State-of-the-art Schema Mechanisms for Developmental Artificial Intelligence

Olivier L. Georgeon[1] [0000-0003-4883-8702],
Filipo S. Perotto[2] [0000-0003-2283-4703], Kristinn R. Thórisson[3],
Arash Sheikhlar[3], and Paul Robertson[4] [0000-0002-4477-0379]

[1] UR CONFLUENCE: Sciences et Humanites (EA 1598), UCLy, France
ogeorgeon@univ-catholyon.fr
[2] ONERA – The French Aerospace Lab – DTIS, Toulouse, France
filipo.perotto@onera.fr
[3] Center for Analysis and Design of Intelligent Agents, Reykjavik University, Iceland
thorisson@ru.is, arash@iiim.is
[4] DOLL Labs, Lexington, MA, USA
paulr@dollabs.com

Abstract. Schema mechanisms are software frameworks designed to reflect learning theories like genetic epistemology, constructivist psychology, and developmental learning. These theories find roots in the work of Jakob von Uexküll on animal behavior, Ernst von Glasersfeld on epistemology, and, more centrally, Jean Piaget ranging from child psychology to theory of knowledge. Gary Drescher coined the generic term schema mechanism and pioneered their implementation in 1991. His work laid the ground in the Constructivist AI field and identified the core challenges: open-ended learning, intrinsic motivation, discovering of empirical regularities, incremental hierarchical abstraction, concept invention, sensorimotor grounding, reflexivity, individuation. This paper reviews a selection of schema mechanisms from the literature, and highlights their contributions to these key challenges, with the goal of unifying their different contributions toward a comprehensive theory of developmental artificial intelligence.

Keywords: schema mechanism · genetic epistemology · constructivist learning · intrinsic motivation · cognitive architectures.

1 Introduction

Throughout his life, Jean Piaget developed and popularized the theory of *genetic epistemology* [32] to account for the genesis of intelligence and knowledge. In parallel, he pioneered *developmental psychology* by establishing the foundational methods for studying mental development in children. Toward the end of his life, he connected genetic epistemology with constructivist epistemology and the work of Ernst von Glasersfeld [11]. Genetic epistemology rests on the key concept of "scheme", closely related to the concept of "functional circles" proposed by Jakob von Uexküll. Ziemke [46] provides a broader examination of the philosophical roots of constructivist epistemology in Kant's philosophy, and Uexküll's biology. Piaget defines a scheme as follows:

> "A scheme is a structure or organization of actions as they transfer or generalize in similar or analogous circumstances." ([30], p. 23).

In this paper, we translate the French term *scheme* with the English term *schema* and we use the plural *schemas*. A schema is the basic unit of knowledge that encapsulates the action and its circumstances, that is, a *pattern of interaction*. Genetic epistemology insists on the primacy of interaction as a condition for the emergence of perception and knowledge, surpassing innatist and empiricist explanations:

> "Knowledge does not originally arise either from a subject conscious of itself or from objects already constituted (from the subject's point of view) that would impose themselves on the subject. Knowledge results from interactions occurring halfway between the subject and the objects, and thus involving both, but due to a complete un-differentiation and not from exchanges between distinct forms.
>
> If, at the beginning, there is neither a subject, in the epistemic sense of the term, nor objects, conceived as such, nor, above all, invariant instruments of exchange, then the initial problem of knowledge will be to construct such mediators. Starting from the contact zone between one's own body and the objects, these mediators will progressively engage more deeply in both complementary directions toward the exterior and the interior. It is from this dual progressive construction that the joint elaboration of both the subject and the objects depends.
>
> The initial instrument of exchange is not perception, as rationalists too easily conceded to empiricism, but rather action itself, with its much greater plasticity. Certainly, perceptions play an essential role, but they partly depend on action as a whole, and some perceptual mechanisms that one might have thought to be innate or very primitive only emerge at a certain level of object construction." (translated from [31], p. 14-15)

To this day, Piaget's ideas remain central to modern theories of knowledge as well as developmental psychology. Indeed, genetic epistemology seems to offer a materialistic explanation of how human psychology develops from birth. The question whether this explanation is too reductionist remains however open. To our knowledge, Piaget never claimed that his theories could be implemented as a mechanistic process in a computer. The first attempt only began eleven years after his death.

[4] [13] [19]

2 Schema mechanisms

Implementing genetic epistemology as a mechanistic process involves reducing Piaget's psychological concept of schema into data structures handled by a computer. This computer controls an artifact (robot or virtual agent) that interacts with its environment. The computer must process these data structures through some mechanism that will support the generation of increasingly intelligent behavior demonstrating the internal development of the robot. Guerin and McKenzie [12] proposed the graphical representation of this developmental process shown in the Figure 1.

Notably, the developmental process is not targeted at reaching a predefined goal-state in a predefined problem space, nor does it aim at maximizing a predefined scalar value as in reinforcement learning. Instead, the problem of cognitive development in artificial systems is intertwined with the problem *intrinsic motivation*, as has been argued for example by Oudeyer [24].

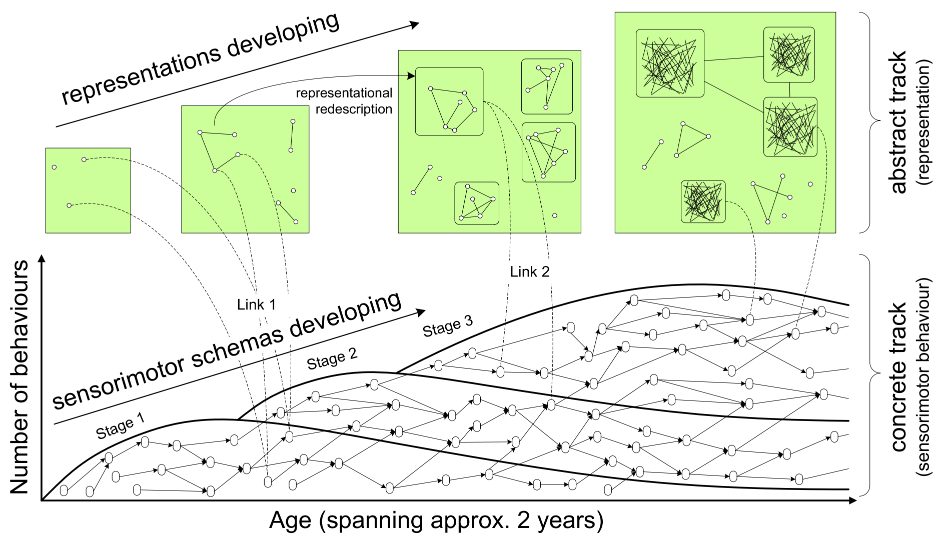

Fig. 1: Conceptual diagram of infant development from [12] Fig. 1. The lower (concrete) track shows a directed acyclic graph of sensorimotor schemas. A node represents a newly created schema. An edge has the meaning "is a necessary precursor". Stage 1: behaviors without objects. Stage 2: behaviors with single objects. Stage 3: object-object behaviors. The schemas now involve relationships among objects, and locations and transforms within space. The higher (abstract) track represents representations of objects by schemas and physical properties influencing their interactions.

