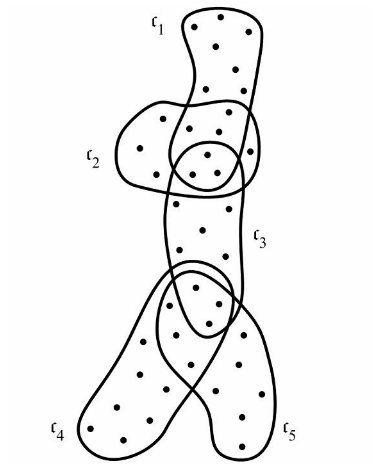

FIGURE 5: Simplified depiction of a level 1 struct in the contracted form: dots stand for primitives and solid lines delineate level 0 class elements c_i's.

2.6 Higher-Level Classes

In the (recursive**) k-level** version of the **class representation**, for k ≥ 2, each corresponding step is specified by a set of **level (k – 1) constraints**. The level (k – 1) struct that is being attached at this step to the previously constructed part of the class element must now be composed out of level (k – 2) admissible class elements[24] only and must satisfy one of the constraints specified for this step.

Analogous to Figure 4, Figure 6 illustrates a (constructive) unit in such a generating process for a level 1 class element.

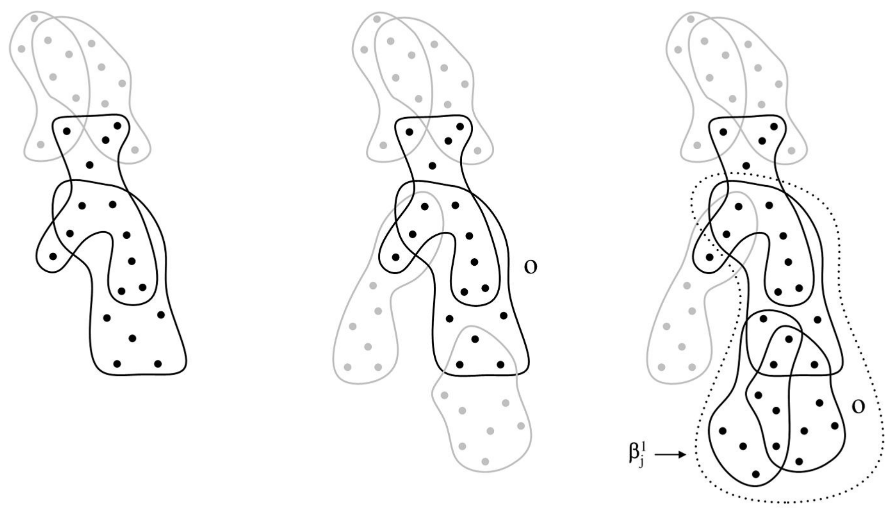

FIGURE 6: Illustration of a generic two-step unit in the construction of a level 1 class element. Dots stand for primitives, and solid lines delineate level 0 class elements, which now serve as basic constructive units. A step by the environment (bottom grey addition in the second struct) is followed by a step made by a level 1 class generating system itself (addition of the shown substruct β_j^1 in the third struct; superscript 1 refers to the level).

When we again move to the next level, for a level 2 class element, such element is an output of a level 2 (or three-levels) class generating system, where at each step, the relevant *part* of this, level 2, class element is composed out of several level 1 class elements and must satisfy one of the level 2 constraints specified for this step.

2.7 Transition to the Next Representational Stage

To clarify the nature of the most basic ETS concept, a primitive, I outline very briefly the concept of **the next stage of representation**. Transition to the next representational stage is associated with a representational compression, in which certain recurring (global) patterns of process interactions, called **transformations**, are compressed into new primitive transformations (for the next stage): each of the interacting processes is compressed into a primal

[24] Each of those must come from a class belonging to a (previously learned or given) set of level (k – 2) classes, comprising the current level (k – 2) class setting.

process and the segment in which the interaction between them occurs is compressed into a **next stage primitive** event (see Fig. 7).