2.1 Drescher's

Drescher notes that "a constructivist account of the development of intelligence holds that the difference between the mind of an adult, and that of an infant, lies in mental structures built by the individual" [6, p. 41]. He pioneered the first schema mechanism in 1991 by designing these mental structures as schemas formalized as the tuple (context, action, result) depicted in Fig. 2. The *context* and the results are sets of binary *items* that can be *positively* or *negatively* included. A schema's context is satisfied when all the positively included items are *On* and all the negatively included items are *Off*. A schema's result asserts that, if the action is taken when the context is satisfied, then the positively included items of the results will be turned *On* and its negatively included items will be turned *Off*.

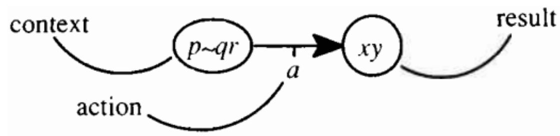

Fig. 2: Schema from [6] Fig. 3.2. The schema noted $p \sim qr/a/xy$ asserts that if action a is taken in the context where item p is On, q is Off, and r is On then the items x and y will be turned On.

The agent is initialized with a set of *primitive* actions, items, and schemas that encode the agent's initial "innate" behaviors. The activation of a schema consists of initiating its action. Upon completion of the action, the predicted result is compared with the actual result. The schema is said to *succeed* if its predicted results are all obtained, and to *fail* otherwise. The ratio of success and failure is called *reliability* and is memorized for each schema.

The learning of new schemas is triggered by the identification of *relevant results* made of items that are slightly more frequent than average. When a relevant result is identified, the agent seeks conditions under which it follows reliably. Such conditions are found through new schemas that allow recovering the context leading to the relevant result, and then performing the action again to asses the reliability of these new schemas.

The learning of new "concepts" occurs when the agent finds schemas that act as "probes" that may evoke the manifestation of a "thing". These schemas are only reliable if the "thing" is present. When such a schema is found, it is considered a "host schema" for a newly spawned "synthetic item" that represents this thing. Eventually, the agent finds out that a single "thing" may manifest itself through different host schemas. Drescher notes that "a synthetic item thus works backward from a thing's manifestation to define the very thing manifested" [6, p. 83]. This addresses the fundamental problem of *concept invention* in that "a synthetic item is a new element of the system's ontology—an element fundamentally different from the prior contents of the system's conceptual vocabulary" [6, p. 81].

The designer defines the agent's goals by assigning *primitive values* to items (positive if desirable, negative if dreaded). The mechanism can identify subgoals as specific sets of items to be activated or deactivated; and it constructs *composite actions* as sequences of schemas used to reach a subgoal. The values of a target item may be propagated to other items in the form of *instrumental values* and *delegated values* when attaining these other items improves the chances to reach the target item. The agent randomly alternates between explicit goal-pursuit and exploration, with probability defined by the designer.

Drescher demonstrates its schema mechanism in the "microworld" made of the two-dimensional grid shown in Fig. 3. "The mechanism controls a simulated robot that has a body, a single hand, and a visual system. The hand can touch and grasp objects, and move them about. The visual system maps a visual field onto a region of the world in the immediate vicinity of ther robot body" [6, p. 114].

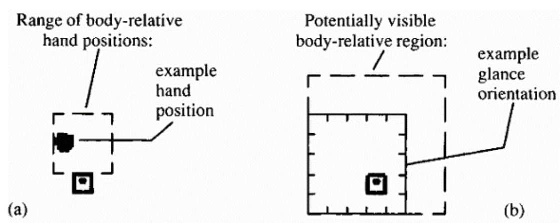

Fig. 3: Drescher's benchmark from [6, Fig. 6.1]. The simulated robot's body is represented by the small square with a black dot. (a): The hand (black spot) can move in the 3×3 grid in front of the body. (b): The visual field is the 5×5 grid in the environment made of a 7×7 grid.

The mechanism initially ignores the topological relations among sensory items, but the demonstration shows that it gradually learns these relation in a process that Drescher calls the "elaboration of the spatial substrate". Such learning of a cognitive map from sensorimotor sequences has recently been further investigated by Raju et al. [34].

Once the spatial substrate is learned, the mechanism manages to construct synthetic items that designate objects as distinct from their current perceptions. Drescher gives the example of "a synthetic item, which we might call `PalpableObjectAt(1,3)` (the name, of course, has no meaning to the mechanism). The item is defined to represent whatever condition makes the schema reliable; we observers know (but the mechanism does not) that this condition is that there be an object at (1, 3)" [6, p. 13]. Synthetic item `PalpableObjectAt(1,3)` may later be associated with `VisibleObjectAt(1,3)` to account for the discovery that they refer to the same thing at position (1, 3) in the world.

In summary, Drescher's schema mechanism stands among the first attempts to investigate Piaget's question of how the child can dynamically construct it's own knowledge of reality from experience of interaction. In our view, Drescher's main contribution to this question rests on the construction of synthetic items to account for "things in the world" that the agent can discover through regularities in sequences of sensorimotor experience. Drescher goes beyond his predecessors (e.g., [14]) who merely categorized passively-received inputs. His schema mechanism was the first that "grounds its synthetic items in the reification of counterfactual assertions" [6, p. 90].

Drescher identified some limitations that may explain why nobody has made his mechanism work in a more complex environment since its creation more than 30 years ago. These limitations follow from the acknowledgment that "whatever experiences an organism or mechanism has had, there are infinitely many mutually contradictory generalizations that are consistent with those experiences. Therefore, any induction apparatus must impose a choice among those consistent generalizations" [6, p. 174]. Among these choices, he acknowledges that "Looking for persistence is built into the schema mechanism's synthetic item facility; thus, the significance of persistence is innate to the mechanism, rather than being acquired" [6, p. 84].

His schema mechanism may lack generality, for example by assuming that a sensory item "is a state element. Each item represents some condition in the world" [6, p. 56]. This assumption presupposes that the world be represented as a predefined set of states, which is acceptable in a microworld but difficult in the physical world. Another assumption was to model the agent's motivation through values associated with sensory items representing elements of the world. Moreover, the exploration mechanism based on hysteresis and habituation also implements designer's assumptions [6, p. 66]. Other motivational principles can be imagined. Conversely, a schema mechanism may need additional built-in assumptions to cope with the complexity of the physical world. This paper reviews other schema mechanisms that make other choices in the hope to move forward towards a general schema mechanism that could work in the real world.

2.2 Perotto's

In the early 2010s, Filipo S. Perotto proposed his own version of a schema mechanism as a result of his PhD thesis [25] and related publications [28, 27, 29, 26]. The algorithm is called CALM, for *Constructivist Anticipatory Learning Mechanism.*

Perotto tacked the same challanges than Dresher, namelly: learning a model of the regularities observed by an agent while interacting with the environment, and extending the representational vocabulary to allow the agent to overpass the raw sensorimotor perception by the creation of more abstract concepts. Also similar to the original Sechema Mechanism, the elementary knowledge structure in CALM are schemas, which are tuples that associate a context, action, and expected results. Perotto, however, adopts a different approach. While Dresher's mechanism is closer to the 1980s symbolic planning languages and structures, Perotto's background lied on reinforcement learning [37] and multiagent systems.

In this way, CALM can be seen as a kind of model-based RL algorithm, trying to learn a model of the transitions (i.e. how the observations change depending on the context and actions), then, at each time step, computing a policy of actions that maximizes the expected discounted sum of rewards, similar to adaptive dynamic programming methods. The main drawback is that CALM can only identify regularities if they are deterministic, differently from the original Schema Mechanism, which is able to detect probabilistic regularities.

To compensate the limitation on deterministic regularities, CALM introduces a new distinction between two kinds of context variables: hidden variables and observable variables as shown in the Fig. 4. The justification is some regularities appear as non-deterministic because the agent cannot perceive all the contextual elements necessary to precisely determine the transition. In this way, the accent is placed on the discovering of abstract or non-observable elements, that once discovered and integrated in the anticipatory knowledge, are able to make the observation deterministic. The idea is, in fact, quite similar to what Dresher called synthetic items in the original mechanism, but the way to create them is different in CALM.

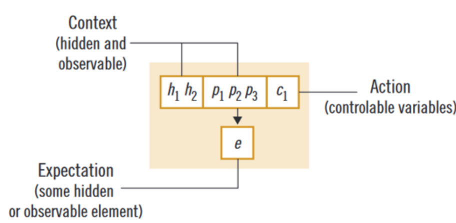

Fig. 4: Perotto's schema from [26] Fig. 2. A schema is composed of three vectors: context containing hidden variables and observable properties $(h_1, h_2, p_1, p_2, p_3)$, action (c_1), and expected result (e).

Another difference of CALM in relation to the original mechanism is that the schemas are organized in a structure called *anticipatory tree*, represented in Fig. 5, which defines a hierarchy, like a decision tree, starting from a completely general context and action, the root node, going to the most specialized ones, the leaf nodes, which represent the schemas.

Fig. 6 shows Perotto's experimental settings [26].

2.3 Learning Intelligent Decision Agent (LIDA)

The LIDA cognitive architecture [18] implements a schema mechanisms inspired by Drescher on top of a perceptual memory module called the Perceptual Associative Memory (PAM). Percepts are not directly received from the environment

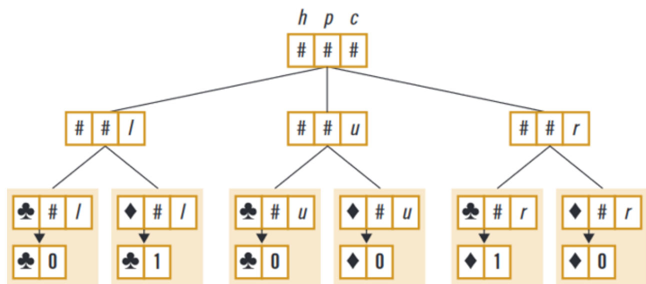

Fig. 5: Example of Perotto's anticipatory tree from [26] Fig. 10: "Final schematic tree for solving the flip problem.".

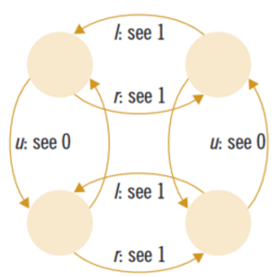

Fig. 6: Perotto's benchmark from [26] Fig.9 "The hyper-flip problem".

but constructed in the PAM through an active sensorimotor process. Other memory modules are implemented including episodic memory, declarative memory, and spatial memory. Knowledge is represented as a directed graph called the *node structure*.

LIDA's schemas work as templates with unbound variables in their context, action, and result. Actions may have unbound variables that qualify and modulate how an action is executed. Schemas are instantiated as *behaviors*.

LIDA implements *affective valence*. Affective valence can bias behavior selection towards desirable outcome. Liking and disliking are distinguished from wanting and dreading with the argument that they relate to distinct neural pathways in the brain [16]. Wanting and dreading are qualified by *incentive salience* of behaviors.

Kugele [17] recently improved LIDA's the schema mechanism to demonstrate contstructivist learning. Drescher's items are replaced by *amodal nodes* organised in the node structure in Percepual Associative Memory. An important difference between LIDA's amodal nodes and Drescher's items is that amodal nodes may not only represent sensory data but also events. Consequently, LIDA can not only represent a particular context in terms of elements of the world state, but also in terms of sensorimotor events.