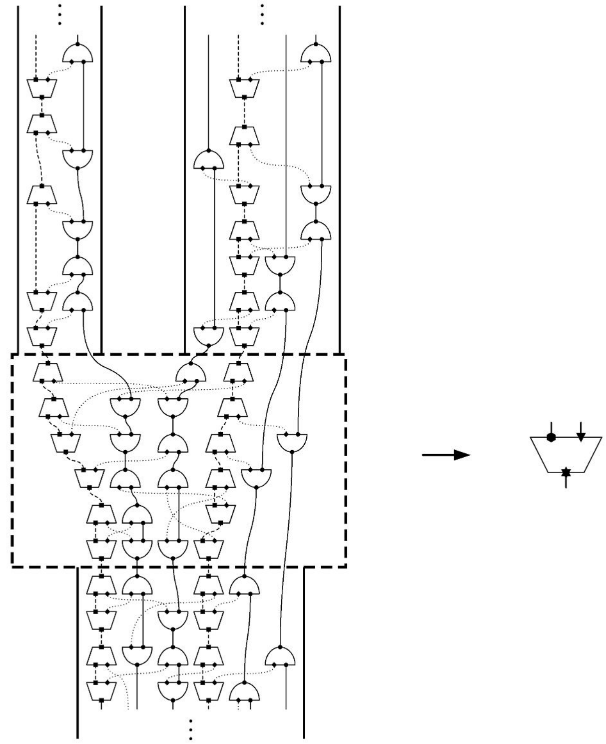

FIGURE 7: A transformation (left) and the corresponding next-stage primitive (right). An illustration of a transformation corresponding to a hypothetical formation of a lithium hydride molecule (terminal process) from hydrogen (left) and lithium (right) initial processes. (Note the reoccurring structural patterns in each of the initial processes: they capture the electrodynamic interactions between the nuclei and the electrons.) The four primitives involved represent emission/absorption of a photon by electron (semi-circles) or nucleus (trapezoids). The body of the transform—delineated by the heavy dashed line—depicts an imaginary restructuring of the two initial processes into the terminal one. The corresponding next-stage primitive—the lithium hydride formation event—is shown on the right.

Note that a transformation is associated with a *disruption* (and the consequent restructuring) of the regular flow of several adjacent processes. In contrast to the latter, the case when several processes simply overlap, without being disrupted, is handled by a simple struct operation called *struct assembly* (not introduced here but could be thought of a struct 'union').

Thus, the primitives at the next representational stage include 'compressed' transformations from the present stage and possibly some primitives from the present stage (that are simply lifted to the next one). As a result, for the fist time, ETS offers a *seamless integration of representational stages* within a single formalism.

3. ETS (TEMPORAL) REPRESENTATION AND ITS INSTANTIATIONS

Coming back to the basic concept of ETS representation, i.e. that of a struct, I wish to emphasise its qualitative difference as a form of representation: *it is a 'purely' temporal/structural representation*, *with which we previously have had no experience at all*, since the main and practically the only forms of applied scientific representation have been numeric representations.

In connection with this, there arises a very natural question that is still not easy to address adequately: How is the ETS representation related to the conventional spatial/numeric, i.e. embodied, representations? It appears that the former is a more abstract, universal, and compact form of object representation that *can be variously instantiated* into more familiar, e.g. spatial, forms of representation. In [12], we give an example of 2D instantiation of the ETS "Bubble Men" class illustrating this idea. It is more fully developed as a 3D example in [18]. Here, I include just two figures—Figs. 8,9 below—out of seventeen in [12] to suggest the basic idea of the example.[25] Note that there are *various* spatial instantiation (in terms of concrete spatial choices/embodiments) *all of which would share the same original temporal representation*.

The proposed formalism also suggests that its temporal, or structural, representation can be thought of as leading to a new, more universal form of the event-based 'object-oriented programming' language, since *any class of processes* can be represented within the ETS formalism. Such 'programs' can be instantiated/run on many different kinds of hardware, including biotic hardware.

[25] Of course, the particular class representation was of main interest in [12].