Fig 7 shows the cognitive cycle. The Dashed rounded rectangle at the bottom shows the schema mechanism as a part of the architecture. The Global Workspace (right) stores a description of the current situation as a graph node. The Procedural Memory module (bottom right) is the memory of schemas. The graph node from Global Worskpace activates relevant schemas. The Action Selection module selects an action, primitive or composite, from among the actions of the relevant schemas. Once the action is selected, it is instantiated as a *behavior*, or a *stream of behavior* for composite actions, and their unbound variables are qualified. The behavior is enacted in the Environment (left) and sensory stimuli (signal) is received by the Sensory Memory module (upper left). Sensory signal is processed to produce amodal nodes that represent the current situation. The Attention Codled module selects some amodal nodes to broadcast in the Global Workspace Module.

Kugele [17] demonstrated constructivist procedural learning within LIDA in a multi-armed bandit environment. When the agent is standing (State S), it can take the action to sit at a one-armed bandit machine, for example Machine 1 (State M_1). The agent can then pay a deposit (state M_1P), and then play, which may yield a win (State M_1W) with probability $p(W|M_1)$, or a loss (State M_1L) with probability $p(L|M_1)$. The agent can choose to stand from any state. Fig. 8 represents the state diagram of this environment in the case of two machines.

The set of possible actions for k machines is $\{$`stand`, `sitM1`, ..., `sitMk`, `play`, `deposit`$\}$. The set of initial nodes (i.e., primitive items) is $\{$`Win`, `Loose`$\}$. Node `Win` is associated with the affective valence (i.e., primitive value) of $+1$, and Node `Loose` of -1. The difficulty comes from that the agent cannot directly perceive the state of the environment. The results show that the agent successfully constructed the amodal nodes (i.e., synthetic items) $\{$`M1`, ..., `Mk`, `M1P`, ..., `MkP`$\}$ that

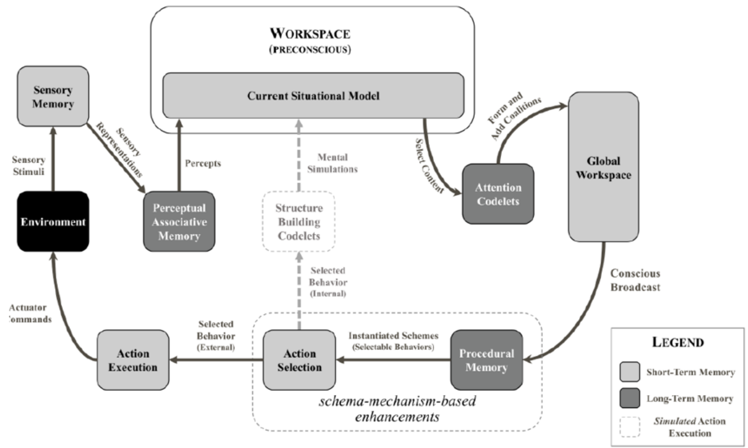

Fig. 7: Simplified depiction of the LIDA cognitive cycle [17, Fig. 4].

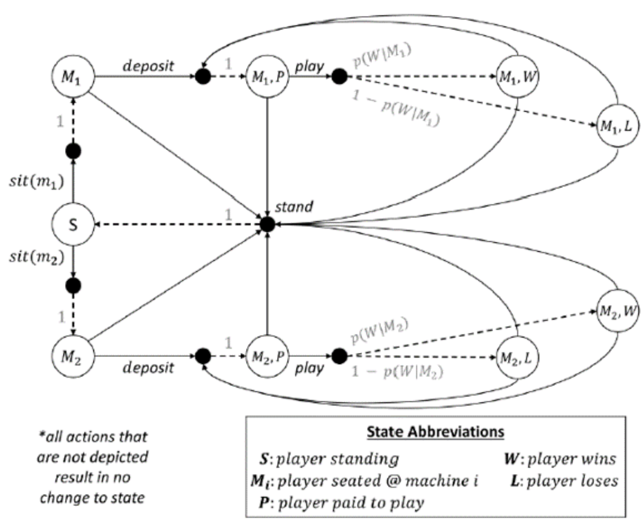

Fig. 8: State diagram of the 2-armed bandit environment [17, Fig. 5]. White nodes depict environmental states. Black nodes depict actions. Links going from white state nodes into black action nodes indicate that this action is taken from that state. Dashed links leaving action nodes are labeled with the probabilities of transitioning to possible states resulting from this action.

represent the actual states of sitting at a machine without or with paid deposit, along with the schemas representing the transition between states. Once this was learned, the agent chose the machine that yielded the highest payoff.

In summary, LIDA's schema mechanism allowed an agent that receives only a binary sensory signal (`Win` or `Loose`) to learn the structure of this environment, and to find adapted behaviors (sitting at the most favorable machine, and then keeping paying deposits and playing) satisfying the motivation of maximizing the value won.

The agent's adaptation to the k-armed bandit environment was made possible by LIDA's distinction between sensori signal and perceptual memory that allows treating sensory signal as mere events rather than elements of the state. Moreover, LIDA treats composite schemas as procedural knowledge (stream of behavior) that are handled as a whole action by the Action Execution module, which facilitates the learning of the sequence `deposit`—`play`, before finding the machine that yields the highest payoff.

2.4 Enactive Cognitive Architecture (ECA)

Similar to Perotto, Georgeon modeled schemas as tuples (pre-condition, decision, post-condition), and organized them hierarchically.

In contrast with Drescher's and Perotto's mechanisms, however, pre-conditions and post-conditions are not properties of the world (hidden or observed) but are other schemas learned previously. The mechanism learns new schemas from the bottom up, with higher-level schemas made of a sequence of two previously-learned lower-level schemas, as illustrated in Fig. 9. The mechanism is not initialized with a top-level goal. Instead, it is initialized with a predefined set of low-level primitive schemas that define the agent's basic possibilities of interaction. In a robot, primitive schemas are hard-coded control loops involving actuators and sensory feedback.

The schema learning mechanism and selection works as follows. At the end of time step t, the agent records or reinforces the schemas:

- $(i_{t-2}, d_{t-1}, i_{t-1})$
- $((i_{t-3}, d_{t-2}, i_{t-2}), d_{t-1}, i_{t-1})$
- $(i_{t-3}, d^2, (i_{t-2}, d_{t-1}, i_{t-1}))$
- $((i_{t-4}, d_{t-3}, i_{t-3}), d^2, (i_{t-2}, d_{t-1}, i_{t-1}))$

If it does not yet exist, the new decision d^2 is constructed different from the decision d_{t-2} that was actually made at time $t-2$. For example, if the agent made decision $d_{t-2} = a0$ and enacted interaction $i_{t-2} = i00$, and then made decision $d_{t-1} = a0$, and enacted interaction $i_{t-1} = i01$, the agent learns the new decision $d^2 = i00a0$ consisting of trying to enact the interaction $i_t = i00$ and then do action $a_{t+1} = a0$. When decision d^2 has been selected and successfully enacted, the mechanism learns higher-level schemas on top of it. The rate of schema construction being constant, the number of schemas grows linearly with time. Older and unused schemas can be forgotten.

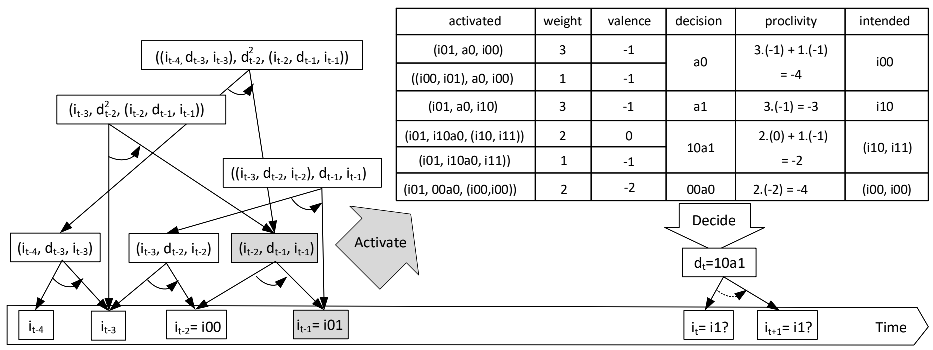

Fig. 9: Schema learning and selection. Schemas are nested tuples: (pre-schema, decision, post-schema). Over time, new decisions and new schemas are learned from the bottom up. Recently enacted schemas (in gray) activate the previously-learned higher-level schema whose pre-schema they match. Activated schemas propose their post-schemas with a proclivity value calculated from the activation weight and the expected valence. The schema with the highest proclivity is selected to try to enact.

In essence, the agent learns habits of interaction that are organized hierarchically, with shorter sequential habits learned first, and longer sequences of habits made of sequences of shorter habits later learned on top of them. Such habit learning relates to Woolford's *sensorimotor sequence reiterator* [45].

Fig. 10 shows Georgeon' experimental settings [10]. This experimentation demonstrates a self-programming effect [9]. Video Demo [7].

Fig. 10: Georgeon's benchmark (Adapted from [10]). The agent can move forward to an empty cell, bump into a wall (green cell), turn to the left or to right by 90°, or "feel" the cell in front, to the left, or to the right. Sensory signal is a single-bit feedback from the action.

2.5 Autocatalytic Endogenous Reflective Architecture (AERA)

AERA[5] is a unified system that can learn about causal relations cumulatively, from experience [41, 42, 20]. The system's knowledge representation uniquely combines ampliative reasoning, autonomous resource management, explanation-driven compositional knowledge representation and reflection in a synergetic architecture that can grow its knowledge from a small "bootstrapping seed" [39]. Its compositional knowledge contains numerous peewee-size "micro-models" that can be combined in explainable ways to model experience [40].

The methodology behind AERA rests on three fundamental principles [44, 38, 21]: 1. During learning, a learner autonomously creates new knowledge through a variety of methods including hypothesis generation and testing; 2. autonomous learning must create models of causal relations through situated goal-driven reasoning, and 3. self-programming is based on transparent operational semantics based on peewee-size uniform knowledge models. Those familiar with Piaget's [2] fundamental insight that human learning requires creation of new information will recognize the first principle. For this principle, Piaget proposed the idea of schemas – information structures that mediate between perception and action, capturing and controlling experience. An important difference between AERA's knowledge representation and other proposals based on this idea (cf. [1, 6]) is its emphasis on non-axiomatic worlds and the resulting strong requirement for transversal defeasibility of acquired knowledge (cf. [33]) – in other words, that since no certainty about anything learned can be guaranteed, the methods for knowledge creation cannot be based on the assumption of knowledge of a "ground truth". The methodology behind AERA unifies this, and the above listed assumptions, in a coherent theoretical approach, with deep implications for knowledge creation in light of novelty: AERA starts out with only a tiny amount of "bootstrap" (seed) knowledge – the remainder of its knowledge, which soon vastly outsizes the seed[6], is then generated autonomously by the system itself, based on hypotheses that are inspired and verified through experience, using ampliative reasoning mechanisms.

- The knowledge representation in AERA is peewee (relatively small) and compositional; AERA can inspect, deconstruct, and reconstruct its models, as well as create hierarchies of models of its experience.
- The models in AERA are falsifiable, have degrees of confidence, and can be related to specific groups with certain attributes specifying their context.
- The choice of models and reasoning methods at any point in time is dynamic, based among other things on the extent of available resources (models and time).

[5] Autocatalytic Endogenous Reflective Architecture; see www.openaera.org – accessed Dec. 6th, 2024.

[6] In our implementation of S1 – an AERA agent that learns to do TV-style interviews, the ratio of the learned knowledge over the seed knowledge, after 20 hours of learning, was 53 [42].

AERA's peewee-size models, similar to e.g. Drescher's [6] schemas and the neuro-symbolic approach presented in [15], represent knowledge by contextualized programs that capture antecedental and successional states of important relationships (e.g. the relationship between a hand and an object that one wants to pick up). Unlike these, AERA places explicit representation of cause-effect relations at its core [43]. During its continuous learning, driven by a self-maintained dynamic goal hierarchy, the most useful (effective and efficient) models of cause-effect relations encountered in the environment are retained, while others are discarded. Based on predictions, informed experimentation, and strategic induction, the system autonomously creates new knowledge cumulatively, effortlessly unifying new information with old, even in light of contradictions. Further separating AERA from other approaches are its atomic elements for knowledge creation, whose operational semantics are transparent to the system's own reasoning, meaning that they can be inspected and learned by the same explicit, defeasible unified reasoning mechanisms as used for learning about the world.