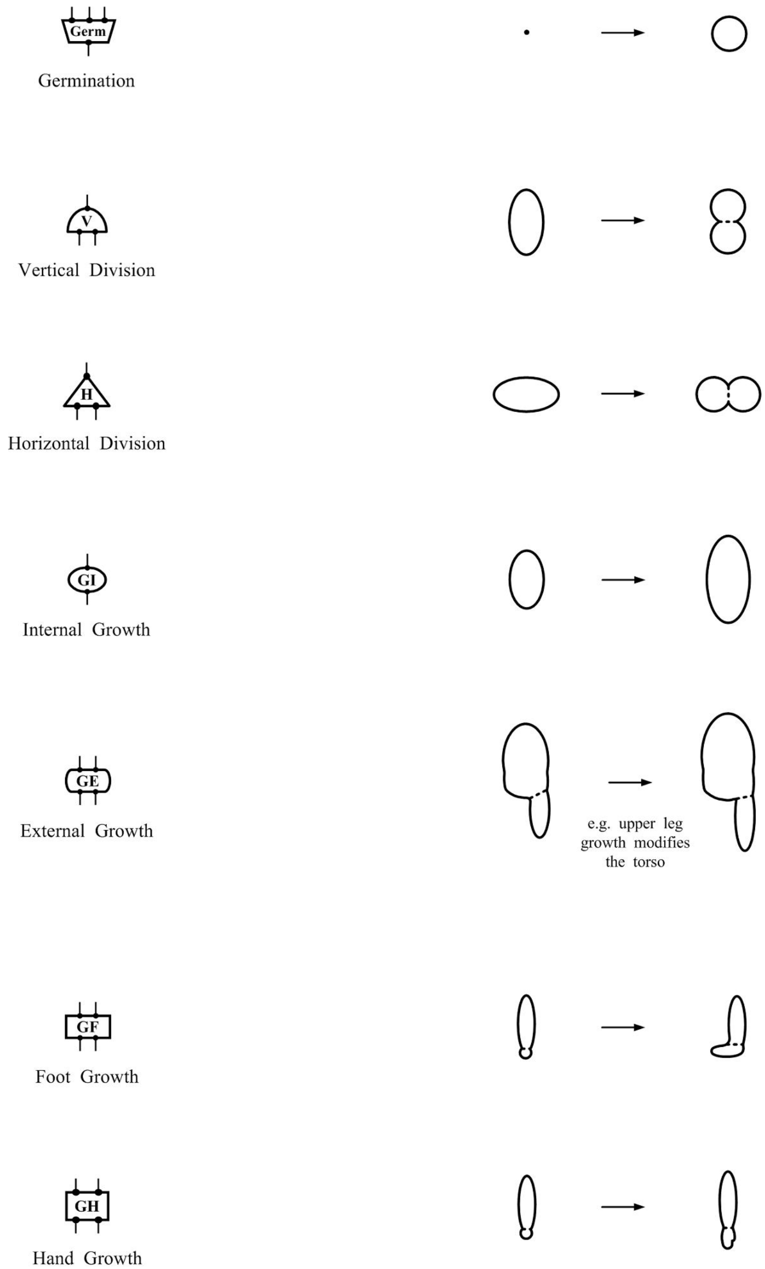

FIGURE 8: Left*:* The primitives used in this example. All primal processes (except the initials for germination) belong to the same class of *oval-like "cells"*, i.e., there is only one primal class. The last three primitives have similar structure: the left initial is associated with a single cell, and the right initial is associated with one of its neighbors; the left terminal is associated with the enlarged original cell and the right, with the appropriately modified neighboring cell. So, the three events each produce the modification of the original cell *and also* of its neighbor. (One could have introduced similar primitives with more initial and terminal sites, responsible for the modification of several neighboring cells, and in general, one can also 'split' each of these primitive into several.) Right: 2D instantiations of the corresponding primitive events.

4. CONCLUSION

I began the paper by pointing out some fundamental inadequacies of the concept of *set* as the foundational conceptual link to the reality of nature.[26] I suggested that, in this capacity, it should be replaced with the concept of *the generative class of (structured) processes*—which is an unprecedented undertaking—and that computer science has the opportunity to become a pathbreaker in developing the appropriate formalisms. An example of such formalism (ETS) was also outlined.[27] Before I rest my case, I would like to quote two quite different sources, which are more than half a century apart. The first one is authored by the eminent American biochemist Franklin M. Harold and the second one, by the brilliant late Oxford philosopher, historian, and archaeologist Robin G. Collingwood:

[26] Cantor himself inadvertently pointed in this direction when he introduced the set as "a multitude that can be thought of as one": for a multitude to be one, this multitude must have a common *generative* origin, a defining characteristics of the class.
[27] For some preliminary applications see [18] – [24].

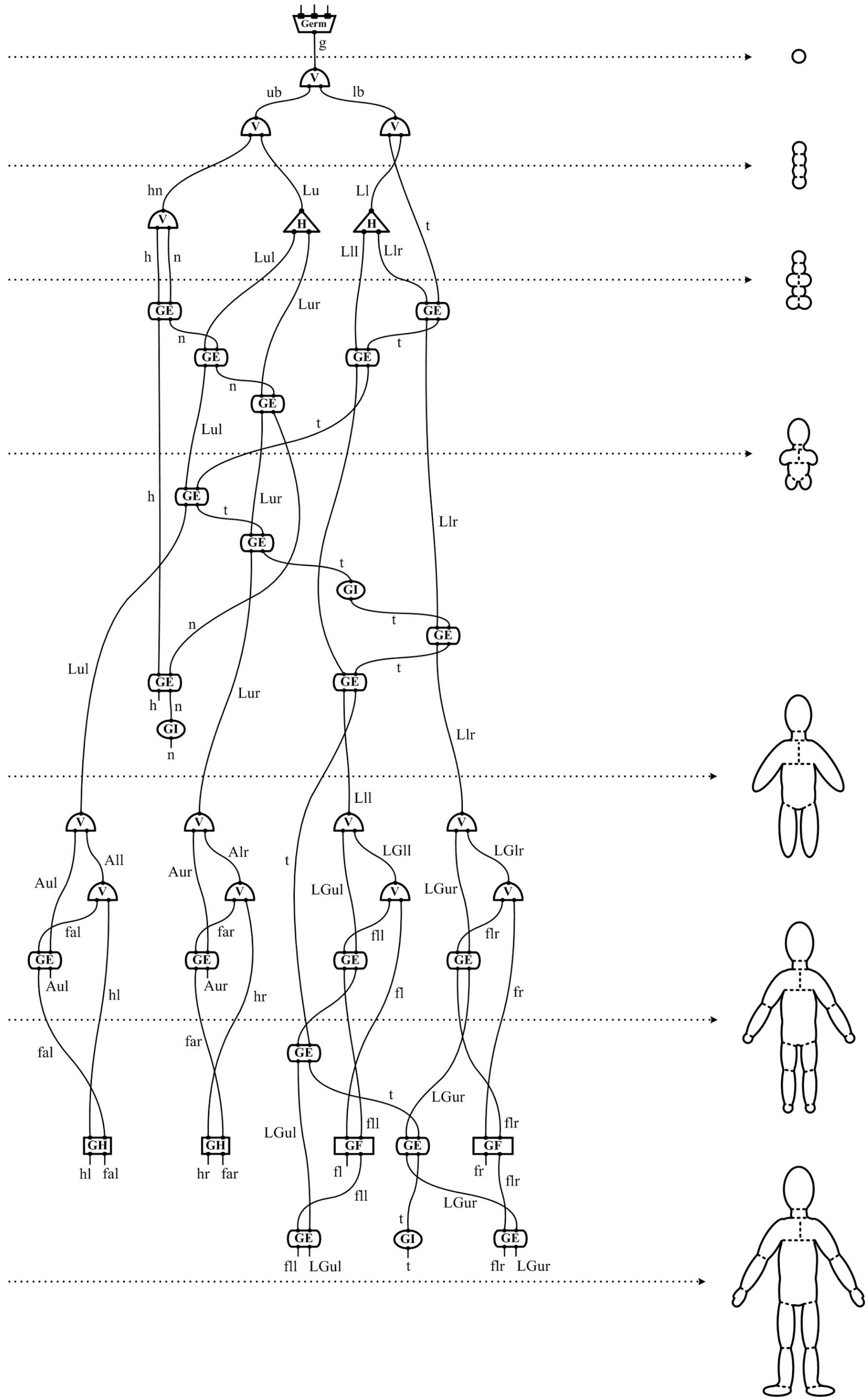

FIGURE 9: Left: A level 0 struct from the Bubble Man class. (For the corresponding 3-level class representation see [12, Section 8). Right: A temporal sequence of 2D instantiations of that struct as it is being built by the class generating system.

> So are we all waiting, not necessarily for a recipe but for *new techniques of apprehending the . . . past*. Without such a breakthrough, we can continue to reason, speculate and argue, but we cannot know. *Unless we acquire novel and powerful methods of historical inquiry, science will effectively have reached a limit*. [25, p. 252; my emphasis]

> [I]f natural science is a form of thought that depends for its existence upon some other form of thought, we cannot adequately reflect upon what natural science tells us without taking into account the form of thought upon which it depends.
> What is this other form of thought? I answer, 'History'.
> . . . A scientific fact is an event in the world of nature. A scientific theory is a hypothesis about that event, which further events verify or disprove. . . . 'The fact that event has happened' is a phrase in the vocabulary of natural science which means 'the fact that the event has been observed'.
> . . . .
> From this I venture to infer that no one can understand natural science unless he understands history; and that no one can answer the question what nature is unless he knows what history is. [26, pp. 176, 177]

Acknowledgement. All figures are adopted from [12].