The approach uniquely unifies cognition and meta-cognition in a single, coherent architecture, enabling efficient and effective self-reflection that lays a foundation for a capacity for cognitive growth. The result is a system that can dynamically employ ampliative reasoning, at any point in time, to predict both the immediate and far future, including its own cognitive resource use ("think time").

Key representational components of this approach include [35, 36]:

- **Facts** are statements that represent a cognitive event, e.g. operations happening in AERA itself; facts are the only entity in AERA that have complete certainty (corresponding to Descartes' axiom "I think, therefore I am").
- A **composite State model (CST)** captures a set of simultaneous (timeless) Facts that the AERA system considers to be (potentially) useful and/or necessary for properly predicting an event's consequences; used to e.g. represent a variety of subsets in the world such as spatial relations of objects, the values of a perceived object's properties, such as its position and color, etc.
- A **cognitive event model (CEM)** captures a (hypothesized) causal influence of a Fact on another Fact; used to represent the effect that an action or event in particular circumstances, e.g. a command initiated by AERA at some timestep, has on the state of the world at a later timestep. When a CEM fires it produces a prediction (while the production of a prediction is represented as a Fact, the confidence in the prediction itself is defeasible and is in part based on the confidence of the CEM).
- A **requirement model (**M_{req}**)** captures the (predicted) requirements/conditions that must be met for a CEM to be relevant in a particular context at a particular time.

AERA's programming paradigm allows the system itself to inspect these types of information structures, use them "for parts" to create variations, as well as create new ones from scratch. AERA's programming language, Replicode, allows it to autonomously use these building blocks for self-programming, in the form of coherent plans, predictions, and goals that it can pursue [23, 22, 38]. These mechanisms have been demonstrated for control of virtual robots, learning and using language, and multimodal interaction [42, 35]).

Both CEMs and M_{req}s have a left-hand side (LHS) and a right-hand side (RHS), composed of sets of Facts and constraints (e.g., math equations). The LHS Facts describe (assumed) conditions under which the RHS Facts would be produced. Figure 11 shows how the above components lead to making a prediction. When some sensory Facts match everything in a CEM's LHS, a relevant M_{req} allows the CTS's instantiation, which means the conditions for making a prediction via a CEM are met. A successful instantiation of a CEM means that everything in its RHS will happen. As the CEM predicts a future state after an event's occurrence, this is verified and marked as a "successful prediction" by AERA only if the prediction matches the state that is observed.

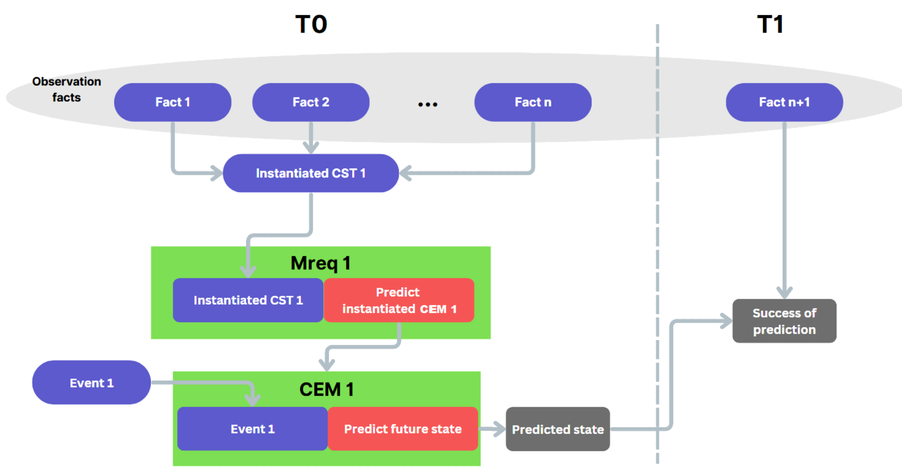

Fig. 11: The connection between CST, M_{req}, and CEM. The observation facts matching $CST1$ allow M_{req} to trigger a prediction by instantiating $CEM1$. AERA checks whether the subsequent observations match the predicted future state.

As an example, the triad of $CST1$, $M_{req}1$, and $CEM1$ can represent the physical movement behavior of an entity when it collides with another entity (e.g., a hand). $CEM1$ represents the dynamics of the movement, connecting the collision event (left-hand side of $CEM1$) to the moved entity's new position (right-hand side of $CEM1$). $CST1$ represents the preconditions of the behavior, such as the entities' position, weight, etc., and the fact that the point of collision and the moved entities' position must initially be the same. $M_{req}1$ predicts that if a striking entity hits another entity with the conditions described by $CST1$, the stricken entity's position will change a specific amount in the next time step.

The interactions of the operational principles of AERA are quite intricate in their entirety; important aspects that are covered elsewhere include induction – the creation of new knowledge – unified abduction-deduction – the creation of plans and explanations – and runtime relevance computations [23]. The unified integration of these operational principles in a single coherent architecture is what gives AERA its power, allowing its seed – initial human-provided knowledge – to be orders of magnitude smaller than what AERA can create on its own through learning from experience.

3 Schema mechanisms and theory of knowledge

Table 1 shows a comparison by the schema mechanisms presented in this paper.

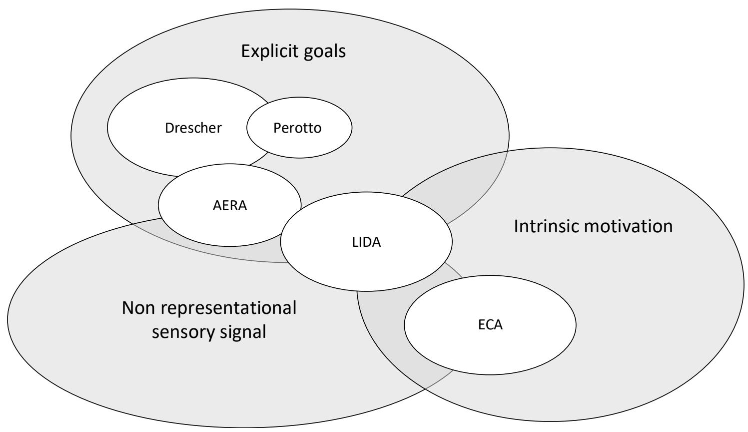

Fig. 12: Coverage of principal dimensions by existing schema mechanisms.

Table 1: Comparison of schema mechanisms

	Drescher	Perotto	AERA	LIDA	ECA
Explicit goal	✓	✓	✓	✓	
Concept invention	✓	✓			
Non-representational sensory signal				✓	✓
Intrinsic motivation				✓	✓
3D spatial schemas				✓	✓
Self-programming					✓

Bettoni has criticized Drescher's schema mechanism in stating that "Drescher's Constructivism is not Piaget's Constructivism, mainly because of its tacit acceptance of *cognitive dogmatism*" ([3], p. 6). Bettoni describes cognitive dogmatism as taking for granted that "patterns and structures of objects, attributes, relations, etc. [...] be as much as possible true copies of 'original' objects, attributes, relations etc. in the world" ([3], p. 1). Indeed, theories of enaction as well as of radical constructivism have insisted that we should not take the sensory signals as representational items of an alleged reality.

The game of Mastermind provides an emblematic example in which the observation is not representational. Player 2 attempts to infer a hidden combination of colored pegs ("hidden state") by proposing guesses ("actions"), which Player 1 responds to with feedback pegs ("observation"). Black pegs indicate a correct color in the correct position, while white pegs indicate a correct color in the wrong position. Since the observation depends on the action, there exist no function or stochastic distribution that map the state to the observation. The absence of such function or distribution is expressed in cognitive terms by that the observation is not "representational" of the state.

Software to play Mastermind have been proposed using diverse techniques such as entropy measure and evolutionary algorithms [5]. These techniques, however, require that the semantics of the feedback is known beforehand. We propose the *constructivist Mastermind analogy* that likens a general learner to someone playing a giant game of mastermind where they start with no knowledge of the hidden combination and even the semantics of actions and feedback. The player may never find the hidden combination or goal but may survive and strive for some time in a satisfactory *knowledge niche*.

As reviewed in Section **??**, most schema mechanisms have been tested in settings in which the sensory signal is representational. Nonetheless, the fact that they have not been tested with non-representational sensory signal does not mean that they would not work or could not be adapted to such settings. Georgeon's mechanism may constitute an illustration of that. In Georgeon's schemas, the sensory data is feedback from action rather than being representational. The pre-condition and post-condition of schemas are are just other schemas all the way down to non-representational primitive schemas. In essence, the agent knows its current context by possibilities of interaction rather than by representational data.

Georgeon's schema mechanism is not targeted at reaching a predefined goal. Since sensory data does not represent the environment's state, the agent cannot have a goal represented as an environment state. Instead, the agent's behavior is driven by the expected valence of each decision. The calculation of expected valence may incorporate predefined preferences for some primitive schemas or different intrinsic motivation principles such as an estimation of information gained.

4 Conclusion

Beyond the first stage called by Piaget the *sensorimotor level*, comes a second stage he calls the *first level of pre-operative thought* ([31], p. 30). It is at this second stage that the subject begins to differentiate itself from the objects. The subject becomes able to manipulate the schemas by thought. Piaget formulates this process as follows:

> "On top of simple actions that ensure direct interdependence between the subject and objects, in certain cases, a new type of action is superimposed, one that is internalized and more precisely conceptualized: for example, in addition to being able to move from A to B, the subject acquires the ability to mentally represent this movement from A to B and to evoke, through thought, other movements" (translated from [31], p. 30).

Future studies on schema mechanisms will have to address this second stage to tackle fundamental questions of knowledge abstraction. We expect this will involve a cognitive architecture that has the capacity to represent spatial properties of schemas, and to simulate the enaction of schema in different spatial frames of reference. LIDA and ECA [8] constitute initial attempts in this direction .

Acknowledgments. .

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arbib, M.A.: The handbook of brain theory and neural networks. MIT press (2003)
2. Berlyne, D., Piaget, J., Piercy, M.: The Psychology of intelligence. Routledge (2003)
3. Bettoni, M.C.: Made-up minds: A constructivist approach to artificial intelligence - a book review **6**(3), 234–240. https://doi.org/10.3233/AIC-1993-63-413
4. Chaput, H.: The Constructivist Learning Architecture: A model of cognitive development for robust autonomous robots. phdthesis
5. Cotta, C., Merelo Guervós, J.J., Mora Garćia, A.M., Runarsson, T.P.: Entropy-driven evolutionary approaches to the mastermind problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI. pp. 421–431. Springer. https://doi.org/10.1007/978-3-642-15871-1_43
6. Drescher, G.L.: Made-up minds: a constructivist approach to artificial intelligence. Artificial intelligence, MIT Press
7. Georgeon, O.L.: The small loop challenge (2012), https://youtu.be/Mstl8BRCbSI
8. Georgeon, O.L., Lurie, D., Robertson, P.: Artificial enactive inference in three-dimensional world. Cognitive Systems Research **86**, 101234 (2024). https://doi.org/10.1016/j.cogsys.2024.101234
9. Georgeon, O.L., Riegler, A.: CASH only: Constitutive autonomy through motorsensory self-programming **58**, 366–374 (2019). https://doi.org/10.1016/j.cogsys.2019.08.006
10. Georgeon, O.L., Ritter, F.E.: An intrinsically-motivated schema mechanism to model and simulate emergent cognition **15-16**, 73–92. https://doi.org/10.1016/j.cogsys.2011.07.003, https://linkinghub.elsevier.com/retrieve/pii/S1389041711000398