REFERENCES

[1] Goldfarb, L. (2008) Representation before computation. To appear in *Natural Computing*. http://www.cs.unb.ca/~goldfarb/NaturalComputing.pdf
[2] Franzén, T. (2004) *Inexhaustibility*. Association for Symbolic Logic & A K Peters, Wellesley, Massachusetts.
[3] Russell, B. (1989) *Wisdom of the West*. Crescent Books, New York.
[4] Goldfarb, L. (2006) Representational formalisms: what they are and why we haven't had any. http://www.cs.unb.ca/~goldfarb/ETSbook/ReprFormalisms.pdf
[5] Loux, M. (2006) *Metaphysics*. Routledge, New York.
[6] Schrödinger, A. (1992) *The problem of Universals*. Humanities Press, New Jersey.
[7] Stamos, D. (2003) *The Species Problem*. Lexington Books, Lanham, Maryland.
[8] Harnad, S. (ed) (1987) *Categorical Perception*. Cambridge University Press, New York.
[9] Goldstone, R. (1994) Influences of categorization on perceptual discrimination. *J. Exp. Psych. General* **123,** 178–200.
[10] TR07-185 (2007) On the Concept of Class and Its Role in the Future of Machine Learning. http://www.cs.unb.ca/~goldfarb/ETSbook/Class.pdf
[11] Bishop, C. (2006) *Pattern Recognition and Machine Learning*. Springer, New York.
[12] TR07-187 (2007) What is a structural representation? A proposal for an event-based representational formalism, Sixth Variation. Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada. http://www.cs.unb.ca/~goldfarb/ETSbook/ETS6.pdf
[13] Goldfarb, L. (2008) Representational formalism in which syntax and semantics are congruent: towards the resolution of Searle's Chinese room challenge. *Proc. 30th Annual Meeting Cognitive Science Society*. http://www.cs.unb.ca/~goldfarb/Congruence.pdf
[14] Preston, J., & Bishop, M. (eds) (2002). *Views into the Chinese Room*. Clarendon Press, Oxford.
[15] Casati, R. and Varzi A. (2006) Events. In Zalta, E (ed), *The Stanford Encyclopedia of Philosophy (Summer 2006 Edition)*. http://plato.stanford.edu/archives/sum2006/entries/events
[16] Tenny, C and Pustejovsky, J. (eds) (2000) *Events as Grammatical Objects*. Center for the Study of Language and Information, Stanford, California.
[17] Shipley, T. and Zacks, J. (eds) (2008) *Understanding Events*. Oxford University Press, Oxford.
[18] Peter-Paul, R. (2008) *On the role of temporal and spatial representations in light of ETS formalism*. Master's thesis, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada.
[19] Korkin, D. (2003) *A New Model for Molecular Representation and Classification: Formal Approach Based on the ETS Framework*. Ph.D. thesis, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada. http://www.cs.unb.ca/dima/phd/thesis.ps
[20] Falconer, S. (2005) *On the Evolving Transformation System Model Representation of Fairy Tales*. Master's thesis, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada. http://www.cs.unb.ca/~goldfarb/Theses/Sean.pdf
[21] Gutkin, A. (2005) *Towards Formal Structural Representation of Spoken Language: An Evolving Transformation System (ETS) Approach*. Ph.D. thesis, School of Informatics, University of Edinburgh, UK. http://www.cstr.ed.ac.uk/downloads/publications/2005/gutkin_phd_full.pdf
[22] Goldfarb, L., Scrimger, I., and Peter-Paul, R. (2008) ETS as a Structural Language for Decision Modeling and Analysis: Planning, Anticipation, and Monitoring. To appear in *Risk and Decision Analysis*. http://www.cs.unb.ca/~goldfarb/Decision.pdf
[23] TR07-184 (2007) On ETS Representation of Human Movement. Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada. http://www.cs.unb.ca/~goldfarb/Movement.pdf
[24] Scrimger, I. (2007) *Structural Representation of the Game of Go*. Master's thesis, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada. http://www.cs.unb.ca/~goldfarb/Theses/Ian.pdf
[25] Harold, F. (2001) *The Way of the Cell*. Oxford University Press, Oxford.
[26] Collingwood, R. (1945) *The Idea of Nature*. Clarendon Press, Oxford.