11. Glasersfeld, E.v.: Radical constructivism: a way of knowing and learning. No. 6 in Studies in mathematics education series, Falmer Press, reprinted edn.
12. Guerin, F., Kruger, N., Kraft, D.: A survey of the ontogeny of tool use: From sensorimotor experience to planning **5**(1), 18–45. https://doi.org/10.1109/TAMD.2012.2209879, http://ieeexplore.ieee.org/document/6248675/
13. Guerin, F., McKenzie, D.: A piagetian model of early sensorimotor development. In: Proceedings of the Eighth International Conference on Epigenetic Robotics. pp. 29–36. Lund University Cognitive Studies, Kognitionsforskning, Lunds universitet, issue: -
14. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena **42**(1), 335–346 (1990). https://doi.org/10.1016/0167-2789(90)90087-6
15. Komrusch, S., Minsky, H.: Symbolic guidance for constructivist learning by neural model. In: Proceedings of the Third International Workshop on Self-Supervised Learning. pp. 63–76. PMLR, https://proceedings.mlr.press/v192/komrusch22a.html, ISSN: 2640-3498
16. Kringelbach, M., Berridge, K.: The functional neuroanatomy of pleasure and happiness **9**, 579–87 (2010)
17. Kugele, S.: Constructivist procedural learning for grounded cognitive agents (2025)
18. Kugele, S., Franklin, S.: Learning in LIDA **66**, 176–200 (2021). https://doi.org/10.1016/j.cogsys.2020.11.001
19. Miller, M.S.P.: Building Minds with Patterns. Independently published
20. Nivel, E., Þórisson, K.R., Thórisson, K.R., Dindo, H., Pezzulo, G., Rodriguez, M., Corbato, C., Steunebrink, B., Ognibene, D., Chella, A., et al.: Autocatalytic endogenous reflective architecture (2013)
21. Nivel, E., Thórisson, K.R.: Self-programming: Operationalizing autonomy. In: 2nd Conference on Artificial General Intelligence (2009). pp. 212–217. Atlantis Press (2009)
22. Nivel, E., Thórisson, K.R.: Replicode: A constructivist programming paradigm and language. Technical RUTR-SCS13001, Reykjavik University School of Computer Science (2013)
23. Nivel, E., Thórisson, K.R.: Towards a programming paradigm for control systems with high levels of existential autonomy. In: International Conference on Artificial General Intelligence. pp. 78–87. Springer (2013)
24. Oudeyer, P.Y., Kaplan, F., Hafner, V.V.: Intrinsic motivation systems for autonomous mental development **11**(2), 265–286 (2007). https://doi.org/10.1109/TEVC.2006.890271
25. Perotto, F.S.: Un Mécanisme Constructiviste d'Apprentissage Automatique d'Anticipations pour des Agents Artificiels Situés. Ph.D. thesis, National Polytechnic Institute of Toulouse, France (2010), https://tel.archives-ouvertes.fr/tel-00620755
26. Perotto, F.S.: A computational constructivist model as an anticipatory learning mechanism for coupled agent environment systems. Constructivist foundations **1**(9), 46–56 (2013)
27. Perotto, F.S., Alvares, L.O.: Incremental inductive learning in a constructivist agent. In: Bramer, M., Coenen, F., Tuson, A. (eds.) Research and Development in Intelligent Systems XXIII, Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, 11-13 December, 2006. pp. 129–142. Springer (2006). https://doi.org/10.1007/978-1-84628-663-6_10, https://doi.org/10.1007/978-1-84628-663-6_10

28. Perotto, F.S., Alvares, L.O.: Learning regularities with a constructivist agent. In: Nakashima, H., Wellman, M.P., Weiss, G., Stone, P. (eds.) 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006. pp. 807–809. ACM (2006). https://doi.org/10.1145/1160633.1160778, https://doi.org/10.1145/1160633.1160778
29. Perotto, F.S., Buisson, J.C., Alvares, L.O.C.: Constructivist anticipatory learning mechanism (calm): Dealing with partially deterministic and partially observable environments. In: International Conference on Epigenetic Robotics (EpiRob), Piscataway, NJ, USA, 29/10/2007-02/11/2007. pp. 117–127. Lund University Cognitive Science (2007)
30. Piaget, J.: La naissance de l'intelligence chez l'enfant. Actualités pédagogiques et psychologiques, Delachaux et Niestlé
31. Piaget, J.: L'épistémologie génétique. Quadrige, PUF
32. Piaget, J.: The Principles of Genetic Epistemology. Psychology Press
33. Pollock, J.L.: Defeasible reasoning. Cognitive science **11**(4), 481–518 (1987)
34. Raju, R.V., Guntupalli, J.S., Zhou, G., Lázaro-Gredilla, M., George, D.: Space is a latent sequence: Structured sequence learning as a unified theory of representation in the hippocampus (arXiv:2212.01508) (2022), http://arxiv.org/abs/2212.01508
35. Sheikhlar, A.: Autonomous causal generalization (2024)
36. Sheikhlar, A., Thórisson, K.R.: Causal generalization via goal-driven analogy. In: International Conference on Artificial General Intelligence. pp. 165–175. Springer (2024)
37. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, 2 edn. (2018)
38. Thórisson, K.R.: A new constructivist ai: from manual methods to self-constructive systems. In: Theoretical Foundations of Artificial General Intelligence, pp. 145–171. Springer (2012)
39. Thórisson, K.R.: Seed-programmed autonomous general learning. In: International Workshop on Self-Supervised Learning. pp. 32–61. PMLR (2020)
40. Thórisson, K.R.: The" explanation hypothesis" in general self-supervised learning. In: IWSSL. pp. 5–27 (2021)
41. Thórisson, K.R., Bieger, J., Li, X., Wang, P.: Cumulative learning. In: Artificial General Intelligence: 12th International Conference, AGI 2019, Shenzhen, China, August 6–9, 2019, Proceedings 12. pp. 198–208. Springer (2019)
42. Thórisson, K.R., Nivel, N., Steunebrink, B.R., Helgason, H.P., Pezzulo, G., Sanz Bravo, R., Schmidhuber, J., Dindo, H., Rodríguez Hernández, M., Chella, A., et al.: Autonomous acquisition of natural situated communication. IADIS International Journal on Computer Science And Information Systems **9**(2), 115–131 (2014)
43. Thórisson, K.R., Talbot, A.: Cumulative learning with causal-relational models. In: Artificial General Intelligence: 11th International Conference, AGI 2018, Prague, Czech Republic, August 22-25, 2018, Proceedings 11. pp. 227–237. Springer (2018)
44. Thórisson, K.R.: Seed-programmed autonomous general learning. In: Proceedings of the First International Workshop on Self-Supervised Learnin. Proceedings of Machine Learning Research, vol. 131, pp. 32–61. http://proceedings.mlr.press/v131/
45. Woolford, F.M.G., Egbert, M.D.: A precarious sensorimotor sequence reiterator for modelling enactive habits. In: ALIFE 2020: The 2020 Conference on Artificial Life. pp. 771–779. MIT Press (2020). https://doi.org/10.1162/isal_a_00249

46. Ziemke, T.: The construction of 'reality' in the robot: Constructivist perspectives on situated artificial intelligence and adaptive robotics **6**(1), 163–233. https://doi.org/10.1023/A:1011394317088, https://doi.org/10.1023/A:1011394317088